

Oracle® Solaris Studio 12.4: 分散メイク (dmake)

ORACLE®

Part No: E57236
2014 年 10 月

Copyright © 2007, 2014, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ, AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

このドキュメントの使用方法	5
分散メイク (dmake)	7
形式	7
説明	7
オプション	9
-c <i>dmake_rcfile</i>	9
-g <i>dmake_group</i>	9
-j <i>dmake_max_jobs</i>	9
-m {serial parallel distributed grid}	10
-o <i>dmake_odir</i>	10
使用法	11
特殊な目的のターゲット	11
dmake ジョブの制御	11
dmake ホスト	11
ユーザーコマンド	13
構築サーバー	14
使用するリモートシェルを指定する	15
環境/マクロ	15
DMAKE_RCFILE	15
DMAKE_GROUP	15
DMAKE_MAX_JOBS	16
DMAKE_ADJUST_MAX_JOBS	16
DMAKE_MODE	16
DMAKE_ODIR	17
DMAKE_OUTPUT_MODE	17
ファイル	18
トラブルシューティング	18
関連項目	20

このドキュメントの使用方法

- 概要 - dmake(1) のマニュアルページと同じ内容について説明します
- 対象読者 - アプリケーション開発者、開発者、アーキテクト、サポートエンジニア
- 必要な知識 - プログラミングの経験、プログラム/ソフトウェア開発テスト、ソフトウェア製品の構築およびコンパイルの能力

製品ドキュメントライブラリ

この製品の最新情報や既知の問題は、ドキュメントライブラリ (http://docs.oracle.com/cd/E37069_01) に含まれています。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support にアクセスして電子サポートを受けることができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

分散メイク (dmake)

このマニュアルは、`dmake(1)` のマニュアルページと同じ内容を含んでいます。

- 7 ページの「形式」
- 7 ページの「説明」
- 9 ページの「オプション」
- 11 ページの「使用法」
- 15 ページの「使用するリモートシェルを指定する」
- 15 ページの「環境/マクロ」
- 18 ページの「ファイル」
- 18 ページの「トラブルシューティング」
- 20 ページの「関連項目」

形式

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

説明

このマニュアルページは Oracle Solaris `make` ユーティリティのマニュアルページを補足します。`dmake` ユーティリティは Linux オペレーティングシステム上で動作しますが、`dmake` オプションはほかの `make` ユーティリティとは無関係です。

分散メイク (`dmake`) はメイクファイルを解析し、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それらのターゲットの構築を分散します。

標準的な `make` ユーティリティーを Oracle Solaris オペレーティングシステム上で使用している場合は、`dmake` への切り替えに伴ってメイクファイルに変更を加える必要があるとしても、変更はわずかです。`dmake` は、Oracle Solaris `make` ユーティリティーの超集合で、Oracle Solaris および Linux の両方のプラットフォーム上で使用できます。`make` が入れ子になっていて、最上位のメイクファイルが「`make`」を呼び出す場合、`$(MAKE)` を使用する必要があります。

ユーザーは `dmake` を `dmake` ホスト上で実行し、ジョブを構築サーバーに分散します。

ジョブを `dmake` ホストに分散することもでき、その場合は `dmake` ホストも構築サーバーになります。`dmake` は、(メイクファイルに基づいて) 同時に構築可能であると `dmake` が判断したメイクファイルターゲットに基づいてジョブを分散します。

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、構築サーバー上でコマンドをリモート実行するためのパスワードを要求されることなく、`rsh` または `ssh` を使用できる必要があります。

次の例は、Oracle Solaris オペレーティングシステムが動作するマシン上で、`which` コマンドと一緒に `rsh` または `ssh` をテストとして使用した結果を示しています。

```
demo% rsh machine_name which dmake
/bin/dmake
```

```
demo% ssh machine_name which dmake
/bin/dmake
```

`rsh` または `ssh` の呼び出しは、追加の出力が返されないクリーンな状態である必要があります。

- `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できる必要があります。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` ホストと同じであると仮定します。この仮定は、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することによって無効にできます。
- ホスト上に `/etc/opt/SPROdmake/dmake.conf` ファイルが存在していて、読み取り可能であり、適切な情報が含まれている。このファイルが存在しない場合は、`dmake` はこのシステムでジョブを 1 つだけ分散します。

`dmake` ホストから、使用される構築サーバーや、各構築サーバーに割り当てられる `dmake` ジョブの数を制御できます。

特定の構築サーバー上で実行できる `dmake` ジョブの数も、そのサーバー上で制限できます。

オプション

このマニュアルページで以下に説明するこれらのオプション、環境変数、およびメイクファイルマクロは、同じ動作を変更します。

優先順位は次のように定義されます。

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

-c *dmake_rcfile*

代替の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 `$(HOME)/.dmakerc` 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内に定義します。

デフォルトのサーバーグループは、実行時構成ファイル内の最初のグループです。

-j *dmake_max_jobs*

実行時構成ファイル内で指定された構築サーバーのグループに対して分散される最大のジョブ総数を指定します。

ジョブの最大数のデフォルトは、構築サーバーグループ内で指定されたすべてのジョブの合計です。

ジョブは、ホストが実行時構成ファイル内で表示される順に、ホストから 1 つずつ減算または加算されます。

たとえば、実行時構成ファイル内で指定されるすべてのジョブの合計が、次のように 8 つだとします。

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

ここで、`dmake_max_jobs` が 11 に指定された場合、`dmake` は次のようにして、現在のジョブの総数の最大値 (8 つ) にジョブをあと 3 つ追加します。

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

また、`dmake_max_jobs` が 4 に指定された場合、`dmake` は次のようにして 4 つのジョブを (元の 8 から) 減算します。

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

-m {serial | parallel | distributed | grid}

次のいずれかのキーワードを指定します。

<code>serial</code>	<code>dmake</code> は <code>make</code> の標準の直列バージョンのように動作します。
<code>parallel</code>	<code>dmake</code> は <code>dmake</code> ホストにのみジョブを分散します。
<code>distributed</code>	<code>dmake</code> は完全な分散モードで動作します。これが <code>dmake</code> のデフォルトです。
<code>grid</code>	<code>dmake</code> は、Oracle Grid Engine (旧称 Solaris Grid Engine) を使用して構築ジョブを分散します。

-o *dmake_odir*

`dmake` が一時出力ファイルを書き込んだり、一時出力ファイルを読み取ったりできる共通の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、このディレクトリまたは指定されるディレクトリは、すべての構築サーバーから表示できる必要があります。

このオプションは、ローカルホスト上の `$(HOME)` ディレクトリと、すべてのリモートホスト上の `$(HOME)` ディレクトリが同一の物理 `$(HOME)` ディレクトリでない場合にのみ使用します。たとえば、`root` ユーザーがこのオプションを使用します。

使用法

特殊な目的のターゲット

dmake によって、いくつかの構築サーバー上にターゲットを同時に構築できます。並行処理によって、大規模なシステムまたはプロジェクトを構築するのに必要な時間が大幅に削減されます。dmake は、並行性とタイミングを制御するために次の特殊なメイクファイルターゲットを提供します。

<code>.NO_PARALLEL</code>	逐次に処理するターゲットを示すには、このターゲットを使用します。
<code>.PARALLEL</code>	並列で処理するターゲットを示すには、このターゲットを使用します。
<code>.LOCAL</code>	ローカルホスト上で逐次に処理するターゲットを示すには、このターゲットを使用します。
<code>.WAIT</code>	依存関係リストにこのターゲットを指定すると、処理が並列の場合であっても、dmake はターゲットより優先される依存関係が完了するまで待機したあと、後続のターゲットを処理します。

これらのターゲットを使用して記述したメイクファイルは、Oracle Solaris とともに配布される make の標準バージョンと互換性を保ちます。標準の make は、エラーを出さずに (かつ処理を行わずに) これらのターゲットを受け入れます。

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するマシンと、各構築サーバーに分散するジョブの数を指定できます。
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できる dmake ジョブの最大の総数を制御できます。

dmake ホスト

dmake は実行を開始すると、ジョブの分散先を知るために、実行時構成ファイルを検索します。

このファイルは通常、`dmake` ホストのホームディレクトリにあり、`.dmakerc` という名前が付いています。

`dmake` は次の場所の実行時構成ファイルを次の順序で検索します。

1. コマンド行で `-c` オプションを使用して指定したパス名
2. `DMAKE_RCFILE` メイクファイルマクロを使用して指定したパス名
3. `DMAKE_RCFILE` 環境変数を使用して指定したパス名
4. `$(HOME)/.dmakerc` 内

実行時構成ファイルが見つからない場合、`dmake` は並列モードに切り替わり、`dmake` ホストに 2 つのジョブ (デフォルト) を分散します。これは、`-j` オプションか、`DMAKE_MAX_JOBS` を使用して変更できます。

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するジョブの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サンプルです。

```
# My machine. This entry causes dmake to distribute to it
falcon { jobs = 1 }
hawk
eagle { jobs = 3 }
# Manager's machine. She's usually at meetings
heron { jobs = 4 }
avocet
```

エントリ `falcon`、`hawk`、`eagle`、`heron`、および `avocet` は、構築サーバーとしてリストされています。

各構築サーバーに分散するジョブの数を指定できます。ジョブのデフォルト数は 2 です。

"#" 文字で始まる任意の行は、コメントとして解釈されます。

注記 - 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指定することもできます。これを実行時構成ファイルに含めない場合、`dmake` ジョブはこれに対して分散されません。

また、実行時構成ファイル内に構築サーバーのグループを構築することもできます。これは、状況に応じて、構築サーバーの異なるグループを容易に切り替える柔軟性を提供します。たとえば、異なるオペレーティングシステムでのビルド用に異なる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できます。構築サーバーはすべて同じアーキテクチャーで、同じ OS がインストールされている必要があります。

次の実行時構成ファイルには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      { jobs = 3 }
    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

形式上のグループは `group` 指令で指定され、グループの構成要素の一覧は中括弧 (`{ }`) で区切られます。

グループの構成要素である構築サーバーは、オプションの `host` デイレクティブによって指定されます。

グループはほかのグループの構成要素となることができます。

個々の構築サーバーは、構築サーバーのグループも含んでいる実行時構成ファイル内にリストできます。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構成要素として扱います。

`dmake` は、次の一覧によって指定されるホストの単一グループに、1 から 4 の優先順位でジョブを分散させます。

1. コマンド行で `-g` オプションの引数として指定されたグループ。
2. `DMAKE_GROUP` メイクファイルマクロで指定されたグループ。
3. `DMAKE_GROUP` 環境変数で指定されたグループ。
4. 実行時構成ファイルにリストされている先頭の形式上のグループ。

実行時構成ファイル内で指定されるグループおよびホストの名前は、二重引用符で囲むことができます。これにより、グループおよびホストの名前の一部として表示される文字シーケンスに関して、高い柔軟性を得ることができます。たとえば、グループの名前が数字で始まる場合、次のように二重引用符で囲むようにしてください。

```
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` ディレクトリに構築サーバーからアクセスできる必要があります。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` ホストと同じであると仮定します。この仮定は、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することによって無効にできます。例:

```
group sparc-cluster {
    host wren   { jobs = 10 , path = "/opt/solarisstudio12.4/bin" }
    host stimp { path = "/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にあります。

このファイルを使用して次を指定します。

- 必須: その構築サーバー上で同時に実行できる (すべてのユーザーからの) `dmake` ジョブの最大の総数。
- オプション: すべての `dmake` ジョブが実行される優先順位。

次は、`dmake.conf` ファイルのサンプルです。

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

このファイルは、構築サーバー上で実行が許可される (すべての `dmake` ユーザーからの) `dmake` ジョブの最大数を 8 に設定します。実行されるジョブの優先順位は、`nice_prio` コマンドを使用すると変更できます。`nice(1)` を参照してください。

注記 - 構築サーバー上に `/etc/opt/SPROdmake/dmake.conf` ファイルが存在しない場合、そのサーバー上で `dmake` ジョブは実行を許可されません。

使用するリモートシェルを指定する

リモートシェルのパスは、`.dmakerc` ファイルに指定できます。

例:

```
host earth { jobs = 3 }
host mars  { jobs = 5 , rsh = "/bin/ssh" }
```

`rsh =` が指定されない場合、`dmake` はデフォルトで `/bin/rsh` を使用します。

`rsh` と同様に、`ssh` がパスワードを必要とせずにリモートホストにログインでき、警告またはエラーを出さないようにする必要があります。

環境/マクロ

環境変数またはメイクファイルマクロのいずれかとして次を定義できます。

DMAKE_RCFILE

代替の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 `$(HOME)/.dmakerc` です。

DMAKE_GROUP

ジョブの分散先とな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内で定義されます。デフォルトのサーバーグループは、実行時構成ファイル内の最初のグループです。

DMAKE_MAX_JOBS

実行時構成ファイル内で指定された構築サーバーのグループに対して分散される最大のジョブ総数を定義します。ジョブの最大数のデフォルトは、構築サーバーグループ内で指定されたすべてのジョブの合計です。ジョブは、ホストが実行時構成ファイル内で表示される順に、ホストから 1 つずつ減算または加算されます。

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むことができます。

- | | |
|-----|--|
| YES | <p><code>dmake</code> 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できます。システムが過負荷でない場合、<code>dmake</code> はユーザーによって定義された制限を使用します。システムが過負荷な場合、<code>dmake</code> は「現在の」制限を、ユーザーによって定義された制限よりも小さく設定します。</p> <p>この変数が設定されない場合、<code>dmake</code> 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。これが <code>dmake</code> のデフォルトです。</p> |
| NO | <p><code>dmake</code> の自動調整メカニズムをオフにします。</p> |

DMAKE_MODE

次のいずれかのキーワードを含むことができます。

- | | |
|-------------|--|
| serial | <p><code>dmake</code> は <code>make</code> の標準の直列バージョンのように動作します。</p> |
| parallel | <p><code>dmake</code> は <code>dmake</code> ホストにのみジョブを分散します。</p> |
| distributed | <p><code>dmake</code> は完全な分散モードで動作します。これが <code>dmake</code> のデフォルトです。</p> |
| grid | <p><code>dmake</code> は、Oracle Grid Engine (以前の Solaris Grid Engine) を使用して構築ジョブを分散します。</p> |

DMAKE_ODIR

`dmake` が一時出力ファイルを書き込んだり、一時出力ファイルを読み取ったりできる共通の物理ディレクトリを定義します。

この環境変数またはマクロは、ローカルホスト上の `$(HOME)` ディレクトリと、すべてのリモートホスト上の `$(HOME)` ディレクトリが同一の物理的 `$(HOME)` ディレクトリでない場合のみ使用します。たとえば、`root` 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。次のいずれかのキーワードを含むことができます。

TEXT1 各構築ジョブが開始されると、`dmake` はシステムとコマンドの名前をログファイルに出力します。

また、コマンドそれ自身から何らかの出力がある場合、ジョブが完了したとき、`dmake` はコマンド出力とともに、システムとコマンドの名前をふたたびログファイルに出力します。

例:

```
host1 --> 1 job
echo "Done host1"
host2 --> 1 job
echo
"Done host2"
host1 --> Job output
echo "Done host1"
Done host1
host2 --> Job output
echo "Done host2"
Done host2
```

これが `dmake` のデフォルトです。

TEXT2 `dmake` は並列ジョブの出力を直列化でき、ログファイルが読みやすくなります。このモードでは、各ジョブが完了すると、`dmake` はコマンドを 1 度だけ出力し、コマンド出力がその直後に続きます。

例:

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

ファイル

<code>\$(HOME)/.dmakerc</code>	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のグループの名前が含まれています。
<code>/etc/opt/ SPROdmake/ dmake.conf</code>	このファイルは構築サーバーに配置され、すべての <code>dmake</code> ユーザーによって構築サーバーに分散できるジョブの最大総数を指定するために使用されます。また、すべての <code>dmake</code> ジョブが実行される際の優先順位を指定するためにも使用されます。

トラブルシューティング

分散モードで `dmake` を使用した場合に何か問題が発生する場合は、次の点を確認してください。

- `$HOME` 環境変数がアクセス可能なディレクトリに設定されているか

```
% ls -la $HOME
```

- ファイル `$HOME/.dmakerc` が存在するか、このファイルの読み取りが可能か、このファイルの情報が正しいか

```
% cat $HOME/.dmakerc
```

- `$HOME/.dmakerc` ファイルに記載されているすべてのホストは活動中である必要があります。`/usr/sbin/ping` コマンドを使用して、各ホストを検査してください。

```
% /usr/sbin/ping $HOST
```

`$HOST` は、`$HOME/.dmakerc` ファイルでホストとして示されているシステムの名前です。

- `dmake` コマンド、`rxm` コマンド、および `rxs` コマンドを見つけるための `which` コマンドを使用して、`dmake` バイナリのパスが正しいことを確認します。

```
% which dmake
```

```
% which rxm
```

```
% which rxs
```

- 各ホストへのリモートログイン (rsh または ssh) がパスワードなしで動作し、各リモートログインの所要時間が許容範囲 (2 秒未満) である。

```
% time rsh $HOST uname -a
```

- 各ホスト上にファイル /etc/opt/SPROdmake/dmake.conf が存在するか。このファイル内の情報は正しいか。このファイルが存在しない場合は、dmake はこのシステムでジョブを 1 つだけ分散します。

```
% rsh $HOST cat /etc/opt/SPROdmake/dmake.conf
```

- 各ホストの dmake バイナリのパスが正しいか

```
% rsh $HOST `which dmake`
```

```
% rsh $HOST `which rxm`
```

```
% rsh $HOST `which rxs`
```

- 各ホストから構築領域を利用できるか (rwx):

```
% cd $BUILD
```

```
% rm $HOST.check.tmp
```

```
% echo "Build area is available from host $HOST" > $HOST.check.tmp
```

```
% rsh $HOST cat $BUILD/$HOST.check.tmp
```

\$BUILD には、構築領域のフルパスを指定してください。

- その \$HOME は各ホストから使用可能か

```
% cd $HOME
```

```
% rm $HOST.check.tmp
```

```
% echo "HOME is available from host $HOST" > $HOST.check.tmp
```

```
% rsh $HOST cat $HOME/$HOST.check.tmp
```

関連項目

Web 上の次のサイトにある Oracle Solaris Studio の完全なドキュメントを参照してください。<http://www.oracle.com/technetwork/server-storage/solarisstudio/documentation/index.html>

追加情報については、次のマニュアルページを参照してください。

- `make(1)`
- `rsh(1)`
- `ssh(1)`
- `hosts(4)`
- `hosts.equiv(4)`
- `attributes(5)`
- `largefile(5)`