

Database Toolbox™ 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

Database Toolbox™ Release Notes

© COPYRIGHT 2004–2017 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.

R2017a

One-step data import	1-2
Expanded data type support	1-2
Connection object and database connection changes	1-2
Functionality being removed or changed	1-3

R2016b

Graph Database Interface: Retrieve graph data from Neo4j	2-2
DatabaseDatastore functionality and object properties changes	2-2
Functionality being removed or changed	2-3

R2016a

MATLAB Interface to SQLite: Create, read, and write data from SQLite database files without external drivers and administration	3-2
--	-----

fetch Function Speed Improvement: Import data faster using the JDBC driver	3-2
Support for 32-bit Windows removed	3-2
Functionality being removed or changed	3-3

R2015b

ODBC Interface Functions: Export and retrieve database information using native ODBC connections	4-2
Read and Write Performance Improvements: Import and export data more quickly	4-2
Data Export Functions: Insert or replace data using table, structure, and dataset arrays	4-2
Functionality being removed or changed	4-2

R2015a

Bug Fixes

R2014b

DatabaseDatastore for applying mapreduce to data contained in relational databases	6-2
Scrollable cursors for accessing data using relative and absolute position inputs	6-2

R2014a

Bug Fixes

R2013b

Fast access to ODBC connections via a native ODBC driver	8-2
table data type support	8-2

R2013a

fetch function accepts user-defined batch sizes	9-2
--	-----

R2012b

Database Explorer app for interactively exchanging data with databases	10-2
Functionality Being Removed or Changed	10-2

R2012a

Execute .SQL Files	11-2
--------------------------	------

Improvements to the Database Constructor	11-2
---	-------------

R2011b

Preferences Now Persistent Across MATLAB Sessions	12-2
Change in Behavior for the update Function	12-2
Warning and Error ID Changes	12-2

R2011a

New datainsert Function Exports MATLAB Cell Array Data into a Database Table	13-2
---	-------------

R2010b

Now Possible to Import Data into MATLAB Dataset Object	14-2
---	-------------

R2010a

New Connection Object Methods	15-2
Enhanced Error Messages	15-2
Improved Write Performance	15-2

R2009b

Bug Fixes

R2009a

Bug Fixes

R2008b

Bug Fixes

R2008a

Bug Fixes

R2007b

Bug Fixes

R2007a

setdbprefs Accepts Structure Input	21-2
Visual Query Builder Generated M-File Includes Placeholder for Password and Assigns Preferences to Structure	21-2
Preference Added for Temporary Registry Output; Ensures Full Output for getdatasources	21-2

R2006b

Enhanced fetch Combines exec with Existing fetch	22-2
Import Data from Multiple Resultsets	22-2
Run Stored Procedures to Return Output Parameters	22-2
Specify Catalog and Schema Using Visual Query Builder .	22-2
Preferences Option to Find Additional Data Sources	22-2
MATLAB Change to Assignment of Nonscalar Structure Array Fields Might Impact Database Toolbox Users	22-3

R2006a

Bug Fixes

fastinsert Function Added	24-2
JDBC Drivers Now Supported for Visual Query Builder on Microsoft Windows Systems	24-2
Define Data Sources from Within the Visual Query Builder	24-2
setdbprefs Function Enhanced	24-3
Dynamically Add JDBC Drivers File	24-3
64-Bit FLOAT for Microsoft SQL Server Software Is Fully Supported	24-3
Generate M-File from VQB	24-3
update Function Enhanced to Export Multiple Records ..	24-3
logintimeout Function Now Supported on Linux Platforms	24-3

Bug Fixes

R2017a

Version: 7.1

New Features

Bug Fixes

Compatibility Considerations

One-step data import

You can import data from a database in one step using the `select` function. This function enables maximum memory savings by importing numeric values using data types as defined in the database. For different ways to import data, see “Data Import Using Database Explorer App or Command Line”.

Expanded data type support

You can import data with a larger variety of data types into MATLAB®. For the full list, see “Data Type Support”.

Connection object and database connection changes

The `connection` object has additional properties grouped into these categories:

- Database Properties
- Catalog and Schema Information
- Database and Driver Information

Additional properties include:

- `ReadOnly`
- `MaxDatabaseConnections`
- `DefaultCatalog`
- `Catalogs`
- `Schemas`
- `DatabaseProductName`
- `DatabaseProductVersion`
- `DriverName`
- `DriverVersion`

For ODBC drivers, the native ODBC interface is now the default database connection type.

Compatibility Considerations

These `connection` object properties have changed:

- `Handle` and `Constructor` is removed.
- `DataSource` replaces `Instance`.
- `LoginTimeout` replaces `TimeOut`.

For details about object properties, see the `connection` object.

Functionality being removed or changed

Functionality	What Happens When You Use It?	Use This Instead	Compatibility Considerations
<code>catalogs</code>	Still runs	<code>Catalogs</code> property of the <code>connection</code> object	Replace all instances of the <code>catalogs</code> function with access of the <code>Catalogs</code> property.
<code>get</code>	Still runs	<code>connection</code> object properties	Access any <code>connection</code> object property.
<code>isreadonly</code>	Still runs	<code>ReadOnly</code> property of the <code>connection</code> object	Replace all instances of the <code>isreadonly</code> function with access of the <code>ReadOnly</code> property.
<code>logintimeout</code>	Still runs	Name-value pair argument <code>'LoginTimeout'</code> in the <code>database</code> function	Remove all instances of the <code>logintimeout</code> function and specify the timeout value using the name-value pair argument <code>'LoginTimeout'</code> .
<code>ping</code>	Still runs	<code>connection</code> object properties	Access these <code>connection</code> object properties: <ul style="list-style-type: none"> • <code>MaxDatabaseConnections</code> • <code>DatabaseProductName</code> • <code>DatabaseProductVersion</code>

Functionality	What Happens When You Use It?	Use This Instead	Compatibility Considerations
			<ul style="list-style-type: none"> • DriverName • DriverVersion
schemas	Still runs	Schemas property of the connection object	Replace all instances of the schemas function with access of the Schemas property.
set	Still runs	connection object properties	You can set only the AutoCommit and ReadOnly properties.
sql2native	Still runs	Nothing	No replacement
setdbprefs('FetchIn	Still runs	Nothing	No replacement
setdbprefs('FetchBa	Still runs	Nothing	No replacement
database.ODBCConnec syntax in the database function	Warns	database syntax	Replace all instances of the database.ODBCConnection syntax with the database syntax in the database function.
resultset	Warns	Nothing	No replacement
resultset object as input argument in close	Warns	Nothing	No replacement
resultset object as input argument in get	Warns	Nothing	No replacement
resultset object as input argument in namecolumn	Warns	Nothing	No replacement
rsmd	Warns	Nothing	No replacement
rsmd object as input argument in get	Warns	Nothing	No replacement

Functionality	What Happens When You Use It?	Use This Instead	Compatibility Considerations
<code>setdbprefs('DataRet</code>	Warns	<code>table</code> data type	Replace all instances of <code>setdbprefs('DataReturnFormat</code> with <code>setdbprefs('DataReturnFormat</code>
<code>setdbprefs('ErrorHa</code>	Warns	Nothing	Replace all instances of <code>setdbprefs('ErrorHandling','</code> with <code>setdbprefs('ErrorHandling','</code>
<code>isconnection</code>	Errors	<code>isopen</code>	Replace all instances of the <code>isconnection</code> function with <code>isopen</code> .
<code>querybuilder</code>	Errors	<code>dexplore</code>	Use the Database Explorer app.
<code>setdbprefs('Default</code>	Errors	Nothing	No replacement
<code>setdbprefs('TempDir</code>	Errors	Nothing	No replacement
<code>setdbprefs('UseRegi</code>	Errors	Nothing	No replacement

Compatibility Considerations

The JDBC/ODBC bridge has been removed for the command line. For ODBC drivers, the syntaxes of the `database` function and other command-line functions use the native ODBC interface by default. The Database Explorer app still uses the JDBC/ODBC bridge for ODBC database connection.

Visual Query Builder has been removed. Use the Database Explorer app instead. When using the Database Explorer app with JDBC drivers for the first time, you must configure data sources.

R2016b

Version: 7.0

New Features

Bug Fixes

Compatibility Considerations

Graph Database Interface: Retrieve graph data from Neo4j

To import graph data in a Neo4j[®] database into MATLAB, use the MATLAB interface to Neo4j. To perform graph network analysis with graph data in MATLAB using the `digraph` object, create a Neo4j database connection. Or, you can explore the graph using MATLAB functionality. If you are familiar with the Cypher[®] query language, you can execute Cypher queries. For details about using the MATLAB interface to Neo4j, see Graph Database.

DatabaseDatastore functionality and object properties changes

To analyze data using common MATLAB functions, such as `mean` and `histogram`, you can create a tall array using the `DatabaseDatastore` object. For details about using tall arrays with Database Toolbox™, see Analyze Large Data in Database Using Tall Arrays.

To create a `DatabaseDatastore` object, use `databaseDatastore` instead of `datastore`.

The `DatabaseDatastore` object has two additional object properties. The `VariableNames` property provides the variable names of the retrieved data table. The `ReadSize` property specifies the number of rows to read from the retrieved data table.

The `read`, `readall`, and `preview` functions return data as a `table`.

Compatibility Considerations

- The `DatabaseDatastore` object properties have changed:
 - The `Cursor` property has been removed.
 - Two properties are added: `VariableNames` and `ReadSize`.

For details about the object properties, see `DatabaseDatastore`.

- This syntax for the `read` function has been removed.

```
data = read(dbds, rowcount)
```

To specify the number of rows to retrieve, set the `ReadSize` property of the `DatabaseDatastore` object instead.

- The data retrieval functionality has changed:

- The functions retrieve data as a `table` by default. Setting the database preference `DataReturnFormat` to `'table'` is not required.
- `read` and `preview` throw an error when there is no more data in the `DatabaseDatastore` object for reading.
- If `read` finds no more data to read, `hasdata` returns logical 0.

Functionality being removed or changed

Functionality	What Happens When You Use It?	Use This Instead	Compatibility Considerations
<code>bestrowid</code>	Errors	Nothing	No replacement
<code>clearwarnings</code>	Errors	Nothing	No replacement
<code>crossreference</code>	Errors	Nothing	No replacement
<code>driver</code>	Errors	Nothing	No replacement
<code>drivermanager</code>	Errors	Nothing	No replacement
<code>isdriver</code>	Errors	Nothing	No replacement
<code>isjdbc</code>	Errors	Nothing	No replacement
<code>isnullcolumn</code>	Errors	Nothing	No replacement
<code>isurl</code>	Errors	Nothing	No replacement
<code>register</code>	Errors	Nothing	No replacement
<code>unregister</code>	Errors	Nothing	No replacement
<code>versioncolumns</code>	Errors	Nothing	No replacement
<code>rsmd</code>	Still runs	Nothing	No replacement
<code>resultset</code>	Still runs	Nothing	No replacement
<code>tables(conn)</code>	Errors	Use syntaxes with at least two input arguments instead.	Remove all instances of the <code>tables(conn)</code> syntax. Replace these instances with any of the remaining syntaxes. For details, see <code>tables</code> .

R2016a

Version: 6.1

New Features

Bug Fixes

Compatibility Considerations

MATLAB Interface to SQLite: Create, read, and write data from SQLite database files without external drivers and administration

To import and update data without an existing database or database administration, use the MATLAB Interface to SQLite. For details, see Working with the MATLAB Interface to SQLite. For the supported functions, see this table.

Function	Purpose
sqlite	Create SQLite connection.
exec	Perform database operation in the SQLite database file.
fetch	Run SQL query and import data from the SQLite database file.
insert	Export data into the SQLite database file.
close	Close SQLite connection.

fetch Function Speed Improvement: Import data faster using the JDBC driver

When you connect to a database using the JDBC driver, importing data is faster using the fetch function.

Support for 32-bit Windows removed

The Database Toolbox no longer supports connection to a database using a 32-bit driver.

Compatibility Considerations

Use a 64-bit database. Or, install a 64-bit driver that works with the 32-bit database. For details, consult with your database administrator.

For Microsoft® Access™, see <http://www.mathworks.com/matlabcentral/answers/235949-how-to-connect-to-32-bit-microsoft-access-database-from-64-bit-matlab>.

Functionality being removed or changed

Functionality	What Happens When You Use It?	Use This Instead	Compatibility Considerations
clearwarnings	Warns	Nothing	No replacement
versioncolumns	Warns	Nothing	No replacement
crossreference	Warns	Nothing	No replacement
bestrowid	Warns	Nothing	No replacement
isnullcolumn	Warns	Nothing	No replacement

R2015b

Version: 6.0

New Features

Bug Fixes

Compatibility Considerations

ODBC Interface Functions: Export and retrieve database information using native ODBC connections

More Database Toolbox functions support the native ODBC interface for exporting data and retrieving database information and metadata. For a list of supported functions, see [Connecting to a Database Using the Native ODBC Interface](#).

Read and Write Performance Improvements: Import and export data more quickly

Data import and export functions can retrieve and write data faster. Particularly, `datainsert` and `update` call the `TRANSACTION` command of SQL to insert or update records faster for these databases: Microsoft SQL Server[®], MySQL[®], Oracle[®], and PostgreSQL.

Data Export Functions: Insert or replace data using table, structure, and dataset arrays

The `datainsert` and `update` functions can export data in tabular, `dataset`, and structure arrays to databases.

Functionality being removed or changed

Functionality	What Happens When You Use It?	Use This Instead	Compatibility Considerations
<code>driver</code>	Warns	Nothing	No replacement
<code>drivermanager</code>	Warns	Nothing	No replacement
<code>isconnection</code>	Warns	<code>isopen</code>	Replace all instances of <code>isconnection</code> with <code>isopen</code> .

R2015a

Version: 5.2.1

Bug Fixes

R2014b

Version: 5.2

New Features

DatabaseDatastore for applying mapreduce to data contained in relational databases

Create a DatabaseDatastore to work with large amounts of data in relational databases. Write custom functions to implement mapreduce to process large amounts of data. To create a DatabaseDatastore, you must create a DatabaseDatastore object. This object is a type of datastore.

Function	Purpose
datastore	Create a DatabaseDatastore.
hasdata	Determine if a DatabaseDatastore contains more data in the cursor object.
preview	Display the first eight records in a DatabaseDatastore.
read	Read data in a DatabaseDatastore.
readall	Read every record in a DatabaseDatastore.
reset	Reset the cursor position in a DatabaseDatastore.

Scrollable cursors for accessing data using relative and absolute position inputs

Fetch data sequentially or scroll up or down in the data without executing the query again. Scrolling within the data offers advantages when you are working with a large data set. An advantage of scrollable cursors is reading data in the middle of a large data set using the cursor position offset.

Create a scrollable cursor using `exec`. Retrieve data from a scrollable cursor using `fetch`. Use relative and absolute position inputs in `fetch` to retrieve data starting from a specific location in the data set.

R2014a

Version: 5.1

Bug Fixes

R2013b

Version: 5.0

New Features

Bug Fixes

Fast access to ODBC connections via a native ODBC driver

Support for native ODBC database connection for Windows[®] platforms. The native ODBC interface is available only for the command line. To use this interface, see Using the Native ODBC Database Connection. The native ODBC interface supports the following functions:

- database
- fetch
- exec
- insert
- fastinsert
- close

table data type support

You can return a `table` data type rather than a cell array. Use the `setdbprefs` command to set the database preference for the `DataReturnFormat` property to `'table'`.

R2013a

Version: 4.1

New Features

Bug Fixes

fetch function accepts user-defined batch sizes

setdbprefs is updated with new properties (FetchInBatches and FetchBatchSize) that support fetch when requesting large data.

R2012b

Version: 4.0

New Features

Bug Fixes

Compatibility Considerations

Database Explorer app for interactively exchanging data with databases

dexplore starts Database Explorer, which is the Database Toolbox GUI for connecting to a database and importing data to the MATLAB workspace. Alternatively, you can start Database Explorer by selecting **Database Explorer** from the **Database Connectivity and Reporting** section of the **Apps** tab in the MATLAB Toolstrip.

Functionality Being Removed or Changed

Functionality	What Happens When You Use It?	Use This Instead	Compatibility Considerations
querybuilder	Warns	dexplore	Continue to use querybuilder for exporting data.

R2012a

Version: 3.11

New Features

Bug Fixes

Execute .SQL Files

The new `runsqlscript` function lets you execute SQL commands from a `.SQL` file on a connected database, and store the results in a `CURSOR` array. You can input the results from executing `runsqlscript` to functions that accept `CURSOR` array inputs.

Improvements to the Database Constructor

When using a JDBC driver, you can input individual connection properties to the database constructor, `database`.

R2011b

Version: 3.10

New Features

Bug Fixes

Compatibility Considerations

Preferences Now Persistent Across MATLAB Sessions

The preferences you set using the Preference dialog box or the `setdbprefs` function now persist across MATLAB sessions.

Compatibility Considerations

In releases before R2011b, if you changed your preferences during a MATLAB session, these preferences would not remain in the next MATLAB session.

Change in Behavior for the `update` Function

`update` lets you update images, Booleans, doubles, and strings in a manner consistent with `fastinsert`.

Warning and Error ID Changes

Many warning and error IDs have changed from their previous versions. These warnings or errors typically appear during a function call.

Compatibility Considerations

If using warning or error IDs, you might need to change the strings you use. For example, if you turned off a warning for a certain ID, the warning might now appear under a different ID. If you use a `try/catch` statement in your code, replace the old identifier with the new identifier. There is no definitive list of the differences, or of the IDs that changed.

R2011a

Version: 3.9

New Features

Bug Fixes

New `datainsert` Function Exports MATLAB Cell Array Data into a Database Table

The new `datainsert` function inserts data from the MATLAB workspace into a database table, much like the `fastinsert` function. The new `datainsert` function is faster.

R2010b

Version: 3.8

New Features

Bug Fixes

Now Possible to Import Data into MATLAB Dataset Object

If you have Statistics Toolbox™ installed, you can now return a `dataset` object rather than a cell array. Use the `setdbprefs` command to set the database preference for the `DataReturnFormat` property to `'dataset'`.

R2010a

Version: 3.7

New Features

Bug Fixes

New Connection Object Methods

Several new connection object methods provide database-specific information. The new methods are:

- `database.catalogs`
- `database.columns`
- `database.schemas`
- `database.tables`

See the individual reference pages for more information on how to use these methods.

Enhanced Error Messages

New enhanced error messages provide more information about the error. For example, the 2009b error message `Drivers not Found/Loaded` is now `Drivers not Found/Loaded. Please verify that login information and database url are valid in 2010b`. This error will appear when the driver input is valid but the database URL is invalid.

Improved Write Performance

New bulk insert code templates provide significant performance upgrades.

R2009b

Version: 3.6

Bug Fixes

R2009a

Version: 3.5.1

Bug Fixes

R2008b

Version: 3.5

Bug Fixes

R2008a

Version: 3.4.1

Bug Fixes

R2007b

Version: 3.4

Bug Fixes

R2007a

Version: 3.3

New Features

Bug Fixes

setdbprefs Accepts Structure Input

The `setdbprefs` function now accepts a structure as input. For example, you can run the following commands to assign values to `s`:

```
s.DataReturnFormat = 'numeric';  
s.ErrorHandling = 'report';
```

You can also do this for other `setdbprefs` properties whose values you want to change. Then set the preferences using the values in `s` by running the command:

```
setdbprefs(s)
```

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

Visual Query Builder Generated M-File Includes Placeholder for Password and Assigns Preferences to Structure

When you run a query in the Visual Query Builder and select **File > Generate M-File**, the resulting M-file now includes a placeholder string `password` in the database statement. If a password is required for the connection, such as for connections established via JDBC drivers, substitute the password for the `password` string. If no password is required, the M-file will run as is. For more information, see *About Generated Files*.

The generated M-file assigns values for the preferences to the structure `s`. For more information, see the `setdbprefs` reference page.

Preference Added for Temporary Registry Output; Ensures Full Output for getdatasources

When you use `getdatasources` to view the data sources for your system, ensure that you view all data sources by specifying a temporary, writable, output directory using the new preference, `TempDirForRegistryOutput`. This is useful when you add data sources and do not have write access for the MATLAB current directory, where the toolbox temporarily writes ODBC registry settings. Without write access, `getdatasources` does not always return data sources you added. In that event, run `setdbprefs` to specify a value for the `TempDirForRegistryOutput` preference, where the value is the full path name to a directory for which you have write access.

R2006b

Version: 3.2

New Features

Bug Fixes

Compatibility Considerations

Enhanced fetch Combines exec with Existing fetch

The new function, `database.fetch`, executes the specified SQL query and imports results into the MATLAB workspace, given the connection handle `conn`. It is provided for convenience, to combine capabilities of the existing `exec` and `cursor.fetch` functions. In statements and code, do not specify `database.fetch` or `cursor.fetch` but rather, just specify `fetch` with the appropriate objects provided as arguments; the toolbox runs `database.fetch` or `cursor.fetch` as appropriate based on the arguments.

Unlike `cursor.fetch`, `database.fetch` does not return a cursor object on which you can run subsequent Database Toolbox functions, but rather returns all data to a MATLAB variable. For more information about `database.fetch` and how it differs from `cursor.fetch`, see the `fetch` reference page, as well as the `database.fetch` and `cursor.fetch` reference pages.

Import Data from Multiple Resultsets

The new function, `fetchmulti`, imports data into the MATLAB workspace from multiple resultsets, which you retrieve via an `exec` call to a stored procedure that contains two or more `select` statements.

Run Stored Procedures to Return Output Parameters

The new function, `runstoredprocedure`, executes a stored procedure using input parameters specified in a cell array to return output parameters. This allows you to retrieve the value of a variable into a MATLAB variable. `runstoredprocedure` overcomes a limitation of `exec`; when you run a stored procedures via `exec`, you can only retrieve resultsets.

Specify Catalog and Schema Using Visual Query Builder

You can now specify the catalog and schema for a data source using the Visual Query Builder. The default is none, meaning you do not need to select values for them.

Preferences Option to Find Additional Data Sources

The new `setdbrprefs` option, `UseRegistryForSources`, instructs the Visual Query Builder to search the Microsoft Windows registry to find any ODBC data sources not uncovered using the system `ODBC.INI`.

MATLAB Change to Assignment of Nonscalar Structure Array Fields Might Impact Database Toolbox Users

In Version 7.3 (R2006b) of the MATLAB software, a change was made to how a nonscalar structure array field is assigned to a single MATLAB variable. For more information, see [Assigning Nonscalar Structure Array Fields to a Single Variable](#) in the MATLAB Release Notes.

Compatibility Considerations

As a result of this change in the MATLAB software, you may need to modify your Database Toolbox M-files.

R2006a

Version: 3.1.1

Bug Fixes

R14SP3

Version: 3.1

New Features

Bug Fixes

fastinsert Function Added

There is a new function, `fastinsert`, that you can use instead of the `insert` function to export data about three times more quickly than `insert`. It also allows exporting for all object types, so that any data you can retrieve from a database you now can export to the database, including binary objects.

While there are no known problems with `fastinsert`, if you receive unexpected results, return to using `insert` and report the problem with `fastinsert` via Technical Support.

Note that the Visual Query Builder insert feature uses the `insert` function instead of `fastinsert`.

JDBC Drivers Now Supported for Visual Query Builder on Microsoft Windows Systems

You now can use the Visual Query Builder (VQB) with JDBC drivers on Windows platforms. Previously, only ODBC drivers were supported.

The `confds` function now displays an enhanced dialog box you use to define JDBC data sources. With it, you save and load data source information via MATLAB MAT-files.

For details, see *Setting Up JDBC Data Sources* in the Database Toolbox documentation.

Define Data Sources from Within the Visual Query Builder

The Visual Query Builder now includes two new items under the **Query** menu:

- **Define ODBC Data Source**—Directly access your Windows ODBC Data Source Administrator dialog box where you define ODBC data sources.
- **Define JDBC Data Source**—Access the Define JDBC Data Source dialog box for defining JDBC data sources to use with the VQB. The function equivalent is `confds`. When you define a JDBC data source, the information is saved in a MAT-file so you can use it again in a later session. Later, open the MAT-file using the Define JDBC Data Source dialog box, or using `setdbprefs('JDBCDataSourceFile','fullpathtomatfile')`.

For details, see *Configuring Your Environment* in the Database Toolbox documentation.

setdbprefs Function Enhanced

New arguments are supported for defining the JDBC data source MAT-file. For details, see the `setdbprefs` reference page.

Dynamically Add JDBC Drivers File

You can dynamically add a JDBC drivers file to the MATLAB Java[®] `classpath` using the MATLAB `javaaddpath` function. You can use this method instead of adding a pointer to the JDBC drivers file in your `classpath.txt` file. The advantage of using `javaaddpath` is that you do not have to restart the MATLAB software session after running the `javaaddpath` statement. The disadvantage is that this only applies to the current session and so you need to run the `javaaddpath` statement in each new session. For details, see *Setting Up JDBC Data Sources* in the Database Toolbox documentation.

64-Bit FLOAT for Microsoft SQL Server Software Is Fully Supported

You now can retrieve 64-bit FLOAT data using Microsoft SQL Server software.

Generate M-File from VQB

After running a query using the Visual Query Builder, you can generate an M-file consisting of Database Toolbox functions that perform the query. This is useful if you know how to run queries with the VQB and want to determine the equivalent functions, particularly the SQL statements in `exec` and `insert`.

update Function Enhanced to Export Multiple Records

The `update` function has been enhanced so that you can export multiple records based on different `where` clauses. The number of `where` clauses must equal the number of records in the array of data you are exporting. For details, see the reference page for `update`.

logintimeout Function Now Supported on Linux Platforms

The `logintimeout` function is now supported on Linux[®] platforms.

R14SP2

Version: 3.0.2

Bug Fixes

