

**MATLAB<sup>®</sup> Compiler SDK<sup>™</sup>**

**MATLAB<sup>®</sup> Production Server<sup>™</sup> Excel<sup>®</sup> Client Guide**



**MATLAB<sup>®</sup>**

R2016a



## How to Contact MathWorks



Latest news: [www.mathworks.com](http://www.mathworks.com)  
Sales and services: [www.mathworks.com/sales\\_and\\_services](http://www.mathworks.com/sales_and_services)  
User community: [www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)  
Technical support: [www.mathworks.com/support/contact\\_us](http://www.mathworks.com/support/contact_us)



Phone: 508-647-7000



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

*MATLAB® Compiler SDK™ MATLAB® Production Server™ Excel® Client Guide*

© COPYRIGHT 1984–2016 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](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

### Patents

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

### Revision History

March 2015	Online only	New for Version 6.0 (Release R2015a)
September 2015	Online only	Revised for Version 6.1 (Release 2015b)
March 2016	Online only	Revised for Version 6.2 (Release 2016a)

## 1

### Integrating with MATLAB Production Server

<b>Install the Deployable Archive</b> .....	<b>1-2</b>
<b>Install the Excel Add-In</b> .....	<b>1-3</b>
Install with the Generated Installer .....	<b>1-3</b>
Install with the Raw Files .....	<b>1-4</b>
<b>Data Marshaling Rules</b> .....	<b>1-6</b>
Default Marshaling Rules .....	<b>1-6</b>
Change Rules for Marshaling Data into MATLAB .....	<b>1-6</b>
Change Rules for Marshaling Data into Excel .....	<b>1-7</b>
<b>Change the Server Configuration</b> .....	<b>1-8</b>
<b>Enable Client Logging</b> .....	<b>1-10</b>
<b>XLA File Not Generated</b> .....	<b>1-12</b>
<b>Server Configuration Add-in Not Enabled</b> .....	<b>1-13</b>
<b>Error Using a Variable Number of Outputs</b> .....	<b>1-14</b>



# Integrating with MATLAB Production Server

---

- “Install the Deployable Archive” on page 1-2
- “Install the Excel Add-In” on page 1-3
- “Data Marshaling Rules” on page 1-6
- “Change the Server Configuration” on page 1-8
- “Enable Client Logging” on page 1-10
- “XLA File Not Generated” on page 1-12
- “Server Configuration Add-in Not Enabled” on page 1-13
- “Error Using a Variable Number of Outputs” on page 1-14

## Install the Deployable Archive

The Production Server Compiler app generates a deployable archive containing the compiled MATLAB® code used by the add-in. This archive needs to be deployed to a MATLAB Production Server™ instance before the add-in will work.

To install the deployable archive on a server instance:

- 1** Locate the archive in the `for_redistribution_files_only\server\` folder.

The file name is similar to *archiveName.ctf*.

- 2** Copy the archive file to the `auto_deploy` folder of the server instance.

For more information, see “MATLAB Production Server” documentation.

## Install the Excel Add-In

### In this section...

“Install with the Generated Installer” on page 1-3

“Install with the Raw Files” on page 1-4

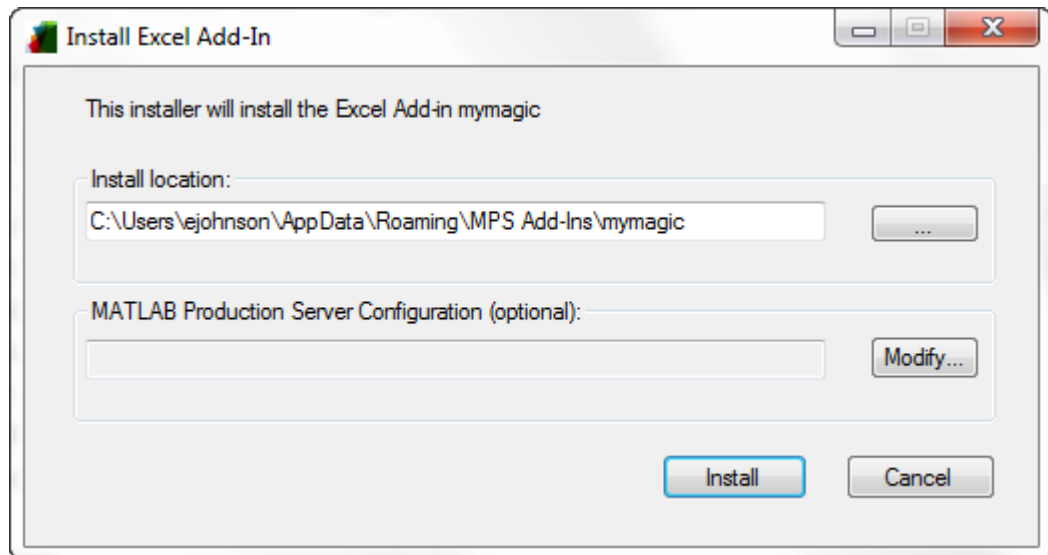
### Install with the Generated Installer

**Note:** The client installer installs under user's local profile which does not require administrative privileges.

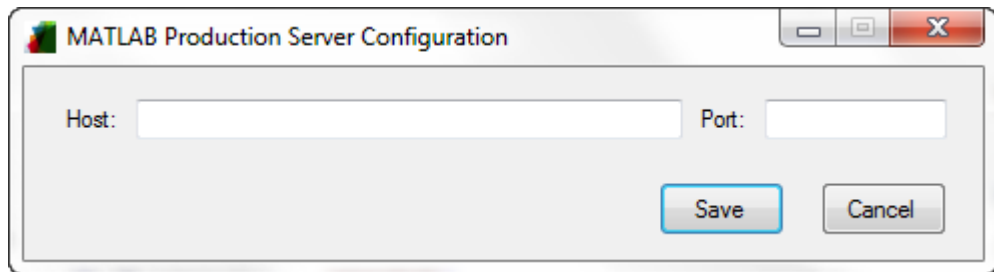
The Production Server Compiler app generates an installer that you use to install the add-in and configure it to connect to a MATLAB Production Server instance.

To install the add-in:

- 1 Locate the installer in the `for_redistribution\client` folder.
- 2 Double click the installer to start it.



- 3 In the **Install location** field, enter the location where the Excel® add-in is installed.
- 4 Specify the URL of the deployable archive powering the Excel add-in.
  - a Click **Modify**.



- b In the **Host** field, enter the host name or the full URL of the machine running the MATLAB Production Server instance with the deployable archive.
  - c In the **Port** field, enter the port number on which the server instance listens for incoming requests.

The default port number is **9910**.
  - d Click **Save**.
- 5 Click **Install**.

## Install with the Raw Files

In the `client\for_redistribution_files_only` you will find the following files:

- `projName.dll`— Dynamic library required by the add-in
- `projName.bas` — VBA module file created for the add-in
- `projName.xla` — Excel add-in file

---

**Note:** If `projName.xla` is not generated, you can still install the add-in using `projName.bas`. See “XLA File Not Generated” on page 1-12 for more information.

---

- `serverconfig.dll`- Excel add-in that manages the server configuration

On the machine running the add-in, you must register the following DLLs with the `regAsm /codebase` command:

- `mymagic.dll`



- `serverconfig.dll`

If `mymagic.xla` is generated, install the add-in into Excel. If not, import `mymagic.bas` to the workbook's Visual Basic project.

---

**Note:** If you do not have administrative privileges on the machine, you can run `regAsm` with the `/regfile` option.

---

## Data Marshaling Rules

### In this section...

“Default Marshaling Rules” on page 1-6

“Change Rules for Marshaling Data into MATLAB” on page 1-6

“Change Rules for Marshaling Data into Excel” on page 1-7

### Default Marshaling Rules

These types of data do not have natural mappings between MATLAB and Excel:

- Dates: Excel has a special data type for dates, and MATLAB does not.
- Blank cells: MATLAB has no equivalent construct for a blank cell in an Excel spread sheet.

If you do not change the marshaling rules when compiling the add-in, the rules for marshaling Excel data into MATLAB are:

- Excel dates are marshaled into MATLAB doubles.
- Empty cells are marshaled into zeros.

If you do not change the marshaling rules when compiling the add-in, the rules for marshaling MATLAB data into Excel are:

- MATLAB NaNs are marshaled into Visual Basic<sup>®</sup> #QNANs.
- MATLAB does not return any Excel dates.

### Change Rules for Marshaling Data into MATLAB

You can change how dates and empty cells are marshaled into MATLAB when compiling the add-in:

- Excel dates can be marshaled as MATLAB strings.
- Empty cells can be marshaled as MATLAB NaNs.

To change the marshaling rules:

- 1 In the class mapper portion of the **MATLAB Compiler** project window, select the signature of the function you want to modify.

- 2 Select **Data Conversion Properties** from the context menu.
- 3 Select the input argument rules to change.
- 4 Click outside of the dialog box to close it.

## Change Rules for Marshaling Data into Excel

You can change how dates and NaNs are marshaled into Excel when compiling the add-in:

- MATLAB NaNs can be converted into zeros.
- MATLAB numeric values can be converted into Excel dates.

---

**Note:** To see a date in the expected format, ensure that the Excel cell is formatted to display its contents in a date format.

---

To change the marshaling rules:

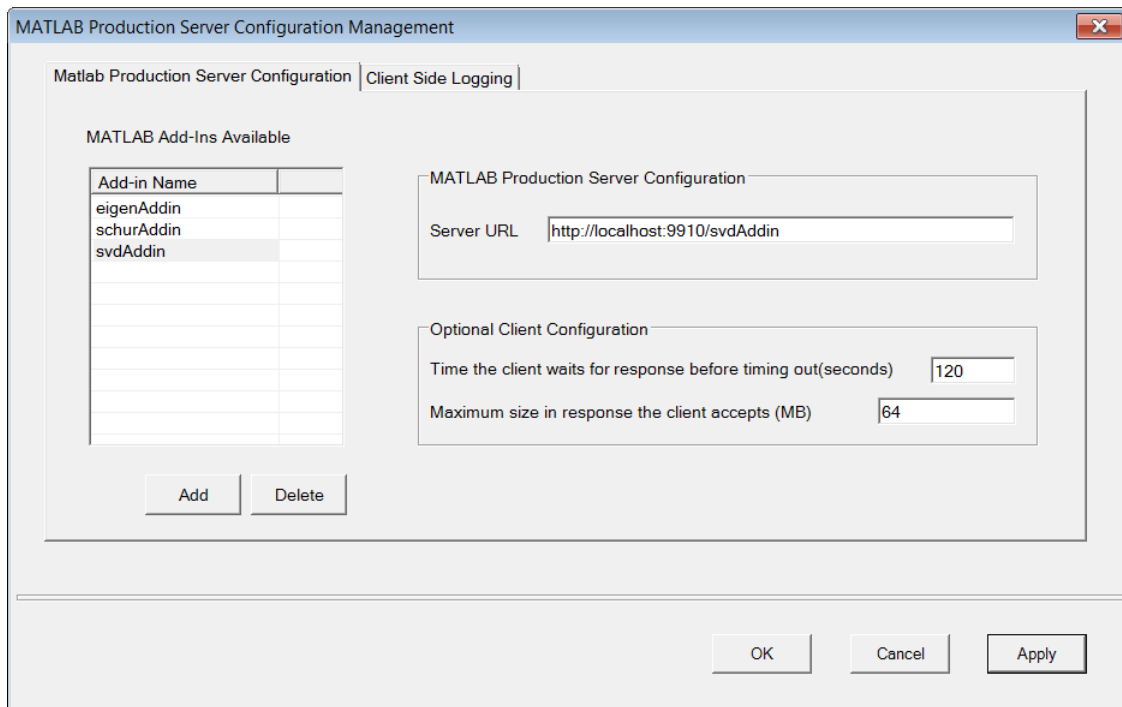
- 1 In the class mapper portion of the **MATLAB Compiler** project window, select the signature of the function you want to modify.
- 2 Select **Data Conversion Properties** from the context menu.
- 3 Select the output argument rules to change.
- 4 Click outside of the dialog to close it.

## Change the Server Configuration

Before you can run an Excel add-in on a MATLAB Production Server instance, you need to configure it.

To configure the server information for an add-in:

- 1 In Excel, select **Add-Ins > Configure MATLAB Production Server**.



- 2 Select the **MATLAB Production Server Configuration** tab.
- 3 Select the add-in to configure. The server URL corresponding to the add-in displays in the **Server URL** field.

If the add-in is not listed, click **Add** under the **MATLAB Add-ins Available** table. Enter the name of the add-in, and click **OK**.

- 4 If the **Server URL** field is empty, enter the URL based on the application protocol you want to use: HTTP or HTTPS. A server certificate needs to be installed in the

certificate store of the client or local machine running the Excel add-in when using HTTPS.

Protocol	URL	Default Port Number
HTTP	http://<HostName>:<PortNumber>/ ctfArchiveName	9910
HTTPS	https://<HostName>:<PortNumber>/ ctfArchiveName	9920

- 5 Modify the **Optional Client Configuration** options if necessary.
- 6 Click **OK**.

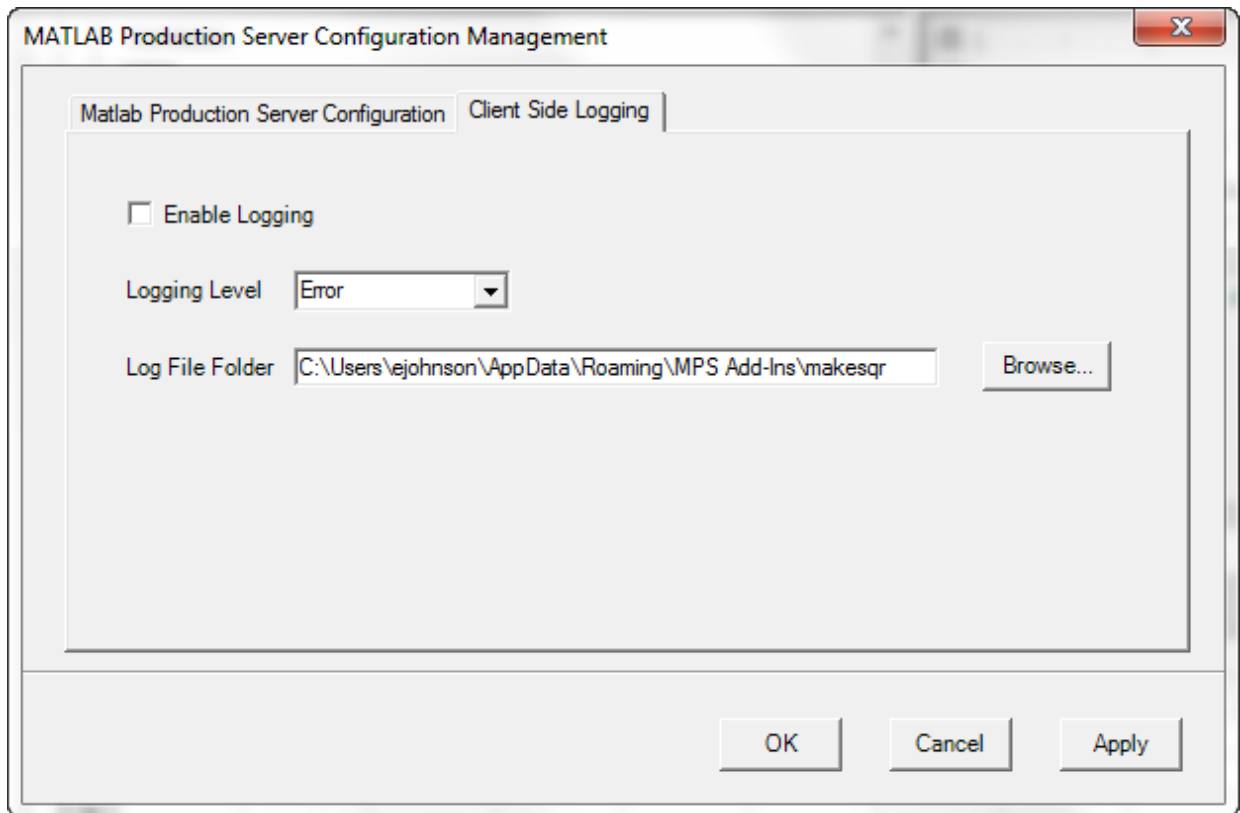
## More About

- “Install the Excel Add-In” on page 1-3

## Enable Client Logging

Client logging writes messages to a persistent log file. The messages are useful for debugging issues with the add-in. Client logging is disabled by default. To activate it:

- 1 In Excel, select **Add-Ins > Configure MATLAB Production Server**.



- 2 Select the **Client Side Logging** tab.
- 3 Check the **Enable Logging** box.
- 4 Select the logging level from the **Logging Level** drop down list.
  - **Error** — Log error messages.
  - **Information** — Log information about server interactions in addition to error messages.

- **Verbose** — Log everything.

**5** In the **Log File Folder** box, enter the folder into which log files are written.

When client logging is enabled all registered MATLAB Production Server add-ins generate log files. The files follow the naming convention *logFolder\addinName-date.log*.

## **XLA File Not Generated**

The compiler may not generate the *projName.xla* file for a number of reasons, including that Excel is not configured to trust access to the VBA project object model. When this happens, you can install the add-in by importing the *projName.bas* file into the workbook's Visual Basic project.



## Server Configuration Add-in Not Enabled

If your trust settings in Excel are configured to either disable all add-ins or to require add-ins to be published by a trusted publisher, it is possible that the **Configure MATLAB Production Server** add-in is not available after installation. In most cases, the add-in is installed but disabled.

To check if the add-in is installed in Excel:

- 1 Select **File>Options**.
- 2 Select **Add-Ins**.
- 3 Look for **ServerConfig.Connect** in the list of disabled add-ins.

You can enable the add-in by adjusting the trust settings in Excel.

## Error Using a Variable Number of Outputs

If your add-in throws the error:

```
Error in myfunc: Object reference not set to an instance of an object
```

The likely cause is that the MATLAB function used by the add-in returns a variable number of outputs.

Add-ins using code run on a MATLAB Production Server instance do not support MATLAB functions that return a variable number of outputs. You can either rewrite your MATLAB function to return a fixed number of outputs, or you can create an add-in that runs locally to your Excel installation.