

# Distributed Computing Toolbox Release Notes

---

These Release Notes describe Version 1.0.1 of the Distributed Computing Toolbox. The following topics are discussed in these notes:

- “Major Bug Fixes” on page 1-2
- “Upgrading from an Earlier Release” on page 1-3
- “Known Software and Documentation Problems” on page 1-4

The Distributed Computing Toolbox Release Notes also provide information about recent versions of the product.

- “Distributed Computing Toolbox 1.0 Release Notes” on page 2-1

## **Printing the Release Notes**

If you would like to print the Release Notes, you can link to a PDF version.

## **Product Support and News**

The Distributed Computing Toolbox Product Support Page on the MathWorks Web site provides important information in addition to what is in the documentation. Refer to this page periodically, as it could be updated with new information at any time. You can find the Distributed Computing Toolbox Product Support Page at

<http://www.mathworks.com/support/product/product.html?product=DM>



## Distributed Computing Toolbox 1.0.1 Release Notes

### 1

|  |     |
|--|-----|
| <b>Major Bug Fixes</b> .....                           | 1-2 |
| Using Multiple Network Interface Cards .....           | 1-2 |
| Checkpoint Directory Cleanup .....                     | 1-2 |
| Lookup Service Behind a Firewall .....                 | 1-2 |
| <br>   |     |
| <b>Upgrading from an Earlier Release</b> .....         | 1-3 |
| Upgrade Distributed Computing Products Together .....  | 1-3 |
| Upgrade License Files .....                            | 1-3 |
| -port Flag Not Supported .....                         | 1-3 |
| <br>   |     |
| <b>Known Software and Documentation Problems</b> ..... | 1-4 |
| All Platforms .....                                    | 1-4 |
| Windows Platform .....                                 | 1-5 |
| Unix/Macintosh Platform .....                          | 1-6 |
| Macintosh Platform .....                               | 1-6 |

## Distributed Computing Toolbox 1.0 Release Notes

### 2

|  |     |
|--|-----|
| <b>Introduction to the Distributed Computing Toolbox</b> ..... | 2-2 |
|--|-----|



# Distributed Computing Toolbox 1.0.1 Release Notes

---

|  |     |
|--|-----|
| <b>Major Bug Fixes</b> . . . . .                           | 1-2 |
| Using Multiple Network Interface Cards . . . . .           | 1-2 |
| Checkpoint Directory Cleanup . . . . .                     | 1-2 |
| Lookup Service Behind a Firewall . . . . .                 | 1-2 |
| <br>   |     |
| <b>Upgrading from an Earlier Release</b> . . . . .         | 1-3 |
| Upgrade Distributed Computing Products Together . . . . .  | 1-3 |
| Upgrade License Files . . . . .                            | 1-3 |
| -port Flag Not Supported . . . . .                         | 1-3 |
| <br>   |     |
| <b>Known Software and Documentation Problems</b> . . . . . | 1-4 |
| All Platforms . . . . .                                    | 1-4 |
| Windows Platform . . . . .                                 | 1-5 |
| Unix/Macintosh Platform . . . . .                          | 1-6 |
| Macintosh Platform . . . . .                               | 1-6 |

## **Major Bug Fixes**

### **Using Multiple Network Interface Cards**

Distributed Computing Toolbox and MATLAB Distributed Computing Engine entities will now work on machines with multiple network interface cards (NICs).

### **Checkpoint Directory Cleanup**

Job and task data were not being removed from the job manager checkpoint directory, even when jobs were destroyed. This resulted in a gradual slowdown in the job manager. Job and task data are now removed from the checkpoint directory when the job is destroyed.

### **Lookup Service Behind a Firewall**

A job manager's lookup service can now operate behind a firewall.

## Upgrading from an Earlier Release

This section describes the issues involved in upgrading from the Distributed Computing Toolbox 1.0 to Version 1.0.1.

### **Upgrade Distributed Computing Products Together**

When upgrading from Version 1.0 to Version 1.0.1, you must upgrade both the Distributed Computing Toolbox and the MATLAB Distributed Computing Engine. The toolbox and engine must be the same version to interact with each other.

### **Upgrade License Files**

Because of feature name changes, old license files do not work with the new versions of the Distributed Computing Toolbox and the MATLAB Distributed Computing Engine. When upgrading the products to Version 1.0.1, you must use updated license files.

### **-port Flag Not Supported**

The `-port` flag for the `startjobmanager` and `startworker` scripts is not supported in Version 1.0.1 of the MATLAB Distributed Computing Engine.

## Known Software and Documentation Problems

This section describes the known software and documentation problems in Version 1.0.1 of the Distributed Computing Toolbox and the MATLAB Distributed Computing Engine, sorted into the following categories:

- “All Platforms” on page 1-4
- “Windows Platform” on page 1-5
- “Unix/Macintosh Platform” on page 1-6
- “Macintosh Platform” on page 1-6

### All Platforms

#### Customizing `classpath.txt`

If you use a customized `classpath.txt` file, you should update it to include the latest `distcomp` entries provided with this release of the Distributed Computing Toolbox in `MATLABROOT/toolbox/local/classpath.txt`.

#### Job Manager Crashes

When a job manager crashes, all its jobs in the running state keep their `State` property set to `running` unless they have a `timeout` set. Tasks without `timeouts` that were running at the time of the crash keep their `State` property of value `running` when the job manager comes back up, but these tasks do not complete after the job manager recovers. Therefore, tasks that are running without `timeouts` at the time the job manager crashes, remain stuck in the `running` state after the job manager recovers.

When the job manager recovers, jobs and tasks in other states continue normally in the job manager and workers.

#### Recreating Job Manager Objects for Stopped Job Managers

If you stop a job manager and start it again, you must recreate objects in the client session that reference that job manager. Use `clear` classes to remove existing objects, and then use `findResource` to create them again.



## Using clear functions

Executing

```
clear functions
```

clears all Distributed Computing Toolbox objects from the current MATLAB session. They still remain in the job manager.

## Worker Log Files

A worker session overwrites its own log file (for example, in `C:\TEMP\MDCE\Log`) each time the worker session is reset by a job's `RestartWorker` property. As a workaround, you can access the same log information in the `mdce-service.log` file, which is not overwritten.

## Empty Values for the FileDependencies Property

You cannot set the value of a job's `FileDependencies` property to an empty value (`{}`) or an empty string (`{ ' ' }`).

## Warning Messages When Using FileDependencies Property

Use of the `FileDependencies` property may cause warning messages to be written to the log of an MDCE worker. This does not impact the execution of tasks by the worker.

## Windows Platform

### Windows Network Access for Workers

Because the worker is installed as a service running under `LocalSystem` by default, it does not have access to mapped network drives. You must use the full UNC path when installing the MDCE service if you want to run from a network share.

Note that if running MDCE under `LocalSystem` (the default), many systems do not allow access to UNC or mapped network shares. In this case, you must run MDCE under a different user with rights to log on as a service. See the section “Setting the User” in the network administration documentation.

## **MATLAB Worker on Windows 2000**

When the MDCE service runs as a service on a worker machine under Windows 2000, startworker might hang. To avoid this problem, you can run the MDCE service in console mode with the commands

```
cd MATLABROOT\toolbox\dml\bin\win32
mdce console
```

The window remains open, displaying MDCE log information. To end the MDCE session, enter ^C (**Ctrl+C**) in the window and reply with Y at the prompt to terminate the batch job.

## **Unix/Macintosh Platform**

### **Client Session on UNIX/Macintosh Requires X Server**

A UNIX system running the client session of the Distributed Computing Toolbox requires an X server for the event system to work. An X server is not necessary for any UNIX system that is running only worker or job manager sessions.

A Macintosh system running a client session of the Distributed Computing Toolbox requires an X server or the MATLAB session will hang. For example, you cannot run `matlab -nodisplay`.

## **Macintosh Platform**

### **Running a Worker on a Macintosh System as Root or from a Display**

Do not run the MDCE worker on a Macintosh system as root or from a display, as this will cause the worker session to hang when it executes a task.

### **Running a Worker and Client as the Same User on the Same Macintosh**

Running the MDCE worker and the DCT client as the same user on the same Macintosh will fail. The worker will hang when it begins to execute tasks.

### **Multicast Lookups on Macintosh**

Running a job manager and a client as different users on a single Macintosh will cause multicast lookups to fail.

## **waitForState and dfeval Functions Do Not Work on Macintosh**

The `waitForState` function waits indefinitely on the Macintosh platform. This will cause some demonstration programs to fail. As an alternative, try to use the other event mechanisms in your programs instead of `waitForState`.

Because the `dfeval` function uses `waitForState`, this also will not work on a Macintosh. As an alternative to `dfeval`, you can use `dfevalasync`.

Though not recommended, if necessary, you can circumvent this problem by using a loop. For example, replace the statement

```
waitForState(job, 'finished')
```

with this block of code:

```
while(1)
    pause(5);
    if strcmp(job.State,'finished')
        break;
    end
end
```

## **Running Toolbox and Engine Sessions on the Same Macintosh**

Because of the limitations posed by the issues on the Macintosh described in this section, it can be difficult to run the Distributed Computing Toolbox and MATLAB Distributed Computing Engine entities on a single Macintosh. You can do this with the following steps:

- 1** Log in remotely to the machine as a user other than `root` (for example, as `user1`) or run a shell for logging in as someone other than `root` with a command such as  

```
ssh user1@localhost
```
- 2** Start MDCE, job managers, and workers as this non-root user.
- 3** Log into the same machine locally as a user other than `user1` (for example, `user2`) by physically sitting at the computer or by using something like VNC. Open up an X11 terminal and start a MATLAB session that includes the Distributed Computing Toolbox.

- 4** Use a unicast to find the MATLAB Distributed Computing Engine entities from the toolbox session. Multicast will not work. For example, to find the job manager on the Macintosh that you are using, type  
`findResource('jobmanager', 'LookupURL', 'localhost')`
- 5** Proceed as usual, but do not use the `waitForState` function in your application.

# Distributed Computing Toolbox 1.0 Release Notes

---

**Introduction to the Distributed Computing Toolbox . . . 2-2**

## **Introduction to the Distributed Computing Toolbox**

The Distributed Computing Toolbox enables you to execute independent MATLAB operations distributed on a cluster of computers.

The goal of this first release of the Distributed Computing Toolbox is to address the requirements of coarse-grain applications, which execute the same MATLAB algorithm in remote MATLAB sessions on different data sets without communication or data exchange between sessions.