

# **Parallel Computing Toolbox™**

## **Release Notes**

---

## How to Contact The MathWorks



[www.mathworks.com](http://www.mathworks.com) Web  
[comp.soft-sys.matlab](mailto:comp.soft-sys.matlab) Newsgroup  
[www.mathworks.com/contact\\_TS.html](http://www.mathworks.com/contact_TS.html) Technical Support



[suggest@mathworks.com](mailto:suggest@mathworks.com) Product enhancement suggestions  
[bugs@mathworks.com](mailto:bugs@mathworks.com) Bug reports  
[doc@mathworks.com](mailto:doc@mathworks.com) Documentation error reports  
[service@mathworks.com](mailto:service@mathworks.com) Order status, license renewals, passcodes  
[info@mathworks.com](mailto:info@mathworks.com) Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.  
3 Apple Hill Drive  
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

*Parallel Computing Toolbox™ Release Notes*

© COPYRIGHT 2006–2008 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

### Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

### Patents

The MathWorks products are protected by one or more U.S. patents. Please see [www.mathworks.com/patents](http://www.mathworks.com/patents) for more information.

<b>Summary by Version</b> .....	<b>1</b>
<b>Version 4.0 (R2008b) Parallel Computing Toolbox Software</b> .....	<b>3</b>
<b>Version 3.3 (R2008a) Parallel Computing Toolbox Software</b> .....	<b>8</b>
<b>Version 3.2 (R2007b) Distributed Computing Toolbox Software</b> .....	<b>12</b>
<b>Version 3.1 (R2007a) Distributed Computing Toolbox Software</b> .....	<b>15</b>
<b>Version 3.0 (R2006b) Distributed Computing Toolbox Software</b> .....	<b>20</b>
<b>Compatibility Summary for Parallel Computing Toolbox Software</b> .....	<b>25</b>



## Summary by Version

This table provides quick access to what is new in each version. For clarification, see “Using Release Notes” on page 1.

<b>Version (Release)</b>	<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>	<b>Related Documentation at Web Site</b>
<b>Latest Version V4.0 (R2008b)</b>	Yes Details	Yes Summary	Bug Reports Includes fixes	Printable Release Notes: PDF  Current product documentation
V3.3 (R2008a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V3.2 (R2007b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V3.1 (R2007a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V3.0 (R2006b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No

### Using Release Notes

Use release notes when upgrading to a newer version to learn about:

- New features
- Changes
- Potential impact on your existing files and practices

Review the release notes for other MathWorks™ products required for this product (for example, MATLAB® or Simulink®) for enhancements, bugs, and compatibility considerations that also might impact you.

If you are upgrading from a software version other than the most recent one, review the release notes for all interim versions, not just for the version you

are installing. For example, when upgrading from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

## **What's in the Release Notes**

### **New Features and Changes**

- New functionality
- Changes to existing functionality

### **Version Compatibility Considerations**

When a new feature or change introduces a reported incompatibility between versions, the **Compatibility Considerations** subsection explains the impact.

Compatibility issues reported after the product is released appear under Bug Reports at the MathWorks Web site. Bug fixes can sometimes result in incompatibilities, so you should also review the fixed bugs in Bug Reports for any compatibility impact.

### **Fixed Bugs and Known Problems**

The MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time and as more information becomes available. This includes provisions for any known workarounds or file replacements. Information is available for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

## Version 4.0 (R2008b) Parallel Computing Toolbox Software

This table summarizes what is new in Version 4.0 (R2008b):

<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>	<b>Related Documentation at Web Site</b>
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.	Bug Reports Includes fixes	Printable Release Notes: PDF  Current product documentation

New features and changes introduced in this version are

- “MATLAB® Compiler Product Support for Parallel Computing Toolbox Applications” on page 3
- “SPMD Construct” on page 4
- “Composite Objects” on page 5
- “Configuration Validation” on page 5
- “Rerunning Failed Tasks” on page 5
- “Enhanced Job Control with Generic Scheduler Interface” on page 5
- “Changed Function Names for Codistributed Arrays” on page 6
- “Determining if a MATLAB Pool is Open” on page 6
- “Upgrade Parallel Computing Products Together” on page 7

### **MATLAB Compiler Product Support for Parallel Computing Toolbox Applications**

This release offers the ability to convert Parallel Computing Toolbox™ applications, using MATLAB® Compiler™, into executables and shared libraries that can access MATLAB® Distributed Computing Server™. For

information on this update to MATLAB Compiler, see “Applications Created with Parallel Computing Toolbox Now Able to Be Compiled”.

## Limitations

- MATLAB Compiler does not support configurations that use the local scheduler or local workers (i.e., workers that run locally on the desktop machine running the MATLAB client session).
- Compiled Parallel Computing Toolbox applications do not support Simulink software. For a list of other unsupported products, see the Web page [http://www.mathworks.com/products/ineligible\\_programs/](http://www.mathworks.com/products/ineligible_programs/).
- When workers are running a task from compiled code, they can execute only compiled code and toolbox code. They cannot execute functions contained in the current directory. Batch and MATLAB pool jobs attempt to change the worker working directory to the client working directory. When noncompiled files in the current directory conflict with compiled versions (for example, M-files with the same name), an error is thrown.

## SPMD Construct

A new single program multiple data (SPMD) language construct allows enhanced interleaving of serial and parallel programming, with interlab communication.

The general form of an `spmd` statement is:

```
spmd
    <statements>
end
```

The block of code represented by `<statements>` executes in parallel on workers in the MATLAB pool. Data on the labs is available for access from the client via Composite objects. For more information, see the `spmd` reference page and “Using SPMD and Composites”.



## Compatibility Considerations

Because `spmd` is a new keyword, it will conflict with any user-defined functions or variables of the same name. If you have any code with functions or variables named `spmd`, you must rename them.

## Composite Objects

Composite objects provide direct access from the client (desktop) program to data that is stored on labs in the MATLAB pool. The data of variables assigned inside an `spmd` block is available via Composites in the client. When a MATLAB pool is open, you can also create Composites directly from the client using the `Composite` function. See also “Using SPMD and Composites”.

## Configuration Validation

The Configurations Manager is enhanced with the capability for validating configurations. Open the Configurations Manager on the MATLAB Desktop by clicking **Parallel > Manage Configurations**. For more information, see “Validating Configurations”.

## Rerunning Failed Tasks

When using a job manager, if a task does not complete due to certain system failures, it can attempt to rerun up to a specified number of times. New properties of a task object to control reruns and access information about rerun attempts are:

- `MaximumNumberOfRetries`
- `AttemptedNumberOfRetries`
- `FailedAttemptInformation`

## Enhanced Job Control with Generic Scheduler Interface

The generic scheduler interface now allows you to cancel and destroy jobs and tasks and to investigate the state of a job. The following new properties of the generic scheduler object facilitate these features:

- `GetJobStateFcn`

- DestroyJobFcn
- DestroyTaskFcn
- CancelJobFcn
- CancelTaskFcn

New toolbox functions to accommodate this ability are:

- getJobSchedulerData
- setJobSchedulerData

For more information on this new functionality, see “Managing Jobs”.

## Changed Function Names for Codistributed Arrays

What was known in previous releases as distributed arrays are henceforth called *codistributed* arrays. Some functions related to constructing and accessing codistributed arrays have changed names in this release.

### Compatibility Considerations

The following table summarizes the changes in function names relating to codistributed arrays. The first three functions behave exactly the same with no change in operation, arguments, etc. The `isa` function takes the argument 'codistributed' in addition to the array in question.

Old Function Name	New Function Name
<code>distributed</code>	<code>codistributed</code>
<code>distributor</code>	<code>codistributor</code>
<code>dcolon</code>	<code>codcolon</code>
<code>isdistributed</code>	<code>isa, (X, 'codistributed')</code>

## Determining if a MATLAB Pool is Open

The function `matlabpool` now allows you to discover if a pool of workers is already open. The form of the command is:

`matlabpool size`

For more information about this option and others, see the `matlabpool` reference page.

## **Upgrade Parallel Computing Products Together**

This version of Parallel Computing Toolbox software is accompanied by a corresponding new version of MATLAB Distributed Computing Server software.

### **Compatibility Considerations**

As with every new release, you must upgrade both Parallel Computing Toolbox software and MATLAB Distributed Computing Server software together. These products must be the same version to interact properly with each other.

## Version 3.3 (R2008a) Parallel Computing Toolbox Software

This table summarizes what is new in Version 3.3 (R2008a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.	Bug Reports Includes fixes	No

New features and changes introduced in this version are

- “Renamed Functions for Product Name Changes” on page 8
- “New batch Function” on page 9
- “Enhanced Job Creation Functions” on page 9
- “Increased Data Size Transfers” on page 9
- “Changed Function Names for Distributed Arrays” on page 9
- “Support for PBS Pro and TORQUE Schedulers” on page 10
- “findResource Now Sets Properties According to Configuration” on page 10
- “parfor Syntax Has Single Usage” on page 11
- “dfeval Now Destroys Its Job When Finished” on page 11

### Renamed Functions for Product Name Changes

As of result of the product name changes, some function names are changing in this release.

### Compatibility Considerations

Two function names are changed to correspond to the new product names:

- `dctconfig` has been renamed `pctconfig`.
- `dctRunOnAll` has been renamed `pctRunOnAll`.

## **New batch Function**

The new `batch` function allows you to offload work from the client to one or more workers. The batch submission can run scripts that can include jobs that distribute work to other workers. For more information, see the `batch` reference page, and “Getting Started” in the Parallel Computing Toolbox User’s Guide.

## **New Matlabpool Job**

The batch functionality is implemented using the new `MATLABpool` job feature. A `MATLAB` pool job uses one worker to distribute a job to other workers, thereby freeing the client from the burden of tracking and job’s progress and manipulating data. For more information, see the `createMatlabPoolJob` reference page.

## **Enhanced Job Creation Functions**

The `createJob` and `createParallelJob` functions have been enhanced to run without requiring a scheduler object as an argument. This is also true for the new `createMatlabPoolJob` function. When a scheduler is not specified, the function uses the scheduler identified in the applicable parallel configuration. For details, see the reference page for each function.

## **Increased Data Size Transfers**

The default size limitation on data transfers between clients and workers has been significantly increased. In previous releases the default limitation imposed by the JVM memory allocation was approximately 50 MB. The new higher limits are approximately 600 MB on 32-bit systems and 2 GB on 64-bit systems. See “Object Data Size Limitations”.

## **Changed Function Names for Distributed Arrays**

Several functions related to distributed arrays have changed names in this release.

## Compatibility Considerations

The following table summarizes the changes in function names relating to distributed arrays.

Old Function Name	New Function Name
darray	distributed, distributor
distribute	distributed
dcolonpartition	defaultPartition
distribdim	distributionDimension
isdarray	isdistributed
labgrid	labGrid
local	localPart
partition	distributionPartition
localspan	globalIndices

## Support for PBS Pro and TORQUE Schedulers

Parallel Computing Toolbox software now fully supports PBS Pro® and TORQUE schedulers. These schedulers are integrated into parallel configurations and scheduler-related functions like `findResource`.

---

**Note** If you do not have a shared file system between client and cluster machines, or if you cannot submit jobs directly to the scheduler from the client machine, any use of third-party schedulers for parallel jobs (including `pmode`, `matlabpool`, and `parfor`) requires that you use the generic scheduler interface.

---

## findResource Now Sets Properties According to Configuration

The `findResource` function now sets the properties on the object it creates according to the configuration identified in the function call.

## Compatibility Considerations

In past releases, `findResource` could use a configuration to identify a scheduler, but did not apply the configuration settings to the scheduler object properties. If your code uses separate statements to find an object then set properties, this still works, but is not necessary any more.

## parfor Syntax Has Single Usage

The `parfor` statement is now recognized only for parallel for-loops, not for loops over a distributed range in parallel jobs.

## Compatibility Considerations

In R2007b, the pre-existing form of `parfor` was replaced by `for i = (drange)`, but both forms of syntax were recognized in that release. Now `parfor` has only one context, so `parfor` statements used in parallel jobs in code for versions prior to R2007a must be modified to use `for (drange)`.

## Limitations

**P-Code Scripts.** For this release, `parfor` is not supported in P-code script files.

**sim Inside parfor-Loops.** Running simulations in a `parfor`-loop with the `sim` command at the top level of the loop is not allowed. A `sim` command visible in a `parfor`-loop generates an error, although you can call `sim` inside a function that is called from the loop. Be sure that the various labs running simulations do not have the same working directory, as interference can occur with the simulation data.

## dfeval Now Destroys Its Job When Finished

When finished performing its distributed evaluation, the `dfeval` function now destroys the job it created.

## Compatibility Considerations

If you have any scripts that rely on a job and its data still existing after the completion of `dfeval`, or that destroy the job after `dfeval`, these scripts will no longer work.

## Version 3.2 (R2007b) Distributed Computing Toolbox Software

This table summarizes what is new in Version 3.2 (R2007b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.	Bug Reports Includes fixes	No

New features and changes introduced in this version are

- “New Parallel for-Loops (parfor-Loops)” on page 12
- “Configurations Manager and Dialogs” on page 13
- “Default Configuration” on page 14
- “Parallel Profiler” on page 14
- “MDCE Script for Red Hat Removed” on page 14

### New Parallel for-Loops (parfor-Loops)

New parallel for-loop (parfor-loop) functionality automatically executes a loop body in parallel on dynamically allocated cluster resources, allowing interleaved serial and parallel code. For details of new parfor functionality, see “Parallel for-Loops (parfor)” in the Distributed Computing Toolbox™ documentation.

### Limitations

**P-Code Scripts.** For this release, parfor is not supported in P-code script files.



## Compatibility Considerations

In past releases, `parfor` was a different function. The new `parfor` uses parentheses in defining its range to distinguish it from the old `parfor`.

New `parfor`:

```
parfor (ii = 1:N); <body of code>; end;
```

Old `parfor`:

```
parfor ii = 1:N; <body of code>; end;
```

For this release, the old form of `parfor` without parentheses is still supported, although it generates a warning. You can read more about the new form of this existing functionality in “Using a for-Loop Over a Distributed Range (for-drange)”. You should update your existing `parfor` code to use the new form of for-loops over a distributed range (`for-drange`), thus,

```
for ii = drange(1:N); <body of code>; end;
```

## Configurations Manager and Dialogs

This release introduces a new graphical user interface for creating and modifying user configurations, and for designating the default configuration used by some toolbox functions. For details about the configurations manager, see “Programming with User Configurations” in the Distributed Computing Toolbox documentation.

## Compatibility Considerations

This new feature has no impact on how configurations are used in a program, only on how configurations are created and shared among users. In previous versions of the product, you modified your configurations by editing the file `matlabroot/toolbox/distcomp/user/distcompUserConfig.m`. Now the configuration data is stored as part of your MATLAB software preferences.

The new configurations manager cannot directly import old-style configurations that were defined in the `distcompUserConfig.m` file. However, a utility called `importDistcompUserConfig`, available on the MATLAB Central Web site, allows you to convert and import your existing configurations into the new configurations manager.

Visit <http://www.mathworks.com/matlabcentral> and search for `importDistcompUserConfig`.

## Default Configuration

This version of the toolbox enables you to select a user configuration to use as the default. Thus, commands such as `pmode` and `matlabpool` will use the default configuration without your having to specify it each time you run the command. You can set the default configuration using the configurations graphical interface, or programmatically with the `defaultParallelConfig` function.

## Parallel Profiler

A new parallel profiler graphical user interface generates reports on lab computation and communication times during execution of parallel jobs. For details about this new feature, see “Using the Parallel Profiler”.

## MDCE Script for Red Hat Removed

The MDCE script `rh_mdce`, specific to Red Hat Linux®, has been removed from `matlabroot/toolbox/distcomp/util/bin`.

## Compatibility Considerations

If you make use of this script, you must replace it with its more generic equivalent,

`matlabroot/toolbox/distcomp/bin/mdce`.

## Version 3.1 (R2007a) Distributed Computing Toolbox Software

This table summarizes what is new in Version 3.1 (R2007a):

<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>	<b>Related Documentation at Web Site</b>
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.	Bug Reports Includes fixes	No

New features and changes introduced in this version are

- “Local Scheduler and Workers” on page 15
- “New pmode Interface” on page 16
- “New Default Scheduler for pmode” on page 16
- “Vectorized Task Creation” on page 16
- “Additional Submit and Decode Scripts” on page 17
- “Jobs Property of Job Manager Sorts Jobs by ID” on page 17
- “New Object Display Format” on page 18
- “Enhanced MATLAB Functions” on page 18
- “darray Function Replaces distributor Function” on page 18
- “rand Seeding Unique for Each Task or Lab” on page 19
- “Single-Threaded Computations on Workers” on page 19

### Local Scheduler and Workers

A local scheduler allows you to schedule jobs and run up to four workers or labs on a single MATLAB client machine without requiring engine licenses. These workers/labs can run distributed jobs or parallel jobs, including pmode

sessions, for all products for which the MATLAB client is licensed. This local scheduler and its workers do not require a job manager or third-party scheduler.

## **New pmode Interface**

The interactive parallel mode (pmode) has a new interface. The pmode command input and displays of the lab outputs are provided in a user interface that you can separate from the MATLAB client Command Window.

## **Compatibility Considerations**

In previous versions of Distributed Computing Toolbox, the pmode interface used the MATLAB Command Window, with the pmode input using a different prompt. The output from the labs was intermingled with the MATLAB client output.

## **New Default Scheduler for pmode**

If you start pmode without specifying a configuration,

```
pmode start
```

pmode automatically starts a parallel job using the local scheduler with labs running on the client machine. For more information about running pmode, see “Interactive Parallel Computation with pmode” in the Distributed Computing Toolbox documentation.

## **Compatibility Considerations**

In the previous version of the toolbox, when pmode was started without specifying a configuration, it searched the network for the first available job manager to use as a scheduler.

## **Vectorized Task Creation**

The `createTask` function can now create a vector of tasks in a single call when you provide a cell array of cell arrays for input arguments. For full details, see the `createTask` reference page.

## Compatibility Considerations

In previous versions of the distributed computing products, if your task function had an input argument that was a cell array of cell arrays, your code will need to be modified to run the same way in this release.

For example, your old code may have been written as follows so that the function `myfun` gets four cell array input arguments:

```
createTask(j, @myfun, 1, {{C1} {C2} {C3} {C4}})
```

In this new version, the same code will produce four tasks. To get the old functionality, you must wrap the four cell arrays in another cell array, so that `createTask` knows to create only one task.

```
createTask(j, @myfun, 1, { {{C1} {C2} {C3} {C4}} })
```

## Additional Submit and Decode Scripts

There are several submit and decode functions provided with the toolbox for your use with the generic scheduler interface. These files are in the directory

```
matlabroot/toolbox/distcomp/examples/integration
```

This version of the toolbox includes new subdirectories for Platform LSF® and PBS, to support network configurations in which the client and worker computers do not share a file system. For more information, see “Supplied Submit and Decode Functions” in the Distributed Computing Toolbox documentation.

## Jobs Property of Job Manager Sorts Jobs by ID

The `Jobs` property of a job manager object now contains the jobs in the order in which they were created, as indicated by the `ID` property of each job. Similarly, the `findJob` function returns jobs sequenced by their ID, unless otherwise specified. This change makes job manager behavior consistent with the behavior of third-party schedulers.

## Compatibility Considerations

In previous versions of the distributed computing products, when using a job manager, jobs were arranged in the `Jobs` property or by `findJob` according to the status of the job.

## New Object Display Format

When you create distributed computing objects (scheduler, job, or task) without a semicolon at the end of the command, the object information is displayed in a new format. This new format is also shown when you use the `display` function to view an object or simply type the object name at the command line.

## Compatibility Considerations

With this enhancement, the output format shown when creating an object has changed.

## Enhanced MATLAB Functions

Several MATLAB functions have been enhanced to work on distributed arrays:

- `cat`
- `find`
- `horzcat`
- `subsindex`
- `vertcat`

For a complete list of MATLAB functions that are enhanced to work on distributed arrays, see “Using MATLAB Functions on Codistributed Arrays” in the Distributed Computing Toolbox documentation.

## `darray` Function Replaces `distributor` Function

The function `darray` now defines how an array is distributed among the labs in a parallel job.

## Compatibility Considerations

In the previous version of the toolbox, the `distributor` function was used to define how an array was distributed. In many cases, you can replace a call to `distributor` with a call to `darray`. For example, if you used `distributor` without arguments as an input to an array constructor,

```
rand(m, n, distributor());
```

you can update the code to read,

```
rand(m, n, darray());
```

## **rand Seeding Unique for Each Task or Lab**

The random generator seed is now initialized based on the task ID for distributed jobs, or the `labindex` for parallel jobs (including `pmode`). This ensures that the set of random numbers generated for each task or lab within a job is unique, even when you have more than 82 tasks or labs.

## **Compatibility Considerations**

In the previous version of the distributed computing products, the `rand` function could by default generate the same set of numbers for some tasks or labs when these exceeded 82 for a job.

## **Single-Threaded Computations on Workers**

Despite the ability in MATLAB software to perform multithreaded computations on multiple-CPU machines, the workers and labs running distributed and parallel jobs perform only single-threaded computations, so that multiprocessor cluster machines can better accommodate multiple workers or labs.

## Version 3.0 (R2006b) Distributed Computing Toolbox Software

This table summarizes what is new in Version 3.0 (R2006b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.	Bug Reports Includes fixes	No

New features and changes introduced in this version are

- “Support for Windows Compute Cluster Server (CCS)” on page 20
- “Windows 64 Support” on page 21
- “Parallel Job Enhancements” on page 21
- “Distributed Arrays” on page 21
- “Interactive Parallel Mode (pmode)” on page 22
- “Moved MDCE Control Scripts” on page 22
- “rand Seeding Unique for Each Task or Lab” on page 23
- “Task ID Property Now Same as labindex” on page 24

### Support for Windows Compute Cluster Server (CCS)

Distributed Computing Toolbox software and MATLAB® Distributed Computing Engine™ software now let you program jobs and run them on a Windows® Compute Cluster Server. For information about programming in the toolbox to use Windows Compute Cluster Server (CCS) as your scheduler, see the `findResource` reference page, and see also “Find a CCS Scheduler”.



## Windows 64 Support

The distributed computing products now support Windows 64 (Win64) for both MATLAB client and MATLAB worker machines.

## Parallel Job Enhancements

### Parallel Jobs Support Any Scheduler

Support for parallel jobs now extends to any type of scheduler. In previous releases, only the MathWorks job manager and mpiexec scheduler object supported parallel jobs. You can now run parallel jobs on clusters scheduled by a job manager, Windows Compute Cluster Server (CCS), Platform LSF, mpiexec, or using the generic scheduler interface. For programming information, see “Programming Parallel Jobs”.

### New labSendReceive Function

The labSendReceive function is introduced in this release. This function performs the same things as both labSend and labReceive, but greatly reduces the risk of deadlock, because the send and receive happen simultaneously rather than by separate statements. For more information, see the labSendReceive reference page.

### Improved Error Detection

This release offers improved error detection for miscommunication between labs running parallel jobs. Most notable among the improvements are error detection of mismatched labSend and labReceive statements.

## Distributed Arrays

Distributed arrays are partitioned into segments, with each segment residing in the workspace of a different lab, so that each lab has its own array segment to work with. Reducing the size of the array that each lab has to store and process means a more efficient use of memory and faster processing, especially for large data sets. For more information, see “Working with Codistributed Arrays”.

There are many new and enhanced MATLAB functions to work with distributed arrays in parallel jobs. For a listing of these functions and their reference pages, see “Job Management”.

### **parfor: Parallel for-Loops**

Parallel for-loops let you run a for-loop across your labs simultaneously. For more information, see “Using a for-Loop Over a Distributed Range (for-drange)” or the `parfor` reference page.

### **Interactive Parallel Mode (pmode)**

The interactive parallel mode (`pmode`) lets you work interactively with a parallel job running simultaneously on a number of labs. Commands you type at the `pmode` command line are executed on all labs at the same time. Each lab executes the commands in its own workspace on its own local variables or segments of distributed arrays. For more information, see “Getting Started with `pmode`”.

### **Moved MDCE Control Scripts**

To provide greater consistency across all platforms, the MDCE control scripts for Windows have moved and those for UNIX® and Macintosh® have new names.

### **Compatibility Considerations**

**Windows Utilities Moved.** In previous versions of the distributed computing products, the MDCE utilities for Windows computers were located in

```
matlabroot\toolbox\distcomp\bin\win32
```

The utilities are now located in

```
matlabroot\toolbox\distcomp\bin
```

The files that have moved are

```
nodestatus  
mdce
```

```
startjobmanager
stopjobmanager
startworker
stopworker
mdce_def.bat
```

**UNIX and Macintosh Utilities Renamed.** In previous versions of the distributed computing products, the MDCE utilities for UNIX and Macintosh computers were called by

```
nodestatus.sh
startjobmanager.sh
stopjobmanager.sh
startworker.sh
stopworker.sh
```

You can now call these with the following commands:

```
nodestatus
startjobmanager
stopjobmanager
startworker
stopworker
```

---

**Note** For UNIX and Macintosh, `mdce` and `mdce_def.sh` have not been moved or renamed.

---

## rand Seeding Unique for Each Task or Lab

The random generator seed is now initialized based on the task ID for distributed jobs, or the `labindex` for parallel jobs (including `pmode`). This ensures that the random numbers generated for each task or lab are unique within a job.

## Compatibility Considerations

In previous versions of the distributed computing products, the `rand` function would by default generate the same set of numbers on each worker.

## **Task ID Property Now Same as labindex**

Although you create only one task for a parallel job, the system copies this task for each worker that runs the job. For example, if a parallel job runs on four workers (labs), the `Tasks` property of the job contains four task objects. The first task in the job's `Tasks` property corresponds to the task run by the lab whose `labindex` is 1, and so on, so that the ID property for the task object and `labindex` for the lab that ran that task have the same value. Therefore, the sequence of results returned by the `getAllOutputArguments` function corresponds to the value of `labindex` and to the order of tasks in the job's `Tasks` property.

## **Compatibility Considerations**

In past releases, there was no correlation between `labindex` and the task ID property.

## Compatibility Summary for Parallel Computing Toolbox Software

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided with the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
<b>Latest Version V4.0 (R2008b)</b>	See the <b>Compatibility Considerations</b> subheading for each of these new features or changes: <ul style="list-style-type: none"> <li>• “Changed Function Names for Codistributed Arrays” on page 6</li> <li>• “Upgrade Parallel Computing Products Together” on page 7</li> </ul>
V3.3 (R2008a)	See the <b>Compatibility Considerations</b> subheading for each of these new features or changes: <ul style="list-style-type: none"> <li>• “Renamed Functions for Product Name Changes” on page 8</li> <li>• “Changed Function Names for Distributed Arrays” on page 9</li> <li>• “findResource Now Sets Properties According to Configuration” on page 10</li> <li>• “parfor Syntax Has Single Usage” on page 11</li> <li>• “dfeval Now Destroys Its Job When Finished” on page 11</li> </ul>

<b>Version (Release)</b>	<b>New Features and Changes with Version Compatibility Impact</b>
V3.2 (R2007b)	<p>See the <b>Compatibility Considerations</b> subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> <li>• “New Parallel for-Loops (parfor-Loops)” on page 12</li> <li>• “Configurations Manager and Dialogs” on page 13</li> <li>• “MDCE Script for Red Hat Removed” on page 14</li> </ul>
V3.1 (R2007a)	<p>See the <b>Compatibility Considerations</b> subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> <li>• “New pmode Interface” on page 16</li> <li>• “New Default Scheduler for pmode” on page 16</li> <li>• “Vectorized Task Creation” on page 16</li> <li>• “Jobs Property of Job Manager Sorts Jobs by ID” on page 17</li> <li>• “New Object Display Format” on page 18</li> <li>• “rand Seeding Unique for Each Task or Lab” on page 19</li> <li>• “darray Function Replaces distributor Function” on page 18</li> </ul>
V3.0 (R2006b)	<p>See the <b>Compatibility Considerations</b> subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> <li>• “Moved MDCE Control Scripts” on page 22</li> <li>• “rand Seeding Unique for Each Task or Lab” on page 23</li> </ul>