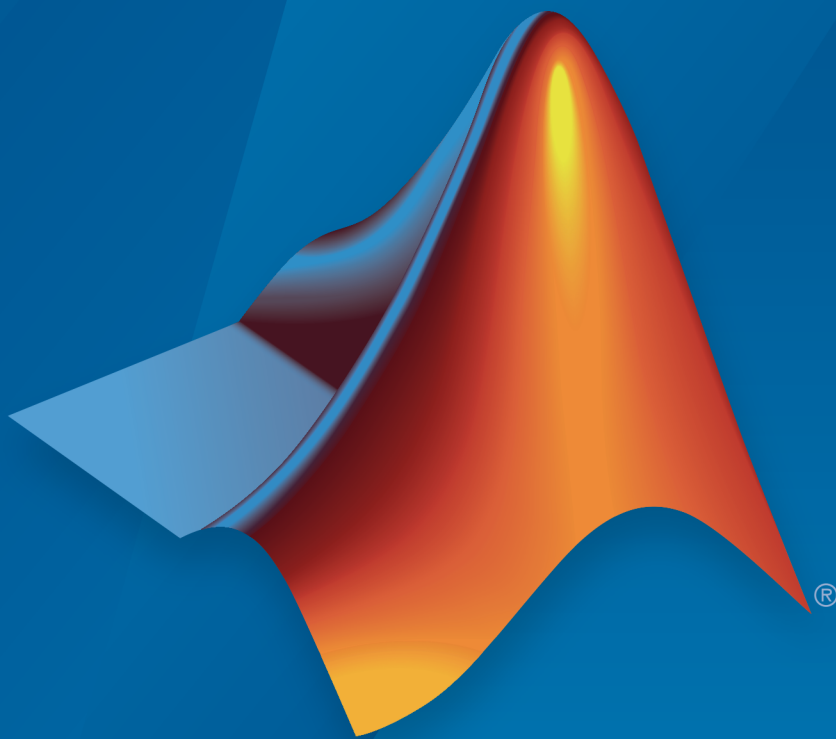


MATLAB[®] Distributed Computing Server[™] Release Notes



MATLAB[®]

How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



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

MATLAB® Distributed Computing Server™ Release Notes

© COPYRIGHT 2012–2015 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 for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

R2015b

Discontinued support for parallel computing products on 32-bit Windows operating systems	1-2
Scheduler integration scripts for SLURM	1-2
Improved performance of mapreduce on Hadoop 2 clusters	1-2
parallel.pool.Constant function to create constant data on parallel pool workers, accessible within parallel language constructs such as parfor and parfeval	1-3
Upgrade parallel computing products together	1-3

R2015a

Support for mapreduce function on any cluster that supports parallel pools	2-2
Using DNS for cluster discovery	2-2
MS-MPI support for MJS clusters	2-2
Ports and sockets in mdce_def file	2-2
Discontinued support for GPU devices on 32-bit Windows computers	2-3

Discontinued support for parallel computing products on 32-bit Windows computers	2-3
---	-----

R2014b

Data Analysis on Hadoop clusters using mapreduce	3-2
Additional MATLAB functions for distributed arrays, including <code>fft2</code>, <code>fftn</code>, <code>ifft2</code>, <code>ifftn</code>, <code>cummax</code>, <code>cummin</code>, and <code>diff</code>	3-2

R2014a

Duplication of an existing job, containing some or all of its tasks	4-2
More MATLAB functions enhanced for distributed arrays .	4-2
Old Programming Interface Removed	4-3
<code>matlabpool</code> Function Being Removed	4-3

R2013b

<code>parpool</code>: New command-line interface (replaces <code>matlabpool</code>), desktop indicator, and preferences for easier interaction with a parallel pool of MATLAB workers	5-2
Automatic start of a parallel pool when executing code that uses <code>parfor</code> or <code>spm</code>	5-2

Option to start a parallel pool without using MPI	5-3
More MATLAB functions enabled for distributed arrays: permute, ipermute, and sortrows	5-3
Upgraded MPICH2 Version	5-4
Discontinued Support for parallel.cluster.Mpiexec	5-4

R2013a

Automatic detection and transfer of files required for execution in both batch and interactive workflows	6-2
---	------------

R2012b

Automatic detection and selection of specific GPUs on a cluster node when multiple GPUs are available on the node	7-2
Detection of MATLAB Distributed Computing Server clusters that are available for connection from user desktops through Profile Manager	7-2

R2015b

Version: 6.7

New Features

Bug Fixes

Compatibility Considerations

Discontinued support for parallel computing products on 32-bit Windows operating systems

This release of MATLAB® products no longer supports 32-bit Parallel Computing Toolbox™ and MATLAB Distributed Computing Server™ on Windows® operating systems.

Compatibility Considerations

You can no longer install the parallel computing products on 32-bit Windows operating systems. If you must use Windows operating systems for the parallel computing products, upgrade to 64-bit MATLAB products on a 64-bit operating system.

Scheduler integration scripts for SLURM

This release offers a new set of scripts containing submit and decode functions to support Simple Linux® Utility for Resource Management (SLURM), using the generic scheduler interface. The pertinent code files are in the folder:

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

where *matlabroot* is your installation location.

The `slurm` folder contains a `README` file of instructions, and folders for `shared`, `nonshared`, and `remoteSubmission` network configurations.

For more information, view the files in the appropriate folders. See also “Program Independent Jobs for a Generic Scheduler” and “Program Communicating Jobs for a Generic Scheduler”.

Improved performance of mapreduce on Hadoop 2 clusters

The performance of `mapreduce` running on a Hadoop® 2.x cluster with MATLAB Distributed Computing Server is improved in this release for large input data.

parallel.pool.Constant function to create constant data on parallel pool workers, accessible within parallel language constructs such as parfor and parfeval

A new `parallel.pool.Constant` function allows you to define a constant whose value can be accessed by multiple `parfor`-loops or other parallel language constructs (e.g., `spmd` or `parfeval`) without the need to transfer the data multiple times.

For more information and examples, see `parallel.pool.Constant`.

Upgrade parallel computing products together

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

Compatibility Considerations

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

Jobs created in one version of Parallel Computing Toolbox software will not run in a different version of MATLAB Distributed Computing Server software, and might not be readable in different versions of the toolbox software. The job data stored in the folder identified by `JobStorageLocation` (formerly `DataLocation`) might not be compatible between different versions of MATLAB Distributed Computing Server. Therefore, `JobStorageLocation` should not be shared by parallel computing products running different versions, and each version on your cluster should have its own `JobStorageLocation`.

R2015a

Version: 6.6

New Features

Bug Fixes

Compatibility Considerations

Support for mapreduce function on any cluster that supports parallel pools

You can now run parallel mapreduce on any cluster that supports a parallel pool.

Using DNS for cluster discovery

In addition to multicast, the discover cluster functionality of Parallel Computing Toolbox can now use DNS to locate MATLAB job scheduler (MJS) clusters. For information about cluster discovery, see [Discover Clusters](#). For information about configuring and verifying the required DNS SRV record on your network, see [DNS SRV Record](#).

MS-MPI support for MJS clusters

On 64-bit Windows platforms, Microsoft[®] MPI (MS-MPI) is now the default MPI implementation for local clusters on the client machine.

For MATLAB job scheduler (MJS) clusters on Windows platforms, you can use MS-MPI by specifying the `-useMSMPI` flag with the `startjobmanager` command.

Ports and sockets in `mdce_def` file

The following parameters are new to the `mdce_def` file for controlling the behavior of MATLAB job scheduler (MJS) clusters.

- `ALL_SERVER_SOCKETS_IN_CLUSTER` — This parameter controls whether all client connections are outbound, or if inbound connections are also allowed.
- `JOBMANAGER_PEERSESSION_MIN_PORT`, `JOBMANAGER_PEERSESSION_MAX_PORT` — These parameters set the range of ports to use when `ALL_SERVER_SOCKETS_IN_CLUSTER = true`.
- `WORKER_PARALLELPPOOL_MIN_PORT`, `WORKER_PARALLELPPOOL_MAX_PORT` — These parameters set the range of ports to use on worker machines for parallel pools.

For more information and default settings for these parameters, see the appropriate `mdce_def` file for your platform:

- `matlabroot\toolbox\distcomp\bin\mdce_def.bat` (Windows)
- `matlabroot/toolbox/distcomp/bin/mdce_def.sh` (UNIX[®])

Compatibility Considerations

By default in this release, `ALL_SERVER_SOCKETS_IN_CLUSTER` is true, which makes all connections outbound from the client. For pre-R2015a behavior, set its value to `false`, which also initiates a set of inbound connections to the client from the MJS and workers.

Discontinued support for GPU devices on 32-bit Windows computers

This release no longer supports GPU devices on 32-bit Windows machines.

Compatibility Considerations

GPU devices on 32-bit Windows machines are not supported in this release. Instead, use GPU devices on 64-bit machines.

Discontinued support for parallel computing products on 32-bit Windows computers

In a future release, support will be removed for Parallel Computing Toolbox and MATLAB Distributed Computing Server on 32-bit Windows machines.

Compatibility Considerations

Parallel Computing Toolbox and MATLAB Distributed Computing Server are still supported on 32-bit Windows machines in this release, but parallel language commands can generate a warning. In a future release, support will be completely removed for these computers, at which time it will not be possible to install the parallel computing products on them.

R2014b

Version: 6.5

New Features

Bug Fixes

Data Analysis on Hadoop clusters using mapreduce

MATLAB Distributed Computing Server supports the use of Hadoop clusters for the execution environment of mapreduce applications. For more information, see:

- Configure a Hadoop Cluster
- Run mapreduce on a Hadoop Cluster

Additional MATLAB functions for distributed arrays, including `fft2`, `fftn`, `ifft2`, `ifftn`, `cummax`, `cummin`, and `diff`

The following functions now support distributed arrays with all forms of codistributor (1-D and 2DBC), or are enhanced in their support for this release:

<code>besselh</code>	<code>erf</code>	<code>isinteger</code>
<code>besseli</code>	<code>erfc</code>	<code>islogical</code>
<code>besselj</code>	<code>erfcinv</code>	<code>isnumeric</code>
<code>besselk</code>	<code>erfcx</code>	<code>median</code>
<code>bessely</code>	<code>erfinv</code>	<code>mode</code>
<code>beta</code>	<code>fft2</code>	<code>pol2cart</code>
<code>betainc</code>	<code>fftn</code>	<code>psi</code>
<code>betaincinv</code>	<code>gamma</code>	<code>rgb2hsv</code>
<code>betaln</code>	<code>gammainc</code>	<code>sph2cart</code>
<code>cart2pol</code>	<code>gammaincinv</code>	<code>std</code>
<code>cart2sph</code>	<code>gammaln</code>	<code>toeplitz</code>
<code>compan</code>	<code>hankel</code>	<code>trapz</code>
<code>corrcoef</code>	<code>hsv2rgb</code>	<code>unwrap</code>
<code>cov</code>	<code>ifft</code>	<code>vander</code>
<code>cummax</code>	<code>ifft2</code>	<code>var</code>
<code>cummin</code>	<code>ifftn</code>	
<code>diff</code>	<code>isfloat</code>	

For a list of MATLAB functions that support distributed arrays, see [MATLAB Functions on Distributed and Codistributed Arrays](#).

R2014a

Version: 6.4

New Features

Bug Fixes

Compatibility Considerations

Duplication of an existing job, containing some or all of its tasks

You can now duplicate job objects, allowing you to resubmit jobs that had finished or failed.

The syntax to duplicate a job is

```
newjob = recreate(oldjob)
```

where `oldjob` is an existing job object. The `newjob` object has all the same tasks and settable properties as `oldjob`, but receives a new ID. The old job can be in any state; the new job state is `pending`.

You can also specify which tasks from an existing independent job to include in the new job, based on the task IDs. For example:

```
newjob = recreate(oldjob, 'TaskID', [33:48]);
```

For more information, see the `recreate` reference page.

More MATLAB functions enhanced for distributed arrays

The following functions now support distributed arrays with all forms of codistributor (1-D and 2DBC), or are enhanced in their support for this release:

```
eye  
ifft  
randi  
rand  
randn
```

Note the following enhancements for some of these functions:

- `ifft` and `randi` are new in support of distributed and codistributed arrays.
- `rand(____, 'like', D)` returns a distributed or codistributed array of random values of the same underlying class as the distributed or codistributed array `D`. This enhancement also applies to `randi`, `randn`, and `eye`.

For a list of MATLAB functions that support distributed arrays, see [MATLAB Functions on Distributed and Codistributed Arrays](#).

Old Programming Interface Removed

The programming interface characterized by distributed jobs and parallel jobs has been removed. This old interface used functions such as `findResource`, `createParallelJob`, `getAllOutputArguments`, `dfeval`, etc.

Compatibility Considerations

The functions of the old programming interface now generate errors. You must migrate your code to the interface described in the R2012a Parallel Computing Toolbox release topic [New Programming Interface](#).

`matlabpool` Function Being Removed

The `matlabpool` function is being removed.

Compatibility Considerations

Calling `matlabpool` continues to work in this release, but now generates a warning. You should instead use `parpool` to create a parallel pool.

R2013b

Version: 6.3

New Features

Bug Fixes

Compatibility Considerations

parpool: New command-line interface (replaces matlabpool), desktop indicator, and preferences for easier interaction with a parallel pool of MATLAB workers

This release introduces a number of enhancements for interacting with parallel pool resources. For more detailed descriptions of these enhancements, see `parpool: New command-line interface (replaces matlabpool), desktop indicator, and preferences for easier interaction with a parallel pool of MATLAB workers` in the Parallel Computing Toolbox release notes.

- Parallel pool syntax replaces MATLAB pool syntax for executing parallel language constructs such as `parfor`, `spmd`, `Composite`, and `distributed`. The pool is represented in MATLAB by a `parallel.Pool` object.
- A new icon at the lower-left corner of the desktop indicates the current pool status. Icon color and tool tips let you know if the pool is busy or ready, how large it is, and when it might time out. You can click the icon to start a pool, stop a pool, or access your parallel preferences.
- Your MATLAB preferences now include a group of settings for parallel preferences. These settings control general behavior of clusters and parallel pools for your MATLAB session. You can access your parallel preferences in the MATLAB toolstrip, from the parallel pool status icon, or by typing `preferences` at the command line.

For more information, see [Parallel Preferences](#).

Compatibility Considerations

This release continues to support MATLAB pool language usage, but this support might discontinue in future releases. You should update your code as soon as possible to use parallel pool syntax instead.

Automatic start of a parallel pool when executing code that uses `parfor` or `spmd`

You can set your parallel preferences so that a parallel pool automatically starts whenever you execute a language construct that runs on a pool, such as `parfor`, `spmd`, `Composite`, `distributed`, `parfeval`, and `parfevalOnAll`.

Compatibility Considerations

The default preference setting is to automatically start a pool when a parallel language construct requires it. If you want to make sure a pool does not start automatically, you must change your parallel preference setting. You can also work around this by making sure to explicitly start a pool with `parpool` before encountering any code that needs a pool.

By default, a parallel pool will shut down *if idle for 30 minutes*. To prevent this, change the setting in your parallel preferences; or the pool indicator tool tip warns of an impending timeout and provides a link to extend it.

Option to start a parallel pool without using MPI

You now have the option to start a parallel pool on a local or MJS cluster so that the pool does not support running SPMD constructs. This allows the parallel pool to keep running even if one or more workers aborts during `parfor` execution. You explicitly disable SPMD support when starting the parallel pool by setting its `'SpmEnabled'` property `false` in the call to the `parpool` function. For example:

```
p = parpool('SpmEnabled',false);
```

Compatibility Considerations

Running any code (including MathWorks toolbox code) that uses SPMD constructs, on a parallel pool that was created without SPMD support, will generate errors.

More MATLAB functions enabled for distributed arrays: `permute`, `ipermute`, and `sortrows`

The following functions now support distributed arrays with all forms of codistributor (1-D and 2DBC), or are enhanced in their support for this release:

<code>ipermute</code>	<code>zeros</code>
<code>permute</code>	<code>ones</code>
<code>sortrows</code>	<code>nan</code>
	<code>inf</code>
<code>cast</code>	<code>true</code>
	<code>false</code>

For more information about these functions and distributed arrays, see [More MATLAB functions enabled for distributed arrays: permute, ipermute, and sortrows in the Parallel Computing Toolbox release notes](#).

Upgraded MPICH2 Version

The parallel computing products are now shipping MPICH2 version 1.4.1p1 on all platforms.

Compatibility Considerations

If you use your own MPI builds, you might need to create new builds compatible with this latest version, as described in [Use Different MPI Builds on UNIX Systems](#).

Discontinued Support for `parallel.cluster.Mpiexec`

Support for clusters of type `parallel.cluster.Mpiexec` is being discontinued.

Compatibility Considerations

In R2013b, any use of `parallel.cluster.Mpiexec` clusters generates a warning. In a future release, support might be completely removed.

R2013a

Version: 6.2

New Features

Bug Fixes

Automatic detection and transfer of files required for execution in both batch and interactive workflows

Parallel Computing Toolbox can now automatically attach files to a job so that workers have the necessary code files for evaluating tasks. When you set a job object's `AutoAttachFiles` to true, an analysis determines what files on the client machine are necessary for the evaluation of your job, and those files are automatically attached to the job and sent to the worker machines.

For more information, see *Automatic detection and transfer of files required for execution in both batch and interactive workflows* in the Parallel Computing Toolbox release notes.

R2012b

Version: 6.1

New Features

Bug Fixes

Automatic detection and selection of specific GPUs on a cluster node when multiple GPUs are available on the node

When multiple workers run on a single compute node with multiple GPU devices, the devices are automatically divided up among the workers. If there are more workers than GPU devices on the node, multiple workers share the same GPU device. If you put a GPU device in 'exclusive' mode, only one worker uses that device. As in previous releases, you can change the device used by any particular worker with the `gpuDevice` function.

Detection of MATLAB Distributed Computing Server clusters that are available for connection from user desktops through Profile Manager

You can let MATLAB discover clusters for you. Use either of the following techniques to discover those clusters which are available for you to use:

- On the **Home** tab in the **Environment** section, click **Parallel > Discover Clusters**.
- In the Cluster Profile Manager, click **Discover Clusters**.

For more information, see [Discover Clusters](#).