

Oracle® Solaris Studio 12.4: Distributed Make (dmake)



Part No: E37093
October 2014

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

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

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.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Copyright © 2007, 2014, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

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.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Contents

Using This Documentation	5
Distributed Make (dmake)	7
Synopsis	7
Description	7
Options	8
-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
Usage	10
Special-purpose Targets	10
Controlling dmake Jobs	11
The dmake Host	11
User Commands	13
The Build Server	13
Specifying Which Remote Shell To Use	14
Environment/Macros	14
DMAKE_RCFILE	14
DMAKE_GROUP	15
DMAKE_MAX_JOBS	15
DMAKE_ADJUST_MAX_JOBS	15
DMAKE_MODE	15
DMAKE_ODIR	16
DMAKE_OUTPUT_MODE	16
Files	17
Troubleshooting	17
See Also	18

Using This Documentation

- **Overview** – Describes the same content as the `dmake(1)` manual page
- **Audience** – Application developers, developer, architect, support engineer
- **Required knowledge** – Programming experience, Program/Software development testing, Aptitude to build and compile software products

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://docs.oracle.com/cd/E37069_01.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.

Distributed Make (dmake)

This manual contains the same content as the `dmake(1)` manual page.

- [“Synopsis” on page 7](#)
- [“Description” on page 7](#)
- [“Options” on page 8](#)
- [“Usage” on page 10](#)
- [“Specifying Which Remote Shell To Use” on page 14](#)
- [“Environment/Macros” on page 14](#)
- [“Files” on page 17](#)
- [“Troubleshooting” on page 17](#)
- [“See Also” on page 18](#)

Synopsis

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

Description

This man page supplements the Oracle Solaris `make` utility man page. The `dmake` utility will run under the Linux operating system, but the `dmake` options will be independent of any other `make` utility.

Distributed Make (`dmake`) parses your makefiles and determines which target can be built concurrently, and distributes the build of those targets over a number of hosts set by you.

If you use the standard `make` utility on the Oracle Solaris operating system, the transition to `dmake` requires little if any alteration to your makefiles. `dmake` is a superset of the Oracle Solaris `make` utility, which can be used both on Oracle Solaris and Linux platforms. With nested makes, if a top-level makefile calls `"make"`, you need to use `$(MAKE)`.

You execute `dmake` on a `dmake host` and distribute *jobs* to *build servers*.

You can also distribute jobs to the `dmake` host, in which case the `dmake` host also becomes a build server. `dmake` distributes jobs based on makefile targets that `dmake` determines (based on your makefiles) can be built concurrently.

You can use any machine as a build server as long as it meets the following requirements:

- From the `dmake` host (the machine you are using) you must be able to use `rsh` or `ssh` without being prompted for a password to remotely execute commands on the build server.

The following example shows the result of using `rsh` or `ssh` with the `which` command as a test on a machine running the Oracle Solaris operating system:

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

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

The `rsh` or `ssh` call must be clean, returning no additional output.

- The `bin` directory in which the `dmake` software is installed must be accessible from the build server. By default, `dmake` assumes that the logical path to the `dmake` executables on the build server is the same as the `dmake` host. This assumption can be overridden by specifying a path name as an attribute of the host entry in the runtime configuration file.
- The `/etc/opt/SPROdmake/dmake.conf` file exists on the host, is readable, and contains the correct information. If this file does not exist, `dmake` will distribute only one job on this system

From the `dmake` host you can control which build servers are used and how many `dmake` jobs are allotted to each build server.

The number of `dmake` jobs that can run on a given build server can also be limited on that server.

Options

These options and the environment variables and makefile macros described later in this man page modify the same behavior.

Their order of precedence is defined as:

1. Command-line options
2. Makefile macros
3. Environment variables

4. `dmake` default

-c *dmake_rcfile*

Specifies an alternate runtime configuration file.

The default runtime configuration file is `$(HOME)/.dmakerc`.

-g *dmake_group*

Specifies the name of the build server group to which jobs are distributed.

You define server groups in the runtime configuration file.

The default server group is the first group in the runtime configuration file.

-j *dmake_max_jobs*

Specifies the maximum total number of jobs that are distributed to the specified group of build servers in the runtime configuration file.

The default maximum number of jobs is the sum of all the specified jobs in a build server group.

The jobs are subtracted from, or added to, hosts by 1 in the order they appear in the runtime configuration file.

For example, if all jobs specified in the runtime configuration file total 8:

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

and *dmake_max_jobs* is specified as 11, `dmake` adds three more jobs to the current total maximum number of jobs (which is eight) as follows:

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

Also, if *dmake_max_jobs* is specified as 4, `dmake` subtracts four jobs (from the original eight) as follows:

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

-m {serial | parallel | distributed | grid}

Specify one of the following key words:

<code>serial</code>	Causes <code>dmake</code> to behave like the standard serial version of <code>make</code> .
<code>parallel</code>	Causes <code>dmake</code> to distribute jobs to only the <code>dmake</code> host.
<code>distributed</code>	Causes <code>dmake</code> to behave in fully distributed mode. This is the <code>dmake</code> default.
<code>grid</code>	Causes <code>dmake</code> to use Oracle Grid Engine (formerly known as Solaris Grid Engine) to distribute build jobs.

-o *dmake_odir*

Specifies a common physical directory that `dmake` can write temporary output files to and read temporary output files from. The directory used is `$(HOME)/.dmake` and this or whichever directory is specified, must be visible to all build servers.

Use this option only if the `$(HOME)` directory on your local host and the `$(HOME)` directory on all of your remote hosts are not the same physical `$(HOME)` directory. For example, a root user would use this option.

Usage

Special-purpose Targets

`dmake` allows targets to be built concurrently on a number of build servers. Concurrent processing can greatly reduce the time required to build a large system or project. `dmake` supplies the following special makefile targets for controlling concurrency and timing.

<code>.NO_PARALLEL</code>	Use this target to indicate which targets are to be processed serially.
<code>.PARALLEL</code>	Use this target to indicate which targets are to be processed in parallel.
<code>.LOCAL</code>	Use this target to indicate which targets are to be processed serially on the local host.

`.WAIT` When you specify this target in a dependency list, `dmake` waits until the dependencies that precede it are finished before processing those that follow, even when processing is parallel.

Makefiles that you write using these targets remain compatible with the standard version of `make` distributed with Oracle Solaris. Standard `make` accepts these targets without error (and without action).

Controlling `dmake` Jobs

The distribution of `dmake` jobs is controlled in two ways:

1. A `dmake` user on a `dmake` host can specify the machines they want to use as build servers and the number of jobs they want to distribute to each build server.
2. The owner (a user that can alter the `/etc/opt/SPROdmake/dmake.conf` file) on a build server can control the maximum total number of `dmake` jobs that can be distributed to that build server.

The `dmake` Host

When `dmake` begins execution it searches for a runtime configuration file to know where to distribute jobs.

Generally, this file is located in your home directory on the `dmake` host and is named `.dmakerc`.

`dmake` searches for the runtime configuration file in the following locations and in the following order:

1. The path name you specify on the command line using the `-c` option
2. The path name you specify using the `DMAKE_RCFILE` makefile macro
3. The path name you specify using the `DMAKE_RCFILE` environment variable
4. In `$(HOME)/.dmakerc`

If a runtime configuration file is not found, `dmake` switches to parallel mode and distributes two jobs (the default) to the `dmake` host. You can change this using the `-j` option, or `DMAKE_MAX_JOBS`.

The runtime configuration file may contain a list of build servers and the number of jobs you want distributed to each build server.

The following is a sample of a simple runtime configuration file:

```
# 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
```

The entries: `falcon`, `hawk`, `eagle`, `heron`, and `avocet` are listed as build servers.

You can specify the number of jobs you want distributed to each build server. The default number of jobs is two.

Any line that begins with the `"#"` character is interpreted as a comment.

Note - This list of build servers includes `falcon` which is also the `dmake` host. The `dmake` host can also be specified as a build server. If you do not include it in the runtime configuration file, no `dmake` jobs are distributed to it.

You can also construct groups of build servers in the runtime configuration file. This provides you with the flexibility of easily switching between different groups of build servers as circumstances warrant. For instance you may define a different group of build servers for builds under different operating systems, or define groups of build servers that have special software installed on them. The build servers must be all the same architecture and have the same OS installed.

The following runtime configuration file contains groups:

```
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 }  
}
```

User Commands

Formal groups are specified by the `group` directive and lists of their constituents are delimited by braces (`{ }`).

Build servers that are constituents of groups are specified by the optional `host` directive.

Groups can be constituents of other groups.

Individual build servers can be listed in runtime configuration files that also contain groups of build servers. In this case `dmake` treats these build servers as constituents of the unnamed group.

`dmake` distributes jobs to a single group of hosts specified by the following list and in precedence from 1 to 4.

1. The group specified on the command-line as an argument to the `-g` option.
2. The group specified by the `DMAKE_GROUP` makefile macro.
3. The group specified by the `DMAKE_GROUP` environment variable.
4. The first formal group listed in the runtime configuration file.

The names of groups and hosts specified in the runtime configuration file may be enclosed in double quotes. This is to allow more flexibility with respect to the character sequences that may appear as part of the group and host names. For example, if the name of the group starts with a digit it should be double-quoted:

```
group "123_sparc"
```

As mentioned above, the `bin` directory in which the `dmake` software is installed must be accessible from the build server. By default, `dmake` assumes that the logical path to the `dmake` executables on the build server is the same as the `dmake` host. This assumption can be overridden by specifying a path name as an attribute of the host entry in the runtime configuration file. For example:

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

The Build Server

The `/etc/opt/SPROdmake/dmake.conf` file is located in the file system of build servers.

Use this file to specify the following:

- **Required:** The maximum total number of `dmake` jobs (from all users) that can run concurrently on that build server.
- **Optional:** The priority under which all `dmake` jobs are to be run.

The following is a sample of a `dmake.conf` file:

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

This file sets the maximum number of `dmake` jobs permitted to run on that build server (from all `dmake` users) to be eight. You can change the priority of the jobs to be run by using the `nice_prio` command. See `nice(1)`.

Note - If the `/etc/opt/SPROdmake/dmake.conf` file does not exist on a build server, no `dmake` jobs will be allowed to run on that server.

Specifying Which Remote Shell To Use

The path to the remote shell can be specified in the `.dmakerc` file.

For example:

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

If `rsh =` is not specified, `dmake` will use `/bin/rsh` by default.

As with `rsh`, you must ensure that `ssh` can login to the remote host without requiring a password, and does not issue any warnings or errors.

Environment/Macros

The following can be defined as either environment variables or makefile macros:

DMAKE_RCFILE

Defines an alternate runtime configuration file. The default runtime configuration file is `$(HOME)/.dmakerc`.

DMAKE_GROUP

Defines the name of the build server group to which jobs are distributed. Server groups are defined in the runtime configuration file. The default server group is the first group in the runtime configuration file.

DMAKE_MAX_JOBS

Defines the maximum total number of jobs that are distributed to the specified group of build servers in the runtime configuration file. The default maximum number of jobs is the sum of all the specified jobs in a build server group. The jobs are subtracted from, or added to, hosts by 1 in the order they appear in the runtime configuration file.

See the `-j` option in this man page for an example.

DMAKE_ADJUST_MAX_JOBS

May contain one of the following key words:

YES	Allows <code>dmake</code> to adjust the limit of parallel jobs according to the current loading of the system. If the system is not overloaded, <code>dmake</code> will use the limit defined by the user. If the system is overloaded, <code>dmake</code> will set the "current" limit to less than the limit defined by the user. If this variable is not set, <code>dmake</code> will adjust the limit of parallel jobs according to the current loading of the system. This is the <code>dmake</code> default.
NO	Causes <code>dmake</code> to switch off the autoadjustment mechanism.

DMAKE_MODE

May contain one of the following key words:

<code>serial</code>	Causes <code>dmake</code> to behave like the standard serial version of <code>make</code> .
<code>parallel</code>	Causes <code>dmake</code> to distribute jobs to only the <code>dmake</code> host.
<code>distributed</code>	Causes <code>dmake</code> to behave in fully distributed mode. This is the <code>dmake</code> default.

`grid` Causes `dmake` to use Oracle Grid Engine (formerly Solaris Grid Engine) to distribute build jobs.

DMAKE_ODIR

Defines a common physical directory that `dmake` can write temporary output files to and read temporary output files from.

Use this environment variable, or macro, only if the `$(HOME)` directory on your local host and the `$(HOME)` directory on all of your remote hosts are not the same physical `$(HOME)` directory. For example, a root user would use this option.

DMAKE_OUTPUT_MODE

Defines the format of the log file. May contain one of the following keywords:

`TEXT1` On starting each build job, `dmake` prints the name of the system and command to the log file.
Also, if the command itself prints any output, then as the job finishes, `dmake` prints the name of the system and command to the log file again along with the command output.

Example:

```
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
```

This is the `dmake` default.

`TEXT2` Allows `dmake` to serialize the output of parallel jobs, which makes the log file more readable. In this mode, `dmake` just prints the command once, immediately followed by the command output, as each job finishes.

Example:

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



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

Files

<code>\$(HOME)/.dmake.rc</code>	The default runtime configuration file. Contains the names of build servers and groups of build servers.
<code>/etc/opt/ SPROdmake/ dmake.conf</code>	Located on build servers, this file is used to specify the maximum total number of jobs that can be distributed to it by all dmake users. It is also used to specify the priority all dmake jobs are to be run under.

Troubleshooting

If there are any problems with using dmake in distributed mode, verify the following:

- The `$HOME` environment variable is set to an accessible directory.

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

- The file `$(HOME)/.dmake.rc` exists, is readable, and contains correct information.

```
% cat $HOME/.dmake.rc
```

- All hosts mentioned in the `$(HOME)/.dmake.rc` file are alive. Use the `/usr/sbin/ping` command to check each host.

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

`$HOST` is the name of the system that is listed as the host in `$(HOME)/.dmake.rc` file.

- Verify that the path to the dmake binaries is correct by using the `which` command to find the `dmake`, `rxm`, and `rxs` commands:

```
% which dmake
```

```
% which rxm
```

```
% which rxs
```

- The remote login (`rsh` or `ssh`) on each host works without a password, and each remote login takes an acceptable time (less than 2 seconds).

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

- The file `/etc/opt/SPROdmake/dmake.conf` exists on each host and contains the correct information. If this file does not exist, `dmake` will distribute only one job on this system:

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

- The path to the `dmake` binaries is correct for each host:

```
% rsh $HOST `which dmake`  
% rsh $HOST `which rxm`  
% rsh $HOST `which rxs`
```

- The build area is available from each host (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
```

where `$BUILD` is the full path to the build area.

- That `$HOME` is available from each host:

```
% cd $HOME  
% rm $HOST.check.tmp  
% echo "HOME is available from host $HOST" > $HOST.check.tmp  
% rsh $HOST cat $HOME/$HOST.check.tmp
```

See Also

Access the complete Oracle Solaris Studio documentation on the web at <http://www.oracle.com/technetwork/server-storage/solarisstudio/documentation/index.html>.

Consult the following man pages for additional information.

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