Distributed Computing Toolbox Release Notes

Contents

Summary by Version

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

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Latest Version V3.2 (R2007b)	Yes Details	Yes Summary	Bug Reports Includes fixes	Printable Release Notes: PDF Current product documentation
V3.1 (R2007a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V3.0 (R2006b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V2.0.1 (R2006a)	No	No	Bug Reports Includes fixes	No
V2.0 (R14SP3+)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V1.0.2 (R14SP3)	Yes Details	No	Bug Reports Includes fixes	No
V1.0.1 (R14SP2)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
New product V1.0 (R14SP1+)	No	No	No	No

About Release Notes

Use release notes when upgrading to a newer version to learn about new features and changes, and the potential impact on your existing files and practices. Release notes are also beneficial if you use or support multiple versions.

If you are not upgrading from the most recent previous version, review 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 New Features and Changes, Version Compatibility Considerations, and Bug Reports for V1.1 and V1.2.

New Features and Changes

These include

- New functionality
- Changes to existing functionality
- Changes to system requirements (complete system requirements for the current version are at the MathWorks Web site)
- Any version compatibility considerations associated with each new feature or change

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility with the previous version, its description includes a **Compatibility Considerations** subsection that details the impact. For a list of all new features and changes that have reported compatibility impact, see the "Compatibility Summary for Distributed Computing Toolbox" on page 25.

Compatibility issues that are reported after the product has been released are added to Bug Reports at the MathWorks Web site. Because bug fixes can sometimes result in incompatibilities, also review fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

MathWorks Bug Reports is a user-searchable database of known problems, workarounds, and fixes. The MathWorks updates the Bug Reports database as new problems and resolutions become known, so check it as needed for the latest information.

Access Bug Reports at the MathWorks Web site using your MathWorks Account. If you are not logged in to your MathWorks Account when you link to Bug Reports, you are prompted to log in or create an account. You then can view bug fixes and known problems for R14SP2 and more recent releases.

The Bug Reports database was introduced for R14SP2 and does not include information for prior releases. You can access a list of bug fixes made in prior versions via the links in the summary table.

Related Documentation at Web Site

Printable Release Notes (PDF). You can print release notes from the PDF version, located at the MathWorks Web site. The PDF version does not support links to other documents or to the Web site, such as to Bug Reports. Use the browser-based version of release notes for access to all information.

Product Documentation. At the MathWorks Web site, you can access complete product documentation for the current version and some previous versions, as noted in the summary table.

Version 3.2 (R2007b) Distributed Computing Toolbox

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 **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes	Printable Release Notes: PDF Current product documentation

New features and changes introduced in this version are

- "New Parallel for-Loops (parfor-Loops)" on page 4
- "Configurations Manager and Dialogs" on page 5
- "Default Configuration" on page 6
- "Parallel Profiler" on page 6
- "MDCE Script for Red Hat Removed" on page 6
- "Upgrade Distributed Computing Products Together" on page 6

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® 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`.

Upgrade Distributed Computing Products Together

This version of Distributed Computing Toolbox is accompanied by a corresponding new version of MATLAB Distributed Computing Engine.

Compatibility Considerations

As with every new release, you must upgrade both Distributed Computing Toolbox and MATLAB Distributed Computing Engine. The toolbox and engine must be the same version to interact with each other.

Version 3.1 (R2007a) Distributed Computing Toolbox

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

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

New features and changes introduced in this version are

- "Local Scheduler and Workers" on page 7
- "New pmode Interface" on page 8
- "New Default Scheduler for pmode" on page 8
- "Vectorized Task Creation" on page 8
- "Additional Submit and Decode Scripts" on page 9
- "Jobs Property of Job Manager Sorts Jobs by ID" on page 9
- "New Object Display Format" on page 10
- "Enhanced MATLAB Functions" on page 10
- "darray Function Replaces distributor Function" on page 10
- "rand Seeding Unique for Each Task or Lab" on page 11
- "Single-Threaded Computations on Workers" on page 11

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 Mode (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 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 Distributed 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. For details of this function, see the `darray` reference page.

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 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

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 **Compatibility Considerations**, 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 12
- "Windows 64 Support" on page 13
- "Parallel Job Enhancements" on page 13
- "Distributed Arrays" on page 13
- "Interactive Parallel Mode (pmode)" on page 14
- "Moved MDCE Control Scripts" on page 14
- "rand Seeding Unique for Each Task or Lab" on page 15
- "Task ID Property Now Same as labindex" on page 15

Support for Windows Compute Cluster Server (CCS)

Distributed Computing Toolbox and MATLAB Distributed Computing Engine now let you program jobs and run them on a Windows Compute Cluster Server. For information about programming in the toolbox to use 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, CCS, 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 Distributed 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 “Toolbox Functions Used in Parallel Jobs and pmode”.

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 parallel mode (pmode) of MATLAB 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 Interactive Parallel Mode".

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.

Version 2.0.1 (R2006a) Distributed Computing Toolbox

This table summarizes what is new in Version 2.0.1 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
No	No	Bug Reports Includes fixes	No

Version 2.0 (R14SP3+) Distributed Computing Toolbox

This table summarizes what is new in Version 2.0 (R14SP3+):

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

New features and changes introduced in this version are

- “Third-Party Schedulers” on page 18
- “Parallel Jobs” on page 18
- “Adding Tasks to Running Jobs” on page 19
- “Engine Process Enhancements” on page 19
- “ErrorIdentifier Changed” on page 21
- “Checkpoint Directory Structure Change” on page 21

Third-Party Schedulers

Distributed Computing Toolbox now supports the use of third-party schedulers as an alternative to the MathWorks job manager for running jobs and distributing tasks to the cluster.

Parallel Jobs

Distributed Computing Toolbox and MATLAB Distributed Computing Engine now support parallel jobs. In a parallel job, the tasks running simultaneously on separate machines are able to communicate with each other during their execution.

Adding Tasks to Running Jobs

If you use the MathWorks job manager to run your jobs, you can now add tasks to running jobs. Typically, you would do this inside your task function. After evaluating some data derived during execution of a task, the task can create more tasks based on that data. Because that task is not complete, its job is still running. The task can access its parent job object with the `getCurrentJob` function.

Engine Process Enhancements

Unicast Communications Between Processes

MDCE processes can now use unicast for all their communications. See the `MDCE_DEF` file for information about setting up this configuration, or see the help for the start scripts; for example,

```
startjobmanager -help
startworker -help
```

Remote Startup of MDCE Processes

Once the mdce service or daemon is running on all your cluster nodes, you can remotely stop and start worker and job manager sessions on those nodes. Use the `-remotehost` flag on the scripts that start and stop job managers and workers. For more information, use the `-help` flag with the process start and stop scripts, for example,

```
startjobmanager -help
stopjobmanager -help
startworker -help
stopworker -help
```

Using Mapped Drives

You can now start and stop the mdce service, job managers, and workers from a mapped drive location. Note that because the worker is installed as a service running under `LocalSystem` by default, it does not have access to mapped network drives.

Monitoring Processes

You can use the nodestatus command to display the status of the mdce service and the job manager and worker processes that it maintains. You can use nodestatus to monitor the local host or any remote host. For example,

```
nodestatus -remotehost node_XYZ
Job manager lookup process:
     Status                          Running

Job manager:
     Name                            OurJobManager
     Running on host                 node_XYZ
     Number of workers               2

Worker:
     Name                            node_XYZ_worker
     Running on host                 node_XYZ
     Status                          Idle
     Job manager                     OurJobManager
     Connection with job manager     Connected

Worker:
     Name                            worker2
     Running on host                 node_XYZ
     Status                          Idle
     Job manager                     OurJobManager
     Connection with job manager     Connected

Summary:
The mdce service on node_XYZmanages the following processes:
     Job manager lookup processes  1
     Job managers                    1
     Workers                         2
```

For a complete list of options, use the -help flag with nodestatus, for example,

```
nodestatus -help
```

ErrorIdentifier Changed

The `ErrorIdentifier` string from a canceled task has changed.

Compatibility Considerations

A canceled task now returns the `ErrorIdentifier`

```
distcomp:task:Cancelled
```

In former releases, it returned

```
dce:task:cancelled
```

Checkpoint Directory Structure Change

This release includes a change to the checkpoint directory structure.

Compatibility Considerations

Because of a change in the structure of the checkpoint directories, the processes (e.g., the job manager) of MATLAB Distributed Computing Engine Version 2 cannot read the checkpoint directories created by the processes of earlier versions.

Version 1.0.2 (R14SP3) Distributed Computing Toolbox

This table summarizes what is new in Version 1.0.2 (R14SP3):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports Includes fixes	No

New features and changes introduced in this version are

Rapid Simulation (RSim) Target

The Rapid Simulation (RSim) target has been enhanced such that RSim executables do not check out a Simulink® license when run by a worker executing a task created by Distributed Computing Toolbox.

Version 1.0.1 (R14SP2) Distributed Computing Toolbox

This table summarizes what is new in Version 1.0.1 (R14SP2):

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

New features and changes introduced in this version are

- "License Files Upgraded" on page 23
- "-port Flag Not Supported" on page 23

License Files Upgraded

New license files for this release support certain feature name changes.

Compatibility Considerations

Because of feature name changes, old license files do not work with this version of Distributed Computing Toolbox and MATLAB Distributed Computing Engine. When upgrading the products from Version 1.0 to Version 1.0.1, you must use updated license files.

-port Flag Not Supported

Control of the ports used by the `startjobmanager` and `startworker` scripts has moved into the `mdce_def` file.

Compatibility Considerations

The `-port` flag for the `startjobmanager` and `startworker` scripts is not supported after Version 1.0 of MATLAB Distributed Computing Engine.

Version 1.0 (R14SP1+) Distributed Computing Toolbox

This table summarizes what is new in Version 1.0 (R14SP1+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
New product	No	No	No

Compatibility Summary for Distributed Computing Toolbox

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
Latest Version V3.2 (R2007b)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "New Parallel for-Loops (parfor-Loops)" on page 4 • "Configurations Manager and Dialogs" on page 5 • "MDCE Script for Red Hat Removed" on page 6 • "Upgrade Distributed Computing Products Together" on page 6

Version (Release)	New Features and Changes with Version Compatibility Impact
V3.1 (R2007a)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "New pmode Interface" on page 8 • "New Default Scheduler for pmode" on page 8 • "Vectorized Task Creation" on page 8 • "Jobs Property of Job Manager Sorts Jobs by ID" on page 9 • "New Object Display Format" on page 10 • "rand Seeding Unique for Each Task or Lab" on page 11 • "darray Function Replaces distributor Function" on page 10
V3.0 (R2006b)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Moved MDCE Control Scripts" on page 14 • "rand Seeding Unique for Each Task or Lab" on page 15
V2.0.1 (R2006a)	None
V2.0 (R14SP3+)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "ErrorIdentifier Changed" on page 21 • "Checkpoint Directory Structure Change" on page 21
V1.0.2 (R14SP3)	None

Version (Release)	New Features and Changes with Version Compatibility Impact
V1.0.1 (R14SP2)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "License Files Upgraded" on page 23 • "-port Flag Not Supported" on page 23
V1.0 (R14SP1+)	None