

Oracle® Developer Studio 12.5 : 分布式创建 (dmake)

文件号码 E71987

版权所有 © 2007, 2016, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，则适用以下注意事项：

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.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。除非您与 Oracle 签订的相应协议另行规定，否则对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的保证，亦不对其承担任何责任。除非您和 Oracle 签订的相应协议另行规定，否则对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

文档可访问性

有关 Oracle 对可访问性的承诺，请访问 Oracle Accessibility Program 网站 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>。

获得 Oracle 支持

购买了支持服务的 Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

目录

使用本文档	7
分布式创建 (dmake)	9
用法概要	9
说明	9
选项	10
-c <i>dmake_rcfile</i>	10
-g <i>dmake_group</i>	11
-j <i>dmake_max_jobs</i>	11
-m {serial parallel distributed grid}	11
-o <i>dmake_odir</i>	12
用法	12
特殊用途目标	12
控制 dmake 作业	12
dmake 主机	13
用户命令	14
生成服务器	15
指定要使用的远程 Shell	15
环境/宏	16
DMAKE_RCFILE	16
DMAKE_GROUP	16
DMAKE_MAX_JOBS	16
DMAKE_ADJUST_MAX_JOBS	16
DMAKE_MODE	16
DMAKE_ODIR	17
DMAKE_OUTPUT_MODE	17
文件	18
疑难解答	18
另请参见	19

使用本文档

- 概述 – 介绍与 [dmake\(1\)](#) 手册页相同的内容
- 目标读者 – 应用程序开发人员、开发者、架构师、支持工程师
- 必备知识 – 编程经验、程序/软件开发测试、生成和编译软件产品的能力

产品文档库

有关该产品及相关产品的文档和资源，可从以下网址获得：http://docs.oracle.com/cd/E60778_01。

反馈

可以在 <http://www.oracle.com/goto/docfeedback>。

分布式创建 (dmake)

本手册包含与 [dmake\(1\)](#) 手册页相同的内容。

- “用法概要” [9]
- “说明” [9]
- “选项” [10]
- “用法” [12]
- “指定要使用的远程 Shell” [15]
- “环境/宏” [16]
- “文件” [18]
- “疑难解答” [18]
- “另请参见” [19]

用法概要

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

说明

本手册页是 Oracle Solaris `make` 实用程序手册页的补充。`dmake` 实用程序将在 Linux 操作系统下运行，但是 `dmake` 选项将独立于任何其他 `make` 实用程序。

分布式创建 (`dmake`) 解析您的 `makefile`，确定哪些目标可以并发生成，并将这些目标的生成作业分配到您设置的一定数目的主机中。

如果您在 Oracle Solaris 操作系统上使用的是标准 `make` 实用程序，则无需修改您的 `makefile` 文件或只需稍加修改就可以转变到使用 `dmake`。`dmake` 是 Oracle Solaris `make` 实用程序的超集，既可以在 Oracle Solaris 平台上使用，也可以在 Linux 平台上使用。对于嵌套的 `make`，如果顶层 `makefile` 调用 `"make"`，则需要使用 `$(MAKE)`。

您可以在一台 `dmake` 主机上执行 `dmake`，并将作业分配到多台生成服务器中。

您还可以将作业分配到 dmake 主机，在这种情况下，dmake 主机也成为生成服务器。dmake 根据 dmake（基于您的 makefile）所确定的可并发生成的 makefile 目标来分配作业。

您可以将任何计算机作为生成服务器，只要其符合以下要求：

- 在 dmake 主机（您正在使用的计算机）上，您必须能够在不提示输入口令的情况下使用 rsh 或 ssh，以便在生成服务器上远程执行命令。

以下示例显示了在运行 Oracle Solaris 操作系统的计算机上将 rsh 或 ssh 与 which 命令一起使用进行测试时的结果：

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

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

rsh 或 ssh 调用必须是清洁的，不返回任何其他输出。

- 必须能够从生成服务器访问安装了 dmake 软件的 bin 目录。缺省情况下，dmake 会假设生成服务器上 dmake 可执行文件的逻辑路径与 dmake 主机上的路径相同。可以通过在运行时配置文件中将路径名称指定为主机条目的属性来覆盖此假设。
- 文件 /etc/opt/SPROdmake/dmake.conf 位于主机上且可读，其中包含正确的信息。如果此文件不存在，dmake 将仅在此系统上分发一个作业。

从 dmake 主机上，您可以控制使用哪些生成服务器以及向每个生成服务器分配多少个 dmake 作业。

也可以在给定生成服务器上限制可以在该服务器上运行的 dmake 作业的数量。

选项

这些选项与本手册页后文中介绍的环境变量和 makefile 宏修改相同的行为。

它们的优先级顺序规定如下：

1. 命令行选项
2. Makefile 宏
3. 环境变量
4. dmake 缺省值

-c dmake_rcfile

指定替代运行时配置文件。

缺省运行时配置文件为 `$(HOME)/.dmake.rc`。

-g *dmake_group*

指定要将作业分配到其中的生成服务器组的名称。

服务器组是在运行时配置文件中定义的。

缺省服务器组是运行时配置文件中的第一个组。

-j *dmake_max_jobs*

在运行时配置文件中指定分配到指定的生成服务器组的最大作业总数。

缺省最大作业数是生成服务器组中指定的所有作业数的总和。

差额作业数将添加到主机或从主机中减去（按照主机在运行时配置文件中的出现顺序在原基础上每轮为每个主机加 1 或减 1）。

例如，如果在运行时配置文件中指定的所有作业总数为 8：

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

而 *dmake_max_jobs* 指定为 11，则 dmake 会将额外的三个作业加到当前的最大作业总数（八）上，如下所示：

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

另外，如果 *dmake_max_jobs* 指定为 4，则 dmake 会从原始八个作业中减去四个，如下所示：

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

-m {serial | parallel | distributed | grid}

指定下列关键字之一：

serial 导致 dmake 的行为与 make 的标准串行版本类似。

parallel 导致 dmake 仅向 dmake 主机分配作业。

<code>distributed</code>	导致 <code>dmake</code> 以完全分布模式运行。这是 <code>dmake</code> 缺省值。
<code>grid</code>	导致 <code>dmake</code> 使用 Oracle Grid Engine（以前称为 Solaris Grid Engine）来分配生成作业。

-o `dmake_odir`

指定 `dmake` 可将临时输出文件写入其中和从中读取临时输出文件的通用物理目录。使用的目录为 `$(HOME)/.dmake`，此目录或指定的任何目录必须对所有生成服务器可见。

仅当本地主机上的 `$(HOME)` 目录与所有远程主机上的 `$(HOME)` 目录不是同一物理 `$(HOME)` 目录时，才可使用该选项。例如，超级用户将使用该选项。

用法

特殊用途目标

`dmake` 允许在多个生成服务器上并发生成目标。并发处理可大大减少生成大型系统或项目所需的时间。`dmake` 提供了以下用于控制并发性和时间安排的特殊 `makefile` 目标。

<code>.NO_PARALLEL</code>	使用该目标指示哪些目标将串行处理。
<code>.PARALLEL</code>	使用该目标指示哪些目标将并行处理。
<code>.LOCAL</code>	使用该目标指示哪些目标将在本地主机上串行处理。
<code>.WAIT</code>	如果在相关项列表中指定了该目标，则 <code>dmake</code> 将等到在它之前的相关项完成后，才处理后面的那些相关项，即使处理是并行执行的也是如此。

您使用这些目标编写的 `Makefile` 仍然与随 Oracle Solaris 分发的 `make` 标准版本兼容。标准 `make` 可接受这些目标，且不会发生错误（也无需采取任何措施）。

控制 `dmake` 作业

`dmake` 作业的分配是以两种方式控制的：

1. `dmake` 主机上的 `dmake` 用户可以指定他们要用作生成服务器的计算机，以及他们要分配给每台生成服务器的作业数。

2. 生成服务器上的所有者（可以更改 `/etc/opt/SPROdmake/dmake.conf` 文件的用户）可以控制可分配给该生成服务器的 dmake 作业的最大总数。

dmake 主机

当 dmake 开始执行时，它将搜索运行时配置文件以了解要将作业分配到何处。

通常，该文件位于 dmake 主机上的起始目录中，名为 `.dmakerc`。

dmake 在以下位置按以下顺序搜索运行时配置文件：

1. 您使用 `-c` 选项在命令行上指定的路径名
2. 您使用 `DMAKE_RCFILE` makefile 宏指定的路径名
3. 您使用 `DMAKE_RCFILE` 环境变量指定的路径名
4. 在 `$(HOME)/.dmakerc` 中

如果未找到运行时配置文件，dmake 将切换到并行模式，并将两个作业（缺省）分配给 dmake 主机。可以使用 `-j` 选项或 `DMAKE_MAX_JOBS` 进行更改。

运行时配置文件可以包含生成服务器列表，以及您要分配给每台生成服务器的作业数。

下面是一个简单的运行时配置文件样例：

```
# 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
```

项 `falcon`、`hawk`、`eagle`、`heron` 和 `avocet` 作为生成服务器列出。

您可以指定要分配给每台生成服务器的作业数。缺省作业数为二。

以 `"#"` 字符开头的任何行都被解释为注释。

注 - 此生成服务器列表中包括了同时还是 dmake 主机的 `falcon`。dmake 主机也可以指定为生成服务器。如果您未将它包含在运行时配置文件中，则不会向它分配 dmake 作业。

也可在运行时配置文件中构造生成服务器组。这使得您可以根据情况灵活地在不同的生成服务器组之间轻松切换。例如，您可以针对不同操作系统下的生成定义一组不同的生成服务器，或定义安装有特殊软件的生成服务器组。这些生成服务器必须具有相同的体系结构，并且安装有相同的操作系统。

下面的运行时配置文件包含组：

```
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 }
}
```

用户命令

正规组是通过 group 指令指定的，其成员列表以大括号 ({ }) 为分界符。

作为组成员的生成服务器由可选的 host 指令指定。

组可以是其他组的成员。

单个的生成服务器可以列在也包含生成服务器组的运行时配置文件中。在这种情况下，dmake 将这些生成服务器视为未命名组的成员。

dmake 按从 1 到 4 的优先级顺序将作业分配给由以下列表指定的单个主机组。

1. 在命令行上指定为 -g 选项的参数的组。
2. 由 DMAKE_GROUP makefile 宏指定的组。
3. 由 DMAKE_GROUP 环境变量指定的组。
4. 在运行时配置文件中列出的第一个正规组。

在运行时配置文件中指定的组名和主机名可以括在双引号中。这在可以用作组名和主机名一部分的字符序列方面提供了更大的灵活性。例如，如果组名称以数字开头，则应使用双引号引起：

```
group "123_sparc"
```

如上所述，必须能够从生成服务器访问安装了 dmake 软件的 bin 目录。缺省情况下，dmake 会假设生成服务器上 dmake 可执行文件的逻辑路径与 dmake 主机上的路径相同。可以通过在运行时配置文件中将路径名称指定为主机条目的属性来覆盖此假设。例如：

```
group sparc-cluster {
    host wren    { jobs = 10 , path = "/opt/solarisstudio12.5/bin" }
    host stimp { path = "/bin" }
}
```

生成服务器

/etc/opt/SPROdmake/dmake.conf 文件位于生成服务器的文件系统中。

使用此文件指定以下内容：

- 必需：可在该生成服务器上并发运行的 dmake 作业（来自所有用户）的最大总数。
- 可选：所有 dmake 作业运行时都将采用的优先级。

下面是 dmake.conf 文件的样例：

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

此文件将允许在该生成服务器上运行的 dmake 作业（来自所有 dmake 用户）的最大数量设置为八个。您可以使用 nice_prio 命令更改将要运行的作业的优先级。请参见 nice (1)。

注 - 如果某个生成服务器上不存在 /etc/opt/SPROdmake/dmake.conf 文件，则不允许任何 dmake 作业在该服务器上运行。

指定要使用的远程 Shell

可以在 .dmakerc 文件中指定远程 shell 的路径。

例如：

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

如果未指定 rsh = ，则缺省情况下 dmake 将使用 /bin/rsh。

与使用 rsh 时一样，您必须确保 ssh 无需口令即可登录到远程主机，并且不发出任何警告或错误。

环境/宏

以下项可以定义为环境变量或 makefile 宏：

DMAKE_RCFILE

定义替代运行时配置文件。缺省运行时配置文件为 `$(HOME)/.dmakerc`。

DMAKE_GROUP

定义要将作业分发到其中的生成服务器组的名称。服务器组是在运行时配置文件中定义的。缺省服务器组是运行时配置文件中的第一个组。

DMAKE_MAX_JOBS

在运行时配置文件中定义分配到指定的生成服务器组的最大作业总数。缺省最大作业数是生成服务器组中指定的所有作业数的总和。差额作业数将添加到主机或从主机中减去（按照主机在运行时配置文件中的出现顺序在原基础上每轮为每个主机加 1 或减 1）。

有关示例，请参见本手册页中的 `-j` 选项。

DMAKE_ADJUST_MAX_JOBS

可以包含下列关键字之一：

- | | |
|---|--|
| 是 | 允许 dmake 根据系统的当前负载调整并行作业的限制。如果系统没有过载，dmake 将使用用户定义的限制。如果系统过载，dmake 会将“当前”限制设置为少于用户定义的限制。
如果未设置该变量，dmake 将根据系统的当前负载调整并行作业的限制。这是 dmake 缺省值。 |
| 否 | 导致 dmake 关闭自动调整机制。 |

DMAKE_MODE

可以包含下列关键字之一：

<code>serial</code>	导致 <code>dmake</code> 的行为与 <code>make</code> 的标准串行版本类似。
<code>parallel</code>	导致 <code>dmake</code> 仅向 <code>dmake</code> 主机分配作业。
<code>distributed</code>	导致 <code>dmake</code> 以完全分布模式运行。这是 <code>dmake</code> 缺省值。
<code>grid</code>	导致 <code>dmake</code> 使用 Oracle Grid Engine（以前称为 Solaris Grid Engine）来分配生成作业。

DMAKE_ODIR

定义 `dmake` 可将临时输出文件写入其中和从中读取临时输出文件的通用物理目录。

仅当本地主机上的 `$(HOME)` 目录与所有远程主机上的 `$(HOME)` 目录不是同一物理 `$(HOME)` 目录时，才可使用该环境变量或宏。例如，超级用户将使用该选项。

DMAKE_OUTPUT_MODE

定义日志文件的格式。可以包含下列关键字之一：

TEXT1	在启动每个生成作业时，<code>dmake</code> 将系统名称和命令输出到日志文件中。 此外，如果命令自身输出了任何内容，则当作业完成时，<code>dmake</code> 会再次将系统的名称和命令以及命令输出一一起输出到日志文件中。 示例： <pre>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</pre> 这是 <code>dmake</code> 缺省值。
TEXT2	允许 <code>dmake</code> 将并行作业的输出序列化，从而使日志文件的可读性更好。在该模式中，每个作业完成时，<code>dmake</code> 仅输出一一次命令（紧跟在命令输出之后）。 示例：

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

文件

<code>\$(HOME)/.dmakerc</code>	缺省运行时配置文件。包含生成服务器和生成服务器组的名称。
<code>/etc/opt/ SPROdmake/ dmake.conf</code>	此文件位于生成服务器上，用于指定所有 dmake 用户可以向其分配的最大作业总数。它还用于指定所有 dmake 作业运行时都将采用的优先级。

疑难解答

如果在分布式模式下使用 dmake 出现任何问题，请验证以下内容：

- `$HOME` 环境变量应设置为可访问的目录。

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

- 文件 `$HOME/.dmakerc` 存在且可读，并包含正确的信息。

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

- `$HOME/.dmakerc` 文件中提及的所有主机都处于活动状态。使用 `/usr/sbin/ping` 命令检查每台主机。

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

`$HOST` 是系统的名称，它作为主机列于 `$HOST` 文件中。

- 使用 `which` 命令查找 `dmake`、`rxm` 和 `rxs` 命令，从而验证 `dmake` 二进制文件的路径是否正确：

```
% which dmake
```

```
% which rxm
```

```
% which rxs
```

- 远程登录 (`rsh` 或 `ssh`) 每一台主机时不需要输入口令，并且每次远程登录所花费的时间处于可接受的范围（小于 2 秒）。

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

- 文件 `/etc/opt/SPROdmake/dmake.conf` 在每台主机中存在并包含正确的信息。如果此文件不存在，`dmake` 将仅在此系统上分配一个作业：

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

- 对于每台主机，`dmake` 二进制文件的路径是正确的：

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

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

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

- 可从每台主机获取生成区域 (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
```

其中，`$BUILD` 是生成区域的完整路径。

- 可从每台主机获取 `$HOME`：

```
% cd $HOME
```

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

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

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

另请参见

访问 Oracle Developer Studio Web 页上的完整 Oracle Developer Studio 文档，网址为 <http://www.oracle.com/technetwork/server-storage/solarisstudio/>。

有关更多信息，请查阅以下手册页。

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

