



分散メーク (dmake)

Sun™ Studio 10

Sun Microsystems, Inc.
www.sun.com

Part No. 819-1618-10
2005 年 1 月, Revision A

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。

フォント技術を含む第三者のソフトウェアは、著作権法により保護されており、提供者からライセンスを受けているものです。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

Sun、Sun Microsystems、Java、および JavaHelp は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC の商標はライセンス規定に従って使用されており、米国および他の各国における SPARC International, Inc. の商標または登録商標です。SPARC の商標を持つ製品は、Sun Microsystems, Inc. によって開発されたアーキテクチャに基づいています。

このマニュアルに記載されている製品および情報は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト(輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む)に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含み、明示的であるか黙示的であるかを問わず、あらゆる説明および保証は、法的に無効である限り、拒否されるものとします。

原典:	<i>Distributed Make (dmake): Sun Studio 10</i> Part No: 819-0506-10 Revision A
-----	--



Please
Recycle



Adobe PostScript

目次

はじめに	v
形式	1
機能説明	1
オプション	2
-c dmake_rcfile	2
-g dmake_group	2
-j dmake_max_jobs	3
-m {serial parallel distributed}	3
-o dmake_odir	3
使用	4
特殊な目的のターゲット	4
.NO_PARALLEL	4
.PARALLEL	4
.LOCAL	4
.WAIT	4
dmake ジョブの制御	5
dmake ホスト	5
ユーザーコマンド	7
構築サーバー	8

環境変数 / マクロ	9
DMAKE_RCFILE	9
DMAKE_GROUP	9
DMAKE_MAX_JOBS	9
DMAKE_ADJUST_MAX_JOBS	9
DMAKE_MODE	9
DMAKE_ODIR	10
DMAKE_OUTPUT_MODE	10
ファイル	11
関連項目	11

はじめに

このマニュアルでは、Sun™ Studio 10 の分散メーク (dmake) コマンドについて説明します。分散メーク (dmake) はメークファイルを解析して、並行して構築可能なターゲットを特定し、設定された多数のホストにの構築作業を分散します。

このマニュアルの情報は、Solaris の make ユーティリティのマニュアルページを補完するものです。

書体と記号について

書体または記号*	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例。	<code>.login</code> ファイルを編集します。 <code>ls -a</code> を実行します。 <code>% You have mail.</code>
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して表します。	<code>% su</code> Password:
<i>AaBbCc123</i> またはゴシック	コマンド行の可変部分。実際の名前や値と置き換えてください。	<code>rm filename</code> と入力します。 <code>rm</code> ファイル名 と入力します。
『 』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照。 この操作ができるのは「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅をこえる場合に、継続を示します。	<code>% grep `^#define` \</code> <code>XV_VERSION_STRING'</code>

* 使用しているブラウザにより、これら設定と異なって表示される場合があります。

シェルプロンプトについて

シェル	プロンプト
UNIX の C シェル	マシン名%
UNIX の Bourne シェルと Korn シェル	\$
スーパーユーザー (シェルの種類を問わない)	#

サポートされるプラットフォーム

この Sun Studio リリースは、SPARC® および x86 ファミリ (UltraSPARC®, SPARC64, AMD64, Pentium, Xeon EM64T) プロセッサアーキテクチャをサポートしています。サポートされるシステムの、Solaris オペレーティングシステムのバージョンごとの情報については、<http://www.sun.com/bigadmin/hcl> にあるハードウェアの互換性に関するリストで参照することができます。ここには、すべてのプラットフォームごとの実装の違いについて説明されています。

このドキュメントでは、“x86” という用語は、AMD64 または Intel Xeon/Pentium 製品ファミリと互換性があるプロセッサを使用して製造された 64 ビットおよび 32 ビットのシステムを指します。サポートされるシステムについては、ハードウェアの互換性に関するリストを参照してください。

Sun Studio ソフトウェアおよびマニュアルページへのアクセス

Sun Studio ソフトウェアおよびマニュアルページは、`/usr/bin/` と `/usr/share/man` ディレクトリにはインストールされません。ソフトウェアにアクセスするには、`PATH` 環境変数を正しく設定しておく必要があります (vii ページの「ソフトウェアへのアクセス方法」を参照)。また、マニュアルページにアクセスするには、`MANPATH` 環境変数を正しく設定しておく必要があります (vii ページの「マニュアルページへのアクセス方法」を参照)。

PATH 変数についての詳細は、`cs(1)`、`sh(1)`、`ksh(1)`、および `bash(1)` のマニュアルページを参照してください。MANPATH 変数についての詳細は、`man(1)` のマニュアルページを参照してください。

注 – この節に記載されている情報は Sun Studio のソフトウェアが `/opt` ディレクトリにインストールされていることを想定しています。製品ソフトウェアが `/opt` 以外のディレクトリにインストールされている場合は、システム管理者に実際のパスをお尋ねください。

ソフトウェアへのアクセス方法

PATH 環境変数を変更してソフトウェアにアクセスできるようにする必要があるかどうか判断するには以下を実行します。

PATH 環境変数を設定する必要があるかどうか判断する

1. 次のように入力して、PATH 変数の現在値を表示します。

```
% echo $PATH
```

2. 出力内容から `/opt/SUNWspro/bin` を含むパスの文字列を検索します。

パスがある場合は、PATH 変数はソフトウェアのツールにアクセスできるように設定されています。このパスがない場合は、次の手順に従って、PATH 環境変数を設定してください。

PATH 環境変数を設定してソフトウェアにアクセスする

- 次のパスを PATH 環境変数に追加します。Forte Developer ソフトウェア、Sun ONE Studio ソフトウェア、または Sun Studio の他のリリースをインストールしている場合は、インストール先のパスの前に、次のパスを追加します。

```
/opt/SUNWspro/bin
```

マニュアルページへのアクセス方法

マニュアルページにアクセスするために MANPATH 環境変数を変更する必要があるかどうかを判断するには以下を実行します。

MANPATH 環境変数を設定する必要があるかどうか判断する

1. 次のように入力して、dbx のマニュアルページを表示します。

```
% man dbx
```

2. 出力された場合、内容を確認します。

dbx(1) のマニュアルページが見つからないか、表示されたマニュアルページがインストールされたソフトウェアの現バージョンのものと異なる場合は、この節の指示に従って、MANPATH 環境変数を設定してください。

MANPATH 環境変数を設定してマニュアルページにアクセスする

- 次のパスを MANPATH 環境変数に追加します。

```
/opt/SUNWspro/man
```

統合開発環境へのアクセス方法

Sun Studio 統合開発環境 (IDE) には、C や C++、Fortran アプリケーションを作成、編集、構築、デバッグ、パフォーマンス解析するためのモジュールが用意されています。

IDE を起動するコマンドは、`sunstudio` です。このコマンドの詳細は、`sunstudio(1)` のマニュアルページを参照してください。

IDE が正しく動作するかどうかは、IDE がコアプラットフォームを検出できるかどうか依存します。このため、`sunstudio` コマンドは、次の 2 つの場所でコアプラットフォームを探します。

- コマンドは、最初にデフォルトのインストールディレクトリ `/opt/netbeans/3.5V` を調べます。
- このデフォルトのディレクトリでコアプラットフォームが見つからなかった場合は、IDE が含まれているディレクトリとコアプラットフォームが含まれているディレクトリが同じであるか、同じ場所にマウントされているとみなします。たとえば IDE が含まれているディレクトリへのパスが `/foo/SUNWspro` の場合は、`/foo/netbeans/3.5V` ディレクトリにコアプラットフォームがないか調べます。

`sunstudio` が探す場所のどちらにもコアプラットフォームをインストールしていないか、マウントしていない場合、クライアントシステムの各ユーザーは、コアプラットフォームがインストールされているか、マウントされている場所 (`/installation_directory/netbeans/3.5V`) を、`SPRO_NETBEANS_HOME` 環境変数に設定する必要があります。

Forte Developer ソフトウェア、Sun ONE Studio ソフトウェア、または他のバージョンの Sun Studio ソフトウェアがインストールされている場合、IDE の各ユーザーは、\$PATH のそのパスの前に、`/installation_directory/SUNWspro/bin` を追加する必要もあります。

\$PATH には、`/installation_directory/netbeans/3.5V/bin` のパスは追加しないでください。

Sun Studio マニュアルへのアクセス方法

マニュアルには、以下からアクセスできます。

- 製品マニュアルは、ご使用のローカルシステムまたはネットワークの製品にインストールされているマニュアルの索引から入手できます。
`file:/opt/SUNWspro/docs/ja/index.html`

製品ソフトウェアが `/opt` 以外のディレクトリにインストールされている場合は、システム管理者に実際のパスをお尋ねください。

- マニュアルは、`docs.sun.comsm` の Web サイトで入手できます。以下に示すマニュアルは、Solaris プラットフォームにインストールされているソフトウェアからアクセスできます。
 - 『Standard C++ Library Class Reference』
 - 『標準 C++ ライブラリ・ユーザズガイド』
 - 『Tools.h++ クラスライブラリ・リファレンスマニュアル』
 - 『Tools.h++ ユーザズガイド』
- `docs.sun.com` Web サイトからは、Solaris および Linux 両方のプラットフォーム用のリリースノート入手できます。
- IDE の全コンポーネントのオンラインヘルプは、IDE 内の「ヘルプ」メニューだけでなく、多くのウィンドウおよびダイアログにある「ヘルプ」ボタンを使ってアクセスできます。

インターネットの Web サイト (<http://docs.sun.com>) から、Sun のマニュアルを参照したり、印刷したり、購入することができます。マニュアルが見つからない場合はローカルシステムまたはネットワークの製品とともにインストールされているマニュアルの索引を参照してください。

注 - Sun では、本マニュアルに掲載した第三者の Web サイトのご利用に関しましては責任はなく、保証するものでもありません。また、これらのサイトあるいはリソースに関する、あるいはこれらのサイト、リソースから利用可能であるコンテンツ、広告、製品、あるいは資料に関して一切の責任を負いません。Sun は、これらのサイトあるいはリソースに関する、あるいはこれらのサイトから利用可能であるコンテンツ、製品、サービスのご利用あるいは信頼によって、あるいはそれに関連して発生するいかなる損害、損失、申し立てに対する一切の責任を負いません。

アクセシブルな製品マニュアル

マニュアルは、技術的な補足をすることで、ご不自由なユーザーの方々にとって読みやすい形式のマニュアルを提供しております。アクセシブルなマニュアルは以下の表に示す場所から参照することができます。製品ソフトウェアが /opt 以外のディレクトリにインストールされている場合は、システム管理者に実際のパスをお尋ねください。

マニュアルの種類	アクセシブルな形式と格納場所
マニュアル (サードパーティ製マニュアルは除く)	形式 : HTML 場所 : http://docs.sun.com
サードパーティ製マニュアル	形式 : HTML 場所 : file:/opt/SUNWspr/docs/ja/index.html の マニュアル索引
<ul style="list-style-type: none">『Standard C++ Library Class Reference』『標準 C++ ライブラリ・ユーザーズガイド』『Tools.h++ クラスライブラリ・リファレンスマニュアル』『Tools.h++ ユーザーズガイド』	
Readme およびマニュアルページ	形式 : HTML 場所 : file:/opt/SUNWspr/docs/ja/index.html の マニュアル索引
オンラインヘルプ	形式 : HTML 場所 : IDE 内の「ヘルプ」メニュー
リリースノート	形式 : HTML 場所 : http://docs.sun.com

関連する Solaris マニュアル

次の表では、docs.sun.com の Web サイトで参照できる関連マニュアルについて説明します。

マニュアルコレクション	マニュアルタイトル	内容の説明
Solaris Reference Manual Collection	マニュアルページのセクションのタイトルを参照。	Solaris オペレーティングシステムに関する情報を提供しています。
Solaris Software Developer Collection	リンカーとライブラリ	Solaris のリンクエディタと実行時リンカーの操作について説明しています。
Solaris Software Developer Collection	マルチスレッドのプログラミング	POSIX と Solaris スレッド API、同期オブジェクトのプログラミング、マルチスレッド化したプログラムのコンパイル、およびマルチスレッド化したプログラムのツール検索について説明します。

開発者向けのリソース

<http://developers.sun.com/prodtech/cc> にアクセスし、以下のようなリソースを利用できます。リソースは頻繁に更新されます。

- プログラミング技術と最適な演習に関する技術文書
- プログラミングに関する簡単なヒントを集めた知識ベース
- ソフトウェアのマニュアル、およびソフトウェアとともにインストールされるマニュアルの訂正
- サポートレベルに関する情報
- ユーザーフォーラム
- ダウンロード可能なサンプルコード
- 新しい技術の紹介

<http://developers.sun.com> でも開発者向けのリソースが提供されています。

技術サポートへの問い合わせ

製品についての技術的なご質問がございましたら、以下のサイトからお問い合わせください (このマニュアルで回答されていないものに限りです)。

<http://jp.sun.com/service/contacting>

分散メーク(dmake)

形式

```
dmake [-c dmake_rcfile] [-g dmake_group] [-j dmake_max_jobs]
[-m serial | parallel | distributed] [-o dmake_ouir]
```

機能説明

注 - このマニュアルの情報は、Solaris の make ユーティリティのマニュアルページを補完するものです。dmake ユーティリティは Linux オペレーティングシステムでも動作しますが、dmake のオプションは他の make ユーティリティと比べて独自のものです。

分散メーク (dmake) はメークファイルを解析して、並行して構築可能なターゲットを特定し、設定された多数のホストにの構築作業を分散します。

標準の make ユーティリティを使用している場合、dmake へ移行する際に必要なメークファイルの変更は、(あったとしても) ほとんどありません。dmake は make ユーティリティのスーパーセットです。make を入れ子にするときは、一番上の makefile が make を呼び出す場合に \$(MAKE) を使用しなければなりません。

分散メークは、"dmake ホスト" で実行され、"ジョブ" を "構築サーバー" に分散します。ジョブは dmake ホストにも分散できます。この場合は、dmake ホストも構築サーバーになります。dmake は、ユーザーのメークファイルの内容を解析し、並行して構築できると判断したターゲットごとにジョブを分散させます。

次の要件を満たしていれば、どのマシンも構築サーバーとして使用できます。

- dmake ホスト (使用中のマシン) から、rsh を使用して構築サーバー上でコマンドをリモートで実行できる。このとき、パスワードの入力を求めるプロンプトは表示されないこと。次の例は、Solaris オペレーティングシステムを実行しているマシンの場合の dmake に対する which コマンドの使用結果を示しています。

```
demo% rsh <分散メークを行うマシン名> which dmake
/opt/SUNWspro/bin/dmake
```

rsh を実行した時に、不要な出力があってははいけません。

- dmake ソフトウェアがインストールされている bin ディレクトリに、構築サーバーからアクセス可能でなければなりません。デフォルトで、dmake は、構築サーバー上の dmake 実行可能ファイルへの論理パスが dmake ホスト上と同じであると仮定します。この仮定を変更するには、実行時構成ファイルのホストエントリの属性としてパス名を指定します。

dmake ホストから、使用する構築サーバー、および、各構築サーバーに割り当てる dmake ジョブ数を指定することができます。特定の構築サーバー上で実行することができる dmake ジョブ数の制限は、そのサーバーにも適用されます。

オプション

-c dmake_rcfile

実行時構成ファイルを指定します。デフォルトの実行時構成ファイルは ~/.dmake.rc です。

-g dmake_group

ジョブの分散先にする構築サーバーグループの名前を指定します。サーバーグループは実行時構成ファイルに定義してください。デフォルトのサーバーグループは、実行時構成ファイル中の最初のグループです。

-j dmake_max_jobs

実行時構成ファイル中の構築サーバーの指定グループに分散される最大ジョブ総数を指定します。デフォルトの最大ジョブ数は、構築サーバーグループに指定されるジョブの総数です。ジョブは、実行時構成ファイルの中で出現する順に1つずつホストから減算、またはホストに加算されます。たとえば、実行時構成ファイルに指定されるジョブの総数が8で、

```
host earth { jobs = 3 }
host mars  { jobs = 5 }
```

<分散する最大ジョブ数> を11と指定する場合、dmakeは次のようにして、現在の最大ジョブ総数(8)にさらに3つのジョブを加算します。

```
host earth { jobs = 5 }
host mars  { jobs = 6 }
```

また、<分散する最大ジョブ数> を4と指定する場合、dmakeは次のようにして、元のジョブ数8から4つのジョブを減算します。

```
host earth { jobs = 1 }
host mars  { jobs = 3 }
```

-m {serial | parallel | distributed}

次のいずれかのキーワードを指定します。

- serial: dmake を標準の make と同じように逐次動作させます。
- parallel: ジョブを dmake ホストにだけ分散させます。
- distributed: dmake を完全な分散モードで動作させます。これは dmake のデフォルトです。
- grid: 構築ジョブの分散に SGE (Sun Grid Engine) を使います。

-o dmake_odir

dmake が一時出力ファイルを書き込みおよび読み取りできる共通物理ディレクトリを指定します。使用されるディレクトリは $\$(HOME)/.dmake$ です。このディレクトリと指定したディレクトリは、すべての構築サーバーから見えなければなりません。このオプションは、ユーザーのローカルホスト上の $\$(HOME)$ ディレクトリとユーザーのすべての遠隔ホスト上の $\$(HOME)$ ディレクトリが、異なる物理 $\$(HOME)$ ディレクトリである場合にだけ使用してください。たとえば、このオプションはスーパーユーザーが使用します。

注 - これらのオプションと環境変数、およびこのマニュアルページで後述するメイクファイルのマクロによって、同じ動作が変更されます。優先順位は次のとおりです。

1. コマンド行オプション
2. メイクファイルのマクロ
3. 環境変数
4. `dmake` のデフォルト

使用

特殊な目的のターゲット

`dmake` を使うと、構築サーバー上でターゲットを並行して構築できます。並行して処理を行うと、大規模なシステムやプロジェクトを構築するのに要する時間を大幅に短縮できます。`dmake` は、並行処理とタイミングを制御するために `.PARALLEL`、`.NO_PARALLEL`、`.LOCAL`、`.WAIT` という、特殊なメイクファイルターゲットを使用します。

`.NO_PARALLEL`

このターゲットは、逐次処理を行うターゲットを指定するために使用します。

`.PARALLEL`

このターゲットは、並行処理を行うターゲットを指定するために使用します。

`.LOCAL`

このターゲットは、ローカルホスト上で逐次処理を行うターゲットを指定するために使用します。

`.WAIT`

依存関係リストにこのターゲットを指定すると、`dmake` は、並行処理の場合でも、この前の依存関係の処理が終了するまで待機してから、次の依存関係を処理します。

このようなターゲットを使って記述してあるメイクファイルは、Solaris 1.x と Solaris 2.x で配布される標準 make でも互換性を保っています。標準の make はこれらのターゲットを許容し、エラーを出しません (処理も行いません)。

dmake ジョブの制御

dmake ジョブの分散は、次の 2 つの方法で制御されます。

1. dmake ホスト上の dmake ユーザーは、構築サーバーとして使用したいマシンと各構築サーバーに分散したいジョブ数を指定することができます。
2. 構築サーバーの所有者 (/etc/opt/SPROdmake/dmake.conf ファイルを変更できるユーザー) は、その構築サーバーに分散可能な dmake ジョブの最大総数を制御することができます。

dmake ホスト

dmake は、実行を開始すると、実行時構成ファイルを検索して、ジョブを分散する場所を調べます。一般に、このファイルは dmake ホスト上のユーザーのホームディレクトリに置いて、.dmakerc という名前を付けます。dmake は、次の場所と順序で実行時構成ファイルを検索します。

1. -c オプションを使ってコマンド行で指定されたパス名
2. DMAKE_RCFILE というメイクファイルのマクロを使って指定されたパス名
3. 環境変数 DMAKE_RCFILE を使って指定されたパス名
4. ~/.dmakerc

実行時構成ファイルが見つからない場合は、dmake は並行モードに切り替え、(デフォルトの) 2 つのジョブを dmake ホストに分散します。-j オプションまたは DMAKE_MAX_JOBS を使用してこの設定を変更することもできます。

実行時構成ファイルには、構築サーバーのリストと、各構築サーバーに分散したいジョブ数を含めることができます。次は、単純な実行時構成ファイルの例です。

```
# 自分のマシン。このエントリにより dmake はこのマシンにジョブを分散します。
falcon { jobs = 1 }
hawk
eagle { jobs = 3 }
# マネージャーのマシン。マネージャーはいつも会議中です
heron { jobs = 4 }
avocet
```

falcon、hawk、eagle、heron、avocet の各エントリが構築サーバーとして一覧表示されています。

各構築サーバーに分散したいジョブ数を指定することができます。デフォルトのジョブ数は 2 です。

"#" 文字で始まる行はコメントとして解釈されます。

注 - この構築サーバーの一覧には、dmake ホストでもある falcon が含まれています。dmake ホストは構築サーバーとしても指定することができます。実行時構成ファイルに falcon を含めない場合、falcon には dmake ジョブが分散されません。

実行時構成ファイルには、構築サーバーのグループも構成することができます。こうすると、状況に応じて、構築サーバーのグループを簡単に切り替えられるという柔軟性が得られます。たとえば、異なるオペレーティングシステムでの構築用に別々の構築サーバーのグループを定義したり、特殊なソフトウェアがインストールされている構築サーバーのグループを定義したりすることができます。ただし、構築サーバーは、すべて同じアーキテクチャで、同じ SunOS をインストールしている必要があります。

次の実行時構成ファイルには、いくつかのグループが含まれています。

```
earth                { jobs = 2 }
mars                 { jobs = 3 }

group lab1 {
    host falcon{ jobs = 3 }
    host hawk
    host eagle{ jobs = 3 }
}

group lab2 {
    host heron
    host avocet{ jobs = 3 }
    host stilt{ jobs = 2 }
}

group labs {
    group lab1
    group lab2
}

group sunos5.x {
    group labs
    host jupiter
    host venus{ jobs = 2 }
    host pluto { jobs = 3 }
}
```

ユーザーコマンド

"group" 命令によって形式上のグループを指定します。グループのメンバーのリストは、中括弧 {} で括ります。グループのメンバーである構築サーバーは、省略可能な "host" 命令によって指定されます。

あるグループが、他のグループのメンバーになっていてもかまいません。

個々の構築サーバーを、構築サーバーのグループを含む実行時構成ファイルに指定することもできます。この場合、dmake はこれらの構築サーバーを名前の付いていないグループのメンバーとして処理します。

dmake は、次に示すリストで指定される単一のホストのグループに、1 ~ 4 の優先順位でジョブを分散させます。

1. -g オプションへの引数としてコマンド行に指定されたグループ

2. DMAKE_GROUP というメークファイルのマクロによって指定されたグループ
3. 環境変数 DMAKE_GROUP によって指定されたグループ
4. 実行時構成ファイルで 1 番目に列挙されている形式上のグループ

実行時構成ファイルに指定するグループとホストの名前は、二重引用符で囲むことができます。これにより、グループとホストの名前をより柔軟な文字シーケンスで表現できます。たとえば、グループの名前が数字で始まる場合、この名前は二重引用符で囲まなければなりません。

```
group "123_sparc"
```

上記のとおり、dmake ソフトウェアがインストールされている bin ディレクトリは、構築サーバーからアクセス可能でなければなりません。デフォルトでは dmake は、構築サーバー上の dmake 実行可能ファイルへの論理パスが dmake ホスト上と同じであると仮定します。この仮定を変更するには、実行時構成ファイルのホストエントリの属性としてパス名を指定します。次に例を示します。

```
group sparc-cluster {
  host wren { jobs = 10 , path = "/export/SUNWspro/bin" }
  host stimp { path = "/opt/SUNWspro/bin" }
}
```

構築サーバー

/etc/opt/SPROdmake/dmake.conf ファイルは、構築サーバーのファイルシステム中に置かれています。このファイルは、次の内容を指定するために使用します。

- Required: 該当する構築サーバー上で並行処理できる、すべてのユーザーからの dmake ジョブの最大総数 (必須指定)。
- Optional: すべての dmake ジョブに適用される、/usr/bin/nice で指定する優先順位 (必要に応じて)。

次に dmake.conf ファイルの例を示します。

```
max_jobs: 8
nice_prio: 5
```

このファイルは、すべての dmake ユーザーから構築サーバー上で実行が許可される最大 dmake ジョブ数を 8 に設定します。ジョブを実行する優先順位は、nice_prio コマンドで変更することができます。詳細は、nice(1) を参照してください。

注 - /etc/opt/SPROdmake/dmake.conf ファイルが構築サーバー上にない場合には、そのサーバー上では dmake ジョブの実行は許可されません。

環境変数 / マクロ

以下は、環境変数またはメークファイルのマクロとして定義することができます。

DMAKE_RCFILE

実行時構成ファイルを定義します。デフォルトの実行時構成ファイルは ~/.dmake.rc です。

DMAKE_GROUP

ジョブの分散先にする構築サーバーグループの名前を定義します。サーバーグループは実行時構成ファイルに定義されます。デフォルトのサーバーグループは、実行時構成ファイルの 1 番目のグループです。

DMAKE_MAX_JOBS

実行時構成ファイルの中で指定されている構築サーバーグループに分散される最大ジョブ数を定義します。デフォルトの最大ジョブ数は、構築サーバーグループに指定されるジョブの総数です。ジョブは、実行時構成ファイルの中で出現する順に 1 ずつホストから減算、またはホストに加算されます。例については、このマニュアルページの -j オプションを参照してください。

DMAKE_ADJUST_MAX_JOBS

次のいずれかのキーワードを含めることができます。

- YES: システムの現在の負荷に応じて、dmake が並列ジョブの制限を調整できるようにします。システムの負荷が過剰でない場合、dmake はユーザーが定義した制限を使用します。システムの負荷が過剰な場合、dmake は「現在の」制限を、ユーザーが定義した制限より下に設定します。

この変数が設定されない場合、dmake はシステムの現在の負荷に応じて並列ジョブの制限を調整します。これは dmake のデフォルトになります。

- NO: dmake の自動調整機構をオフに切り替えます。

DMAKE_MODE

次のいずれかのキーワードを含めることができます。

- serial: dmake を標準の make と同じように逐次動作させます。
- parallel: dmake がジョブを dmake ホストにだけ分散するようにします。この動作は、Sun WorkShop TeamWare 1.x の ParallelMake プログラムと同じです。
- distributed: dmake を完全な分散モードで動作させます。これは dmake のデフォルトです。
- grid: 構築ジョブの分散に SGE (Sun Grid Engine) を使います。

DMAKE_ODIR

dmake が一時出力ファイルを書き込む先、もしくは一時出力ファイルを読み取る元の共通の物理ディレクトリを定義します。この環境変数やマクロは、ローカルホスト上のホームディレクトリとすべてのリモートホスト上のホームディレクトリが同じ物理ホームディレクトリでない場合にだけ使用します。たとえば、スーパーユーザーがこのオプションを使用します。

DMAKE_OUTPUT_MODE

ログファイルの形式を定義します。次のキーワードのどれか 1 つを含むことができます。

- TEXT1: 構築ジョブを開始するたびに、dmake はシステム名とコマンド名をログファイルに出力します。

また、コマンドそのもの出力がある場合は、ジョブが終了するときに、そのコマンド出力とともに、再びシステム名とコマンド名をログファイルに出力します。

例:

```
host1 --> 1 個のジョブ
echo "Done host1"
host2 --> 1 個のジョブ
echo
"Done host2"
host1 --> ジョブ出力
echo "Done host1"
Done host1
host2 --> ジョブ出力
echo "Done host2"
Done host2
```

dmake のデフォルトはこの設定です。

- TEXT2: 並列ジョブの出力を直列化します。ログファイルが見やすくなります。このモードでは、dmake は、1つのジョブが終了すると、一度だけコマンドを出力し、その直後に、コマンド出力を続けます。

例:

```
echo "Done host1"  
Done host1  
echo "Done host2"  
Done host2
```

ファイル

- \$(HOME)/.dmakerc: デフォルトの実行時構成ファイル。構築サーバーと構築サーバーグループの名前が含まれます。
- /etc/opt/SUNWdmake/dmake.conf: 構築サーバー上にあるこのファイルは、すべての dmake ユーザーによって構築サーバーに分散することができる最大ジョブ総数を指定するために使用されます。また、すべての dmake ジョブに適用される /usr/bin/nice で指定する優先順位を指定するためにも使用されます。

関連項目

HTML ブラウザで以下の URL を指定して、完全な Sun Studio マニュアルセットにアクセスできます。

</opt/SUNWspro/docs/ja/index.html> (Solaris の場合)

</opt/sun/sunstudio10/docs/ja/index.html> (Linux の場合)

デフォルトの /opt 以外のディレクトリにコンパイラがインストールされている場合は、実際のパスをシステム管理者にお尋ねください。

更に詳しい情報は、以下のマニュアルページを参照してください。

- [make\(1\)](#)
- [rsh\(1\)](#)
- [hosts\(4\)](#)
- [hosts.equiv\(4\)](#)
- [attributes\(5\)](#)
- [largefile\(5\)](#)
- [qrsh\(1\)](#)

