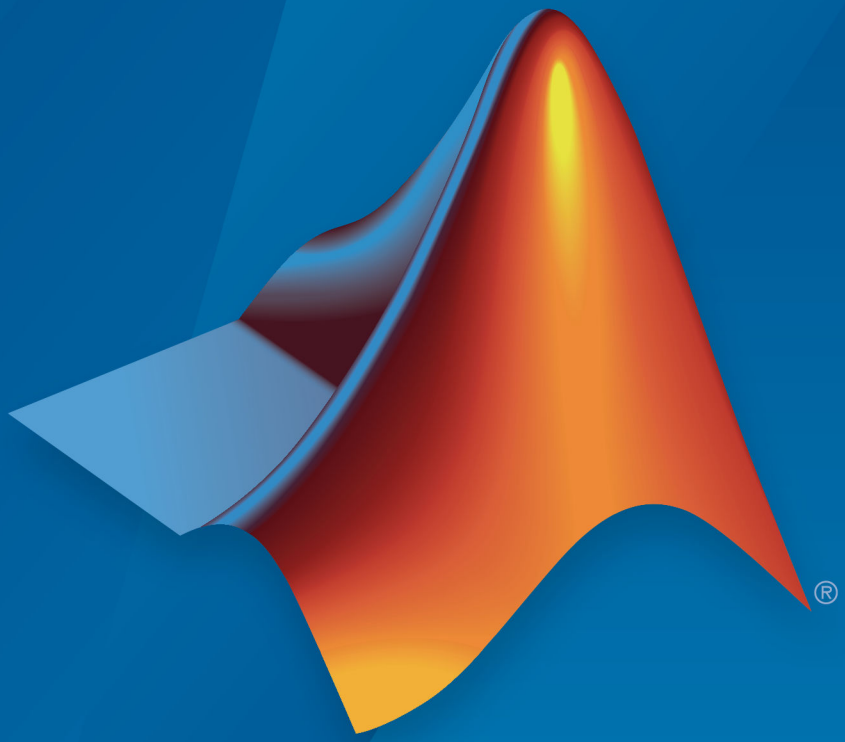
MATLAB® Production Server™

Excel® Client Guide



MATLAB®

R2018b

MathWorks®

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® Production Server™ Excel® Client Guide

© COPYRIGHT 1984–2018 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.

Revision History

September 2015	Online only	New for Version 2.2 (Release R2015b)
March 2016	Online only	Revised for Version 2.3 (Release 2016a)
September 2016	Online only	Revised for Version 2.4 (Release 2016b)
March 2017	Online only	Revised for Version 3.0 (Release 2017a)
September 2017	Online only	Revised for Version 3.0.1 (Release R2017b)
March 2018	Online only	Revised for Version 3.1 (Release R2018a)
September 2018	Online only	Revised for Version 4.0 (Release R2018b)

Contents

1 Integrating With MATLAB Production Server

Create and Install a Deployable Archive with Excel Integration For MATLAB Production Server **1-2**
- Create Function In MATLAB **1-2**
- Create Deployable Archive with Excel Integration Using Production Server Compiler App **1-2**
- Customize the Application and Its Appearance **1-3**
- Package the Application **1-6**
- Install the Deployable Archive with Excel Integration **1-6**

Install the Excel Add-In **1-8**
- Install with the Generated Installer **1-8**
- Install with the Raw Files **1-9**

Data Marshaling Rules **1-10**
- Default Marshaling Rules **1-10**
- Change Rules for Marshaling Data into MATLAB **1-10**
- Change Rules for Marshaling Data into Excel **1-11**

Change the Server Configuration **1-12**

Enable Client Logging **1-14**

XLA File Not Generated **1-16**

Server Configuration Add-in Not Enabled **1-17**

Error Using a Variable Number of Outputs **1-18**

Integrating With MATLAB Production Server

- “Create and Install a Deployable Archive with Excel Integration For MATLAB Production Server” on page 1-2
- “Install the Excel Add-In” on page 1-8
- “Data Marshaling Rules” on page 1-10
- “Change the Server Configuration” on page 1-12
- “Enable Client Logging” on page 1-14
- “XLA File Not Generated” on page 1-16
- “Server Configuration Add-in Not Enabled” on page 1-17
- “Error Using a Variable Number of Outputs” on page 1-18

Create and Install a Deployable Archive with Excel Integration For MATLAB Production Server

Supported platform: Windows®

This example shows how to create a deployable archive with Excel integration from a MATLAB function. You can then give the generated archive to a system administrator to deploy on MATLAB Production Server.

Create Function In MATLAB

In MATLAB, examine the MATLAB program that you want to package.

For this example, write a function `mymagic.m` as follows.

```
function y = mymagic(x)
% MYMAGIC Magic square of size x.
%   Y = MYMAGIC(X) returns a magic square of size x.
%   This file is used as an example for the MATLAB Compiler product.
%   Copyright 2001-2007 The MathWorks, Inc.
y = magic(x);
```

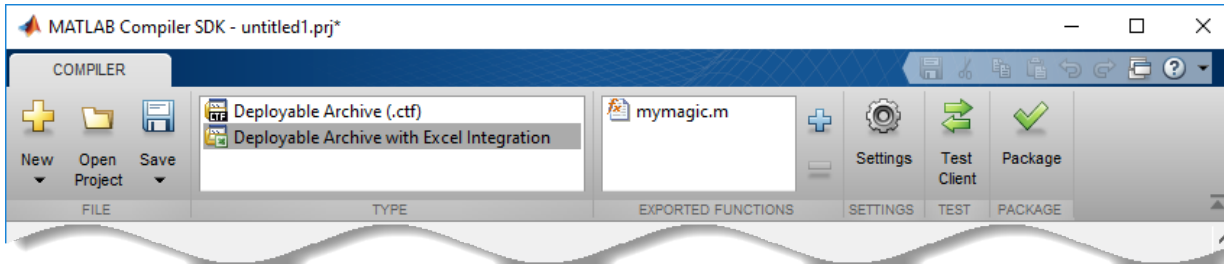
At the MATLAB command prompt, enter `mymagic(3)`.

The output is:

```
ans =
     8     1     6
     3     5     7
     4     9     2
```

Create Deployable Archive with Excel Integration Using Production Server Compiler App

1. On the **MATLAB Apps** tab, on the far right of the **Apps** section, click the arrow. In **Application Deployment**, click **Production Server Compiler**. In the **MATLAB Compiler SDK** project window, click **Deployable Archive with Excel integration**.



Alternatively, you can open the **Production Server Compiler** app by entering `productionServerCompiler` at the MATLAB prompt.

2. In the **MATLAB Compiler SDK** project window, specify the files of the MATLAB application that you want to deploy.

   1. In the **Exported Functions** section, click .
   2. In the **Add Files** window, browse to the example folder, and select the function you want to package.

      Click **Open**.

   The function `mymagic.m` is added to the list of main files.

Customize the Application and Its Appearance

You can customize your deployable archive with Excel integration, and add more information about the application as follows:

- **Archive information** — Editable information about the deployed archive with Excel integration.
- **Client configuration** — Configure the MATLAB Production Server client. Select the **Default Server URL**, decide wait time-out, and maximum size of response for the client, and provide an optional self-signed certificate for `https`.
- **Additional files required for your archive to run** — Additional files required by the generated archive to run. These files are included in the generated archive installer. See “Manage Required Files in Compiler Project” (MATLAB Compiler SDK).
- **Files installed with your archive** — Files that are installed with your archive on the client and server. The files installed on the server include:

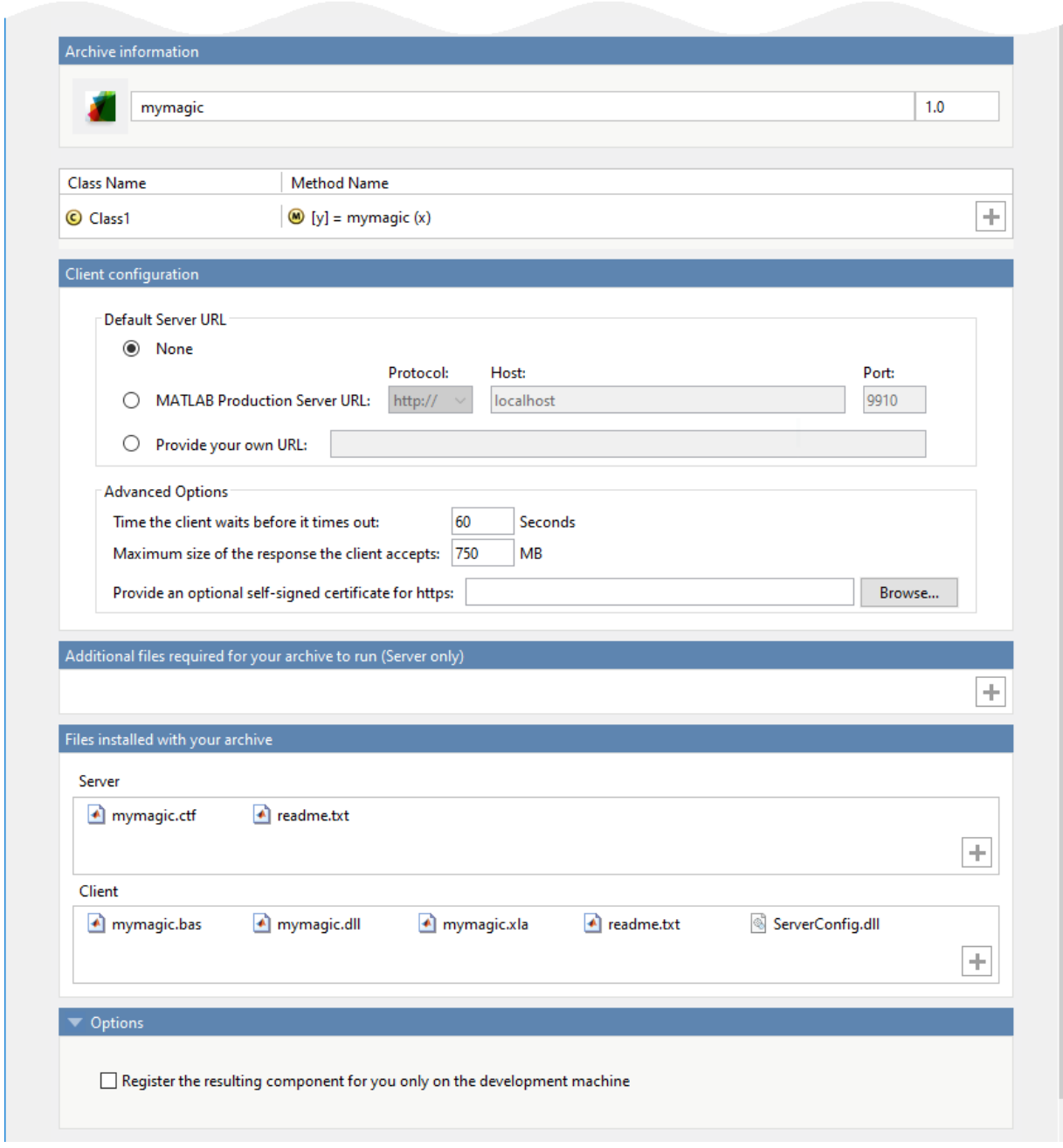
- Generated deployable archive (.ctf)
- Generated `readme.txt`

The files installed on the client include:

- `mymagic.bas`
- `mymagic.dll`
- `mymagic.xla`
- `readme.txt`
- `ServerConfig.dll`

See “Specify Files to Install with Application” (MATLAB Compiler SDK).

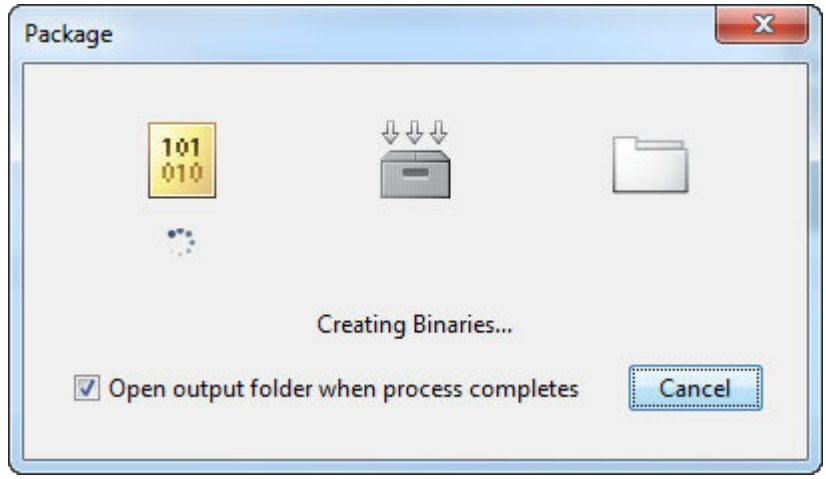
- **Options** — The option **Register the resulting component for you only on the development machine** exclusively registers the packaged component for one user on the development machine.


Archive information
mymagic
1.0
Class Name
Method Name
Class1
[y] = mymagic (x)
Client configuration
Default Server URL
None
Protocol:
Host:
Port:
MATLAB Production Server URL:
http://
localhost
9910
Provide your own URL:
Advanced Options
Time the client waits before it times out:
60
Seconds
Maximum size of the response the client accepts:
750
MB
Provide an optional self-signed certificate for https:
Browse...
Additional files required for your archive to run (Server only)
Files installed with your archive
Server
mymagic.ctf
readme.txt
Client
mymagic.bas
mymagic.dll
mymagic.xla
readme.txt
ServerConfig.dll
Options
Register the resulting component for you only on the development machine

Package the Application

1 To generate the packaged application, click **Package**.

In the Save Project dialog box, specify the location to save the project.



2 In the **Package** dialog box, verify that **Open output folder when process completes** is selected.

When the deployment process is complete, examine the generated output.

- `for_redistribution` — Folder containing the installer to distribute the archive on the MATLAB Production Server client and server
- `for_redistribution_files_only` — Folder containing the files required for redistributing the application on the MATLAB Production Server client and server
- `for_testing` — Folder containing the raw generated files to create the installer
- `PackagingLog.txt` — Log file generated by MATLAB Compiler™

Install the Deployable Archive with Excel Integration

The archive must be deployed to a MATLAB Production Server instance before the add-in works.

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. The server instance automatically deploys it and makes it available to interested clients.

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

See Also

`deploytool` | `mcc` | `productionServerCompiler`

More About

- Production Server Compiler

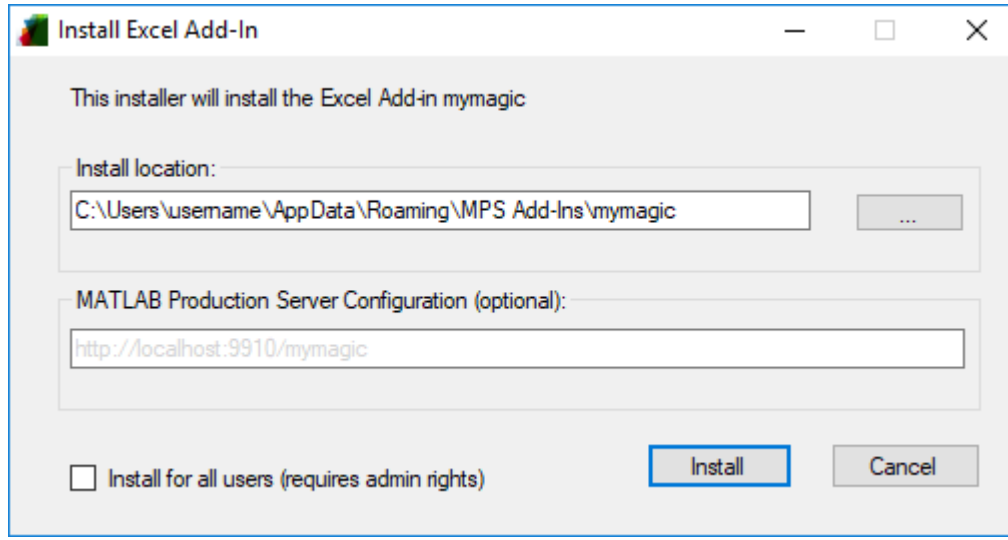
Install the Excel Add-In

The **Production Server Compiler** app is used to “Create and Install a Deployable Archive with Excel Integration For MATLAB Production Server” on page 1-2. You can install the add-in and configure it to connect to a MATLAB Production Server instance using the generated installer or raw files.

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

Install with the Generated Installer

1. Double-click the installer in the `for_redistribution\client` folder.



2. In the **Install location** field, enter the location where the Excel add-in is installed.
3. In the **MATLAB Production Server Configuration** field, specify the URL of the deployable archive that contains the Excel add-in.

   Click **Install**.

Install with the Raw Files

In the `client\for_redistribution_files_only` folder, you find the following files:

- *`projName`*`.dll`— Dynamic library required by the add-in
- *`projName`*`.bas` — VBA module file created for the add-in
- `serverconfig.dll`- Excel add-in that manages the server configuration
- *`projName`*`.xla` — Excel add-in file

> **Note** The compiler might not generate the *`projName`*`.xla` file for various 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 Visual Basic project of the workbook.

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 Visual Basic project of the workbook.

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

See Also

More About

- “Create and Install a Deployable Archive with Excel Integration For MATLAB Production Server” on page 1-2

Data Marshaling Rules

In this section...
“Default Marshaling Rules” on page 1-10
“Change Rules for Marshaling Data into MATLAB” on page 1-10
“Change Rules for Marshaling Data into Excel” on page 1-11

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® `#QNAN`s.
- 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 character arrays.
- 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 box to close it.

See Also

More About

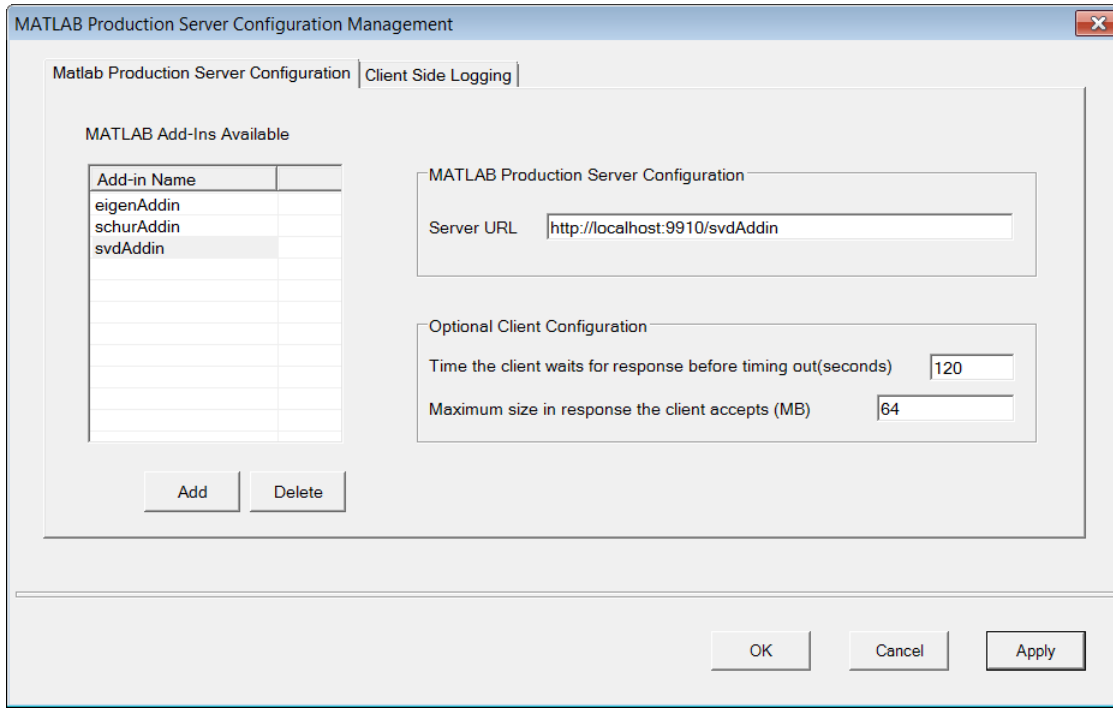
- “Install the Excel Add-In” (MATLAB Compiler SDK)

Change the Server Configuration

Before you can run an Excel add-in on a MATLAB Production Server instance, you must 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 must 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**.

See Also

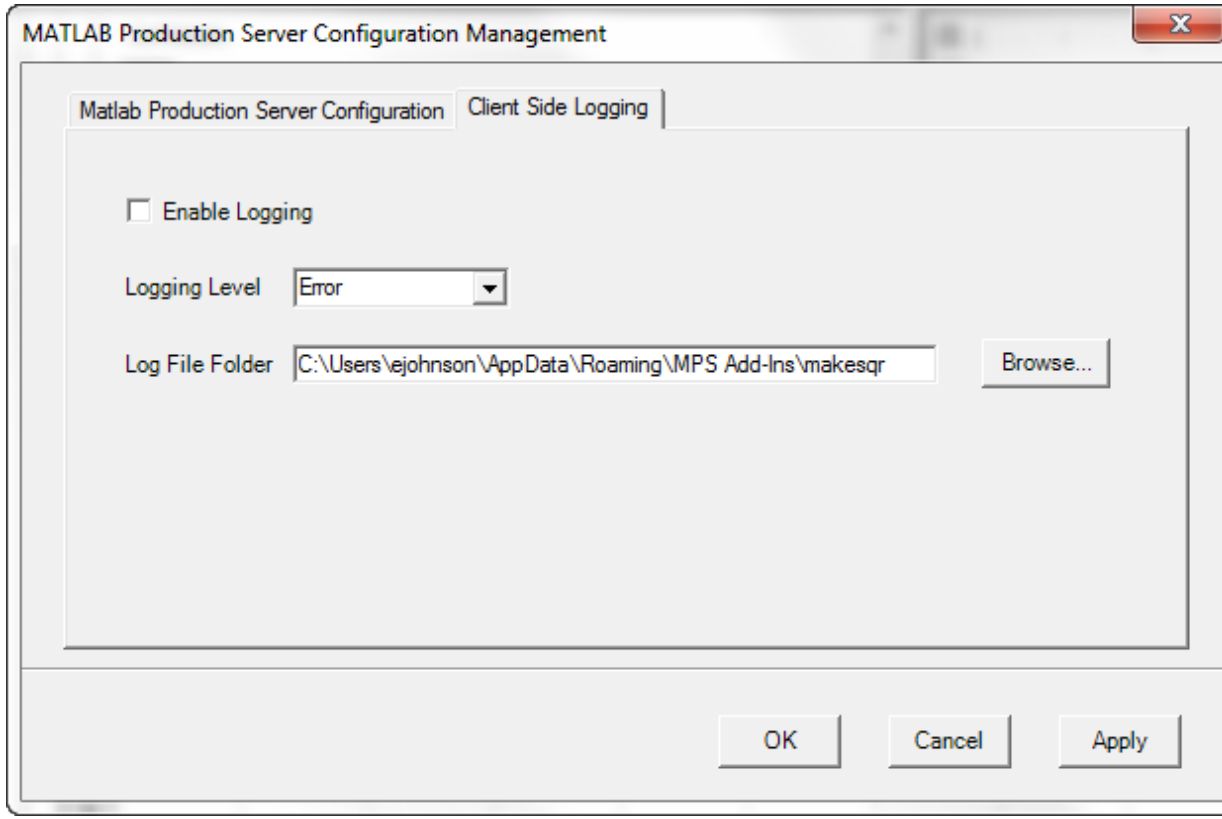
More About

- “Install the Excel Add-In” on page 1-8

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 various 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.