

OPC Toolbox™ Release Notes



MATLAB® & SIMULINK®

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

OPC Toolbox™ Release Notes

© COPYRIGHT 2005–2016 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.

R2016a

Bug Fixes

R2015b

Support for OPC-UA standard 2-2

R2015a

Bug Fixes

R2014b

Bug Fixes

R2014a

OPC Client App Renamed OPC Data Access Explorer	5-2
--	------------

R2013b

Bug Fixes

R2013a

Namespace browser for viewing item names on OPC HDA and OPC DA servers	7-2
opchda function for creating an OPC HDA client object	7-2
New and enhanced troubleshooting utilities	7-2

R2012b

Bug Fixes

R2012a

Conversion of Error and Warning Message Identifiers	9-2
--	------------

R2011b

64-Bit Support 10-2

R2011a

Support for Historical Data Access Servers 11-2
 New OPC HDA Functions 11-2
 Getting Help 11-3
 Demos 11-3

R2010b

Bug Fixes

R2010a

Bug Fixes

R2009b

Bug Fixes

R2009a

Bug Fixes

R2008b

Bug Fixes

R2008a

Bug Fixes

R2007b

Graphical Display of OPC Data	18-2
Time Series Objects Now Supported	18-2

R2007a

Bug Fixes

R2006b

Bug Fixes

R2006a

opcregister Function Enhanced 21-2

R14SP3

Full Event Log 22-2

Changing TimerPeriod Property 22-2

R14SP2+

**Support for Simulink Software: OPC Toolbox Block
Library 23-2**

New Item Property: QualityID 23-2

R2016a

Version: 4.0.1

Bug Fixes

R2015b

Version: 4.0

New Features

Bug Fixes

Support for OPC-UA standard

This release of OPC Toolbox™ includes support for the OPC Unified Architecture (UA) standard protocol.

For information on getting started with OPC UA servers, nodes, and clients, see Unified Architecture.

Note OPC UA support is available only for 64-bit Windows® operating systems.

R2015a

Version: 3.3.3

Bug Fixes

R2014b

Version: 3.3.2

Bug Fixes

R2014a

Version: 3.3.1

New Features

Bug Fixes

OPC Client App Renamed OPC Data Access Explorer

The app known as OPC Client in past releases is now called the OPC Data Access Explorer. `opcDataAccessExplorer` is a new corresponding command-line function that opens the app programmatically. The function `opcTool` continues to open this app.

R2013b

Version: 3.3

Bug Fixes

R2013a

Version: 3.2

New Features

Bug Fixes

Namespace browser for viewing item names on OPC HDA and OPC DA servers

There are new tools that let you graphically browse the name space for OPC HDA or OPC DA servers:

- For more information on the OPC DA namespace browser, see the `browsnamespace` reference page.
- For more information on the OPC HDA namespace browser, see the `browseNameSpace` reference page.

opchda function for creating an OPC HDA client object

A new function, `opchda`, creates an OPC HDA client object in MATLAB. For more information on this function, see the `opchda` reference page.

New and enhanced troubleshooting utilities

The troubleshooting utility `opcsupport` has been enhanced to support OPC HDA as well as OPC DA functionality.

Two new troubleshooting utilities are provided specifically for OPC HDA and OPC DA:

- For more information on the OPC DA troubleshooting utility, see the `opc.daSupport` reference page.
- For more information on the OPC HDA troubleshooting utility, see the `opc.hdaSupport` reference page.

R2012b

Version: 3.1.2

Bug Fixes

R2012a

Version: 3.1.1

New Features

Bug Fixes

Compatibility Considerations

Conversion of Error and Warning Message Identifiers

For R2012a, error and warning message identifiers have changed in OPC Toolbox.

Compatibility Considerations

If you have scripts or functions that use message identifiers that changed, you must update the code to use the new identifiers. Typically, message identifiers are used to turn off specific warning messages, or in code that uses a try/catch statement and performs an action based on a specific error identifier.

To determine the identifier for a warning, run the following command just after you see the warning:

```
[MSG,MSGID] = lastwarn;
```

This command saves the message identifier to the variable `MSGID`.

To determine the identifier for an error, run the following commands just after you see the error:

```
exception = MException.last;  
MSGID = exception.identifier;
```

Tip Warning messages indicate a potential issue with your code. While you can turn off a warning, a suggested alternative is to change your code so that it runs without warnings.

R2011b

Version: 3.1

New Features

Bug Fixes

64-Bit Support

This release supports OPC Toolbox in 64-bit MATLAB®.

Before using OPC Toolbox with 64-bit MATLAB for the first time, you must run `opcregister` to install the 64-bit OPC Core Components. This applies even if you have previously installed the OPC Core Components, either through 32-bit MATLAB or installation of an OPC Server on the local host. Failure to install the 64-bit OPC Core Components that ship with this version of OPC Toolbox can result in local OPC servers being unavailable to OPC Toolbox.

R2011a

Version: 3.0

New Features

Bug Fixes

Support for Historical Data Access Servers

OPC Toolbox now supports OPC Historical Data Access (OPC HDA) specification version 1.2. This allows you to access and visualize historical process data from an OPC Historical Data Access server (commonly referred to as an historian).

Key features of OPC HDA support are:

- You can browse networks for Historical Data Access servers.
- A single OPC HDA client object allows you to manage connections to an HDA server, to browse the server name space, and to read raw and processed data from the server.
- An OPC Historical Data Access object allows you to easily manipulate and visualize data retrieved from OPC HDA servers.

New OPC HDA Functions

The OPC HDA read functions include:

- `readRaw` — Reads raw historical data for a selection of item tags over a selected time span.
- `readAtTime` — Reads data from the server at specific time intervals. It can return array data as an output, by specifying an optional `'datatype'` argument.
- `readProcessed` — Reads processed data from the server for chosen item tags using an aggregate type defined from a list of available types. It can return array data as an output, by specifying an optional `'datatype'` argument.
- `readModified` — Reads all modified data between two times for a particular item.

The OPC HDA data object functions include:

- `tsunion` — (Time stamp union) Converts all data to have the same time stamps, by resampling at the union of the time stamps of all items.
- `tsintersect` — (Time stamp intersection) Converts all data to have the same time stamps, using the time stamps common to all items in the array. Removes all data that is not common to all elements (items) in the data object.
- `resample` — Converts all data to have the given (regularly) sampled time stamps that you provided. Combines elements with the same item ID, so that `resample` creates data objects with unique item IDs.
- `plot` — Generates a plot so you can visualize the historical data.

Getting Help

For a command-window listing of available functions and links to their specific help, type:

```
help opc.hda
```

You access the Historical Data Access (HDA) functions through the `opc.hda` package. For this reason, to get help on a particular HDA function, you must prefix the function name with `opc.hda`. For example, to get help on the `getServerInfo` function, type:

```
help opc.hda.getServerInfo
```

Other prefixes, such as `opc`, `opc.hda.Data`, or `opc.hda.Client` might be necessary, as listed in the output for `help opc.hda`.

To view OPC HDA documentation in the help browser, see

- Quick Start: Using OPC Historical Data Access Functions
- Historical Data Access User's Guide
- OPC Historical Data Access (HDA) functions category

Demos

Demos for OPC Historical Data Access Tutorials are in the help documentation under the **Demos** node in OPC Toolbox.

R2010b

Version: 2.1.6

Bug Fixes

R2010a

Version: 2.1.5

Bug Fixes

R2009b

Version: 2.1.4

Bug Fixes

R2009a

Version: 2.1.3

Bug Fixes

R2008b

Version: 2.1.2

Bug Fixes

R2008a

Version: 2.1.1

Bug Fixes

R2007b

Version: 2.1

New Features

Bug Fixes

Graphical Display of OPC Data

The new `trend` function provides a graphical display of live OPC data for an OPC group object. You can now watch live data stream to a MATLAB figure window. The `trend` functionality is independent of OPC logging tasks. More information is available on the `trend` reference page.

Time Series Objects Now Supported

OPC Toolbox software now supports time series objects, allowing easier analysis and visualization of time domain data in MATLAB software. The addition of this functionality involves the extension of the existing OPC Toolbox functions `getdata` and `opcread` to support the creation of MATLAB time series objects. The new functions `opconstruct2timeseries` and `opcqid` have also been added to the toolbox to support this feature. More information is available on the reference pages for these functions.

R2007a

Version: 2.0.4

Bug Fixes

R2006b

Version: 2.0.3

Bug Fixes

R2006a

Version: 2.0.2

New Features

opregister Function Enhanced

The `opregister` function has been enhanced with a `'-silent'` option to install OPC Foundation Core components without dialog boxes.

R14SP3

Version: 2.0.1

New Features

Full Event Log

The event log is no longer cleared when a new event arrives and the event log is full. Instead, the oldest event is removed to make space for the new event. For more information, see the reference page for the `EventLogMax` property, by typing

```
doc EventLogMax
```

Changing TimerPeriod Property

You can now change the `TimerPeriod` property while a client object is connected.

R14SP2+

Version: 2.0

New Features

Support for Simulink Software: OPC Toolbox Block Library

This release of OPC Toolbox software includes support for communicating with OPC servers from Simulink® software. The OPC Toolbox block library includes blocks for reading data from an OPC server, writing data to an OPC server, and running a Simulink simulation in pseudo real time (by slowing the simulation to run at system time). For more information on using OPC Toolbox software with Simulink software, see *Using the OPC Toolbox Block Library in the User's Guide*.

New Item Property: QualityID

The `QualityID` property has been added to `daitem` objects. The `QualityID`, expressed as a 16-bit unsigned integer, represents the quality of the data item when last read. To work with the `QualityID` property, you use the function `opcqparts` to convert the `QualityID` property into vendor, major, substatus, and limit status information; and the function `opcqstr` to convert the `QualityID` property into a string. For more information, see the `QualityID` property reference page.