Connectivity Guide

KEPServerEX®, DDE, and Excel

September 2018
Ref. 1.004

Table of Contents

1. Overview

This guide demonstrates how to establish a connection between Microsoft Excel and the KEPServerEX data server. Before continuing with this tutorial, configure a server project. Select the appropriate driver and settings or run the **Simulation Driver Demo** included with KEPServerEX. The Simulation Driver Demo project is used for all examples in this tutorial.

2. Requirements

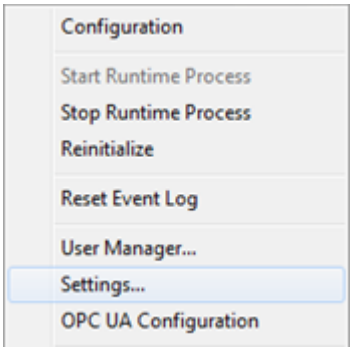
The following are necessary to perform DDE connections to the server:

- In the Installation Features dialog of the server installation, select **DDE (Dynamic Data Exchange)** from the Native Client Interfaces tree.
- A valid DDE client (such as any version of Excel) must be used.

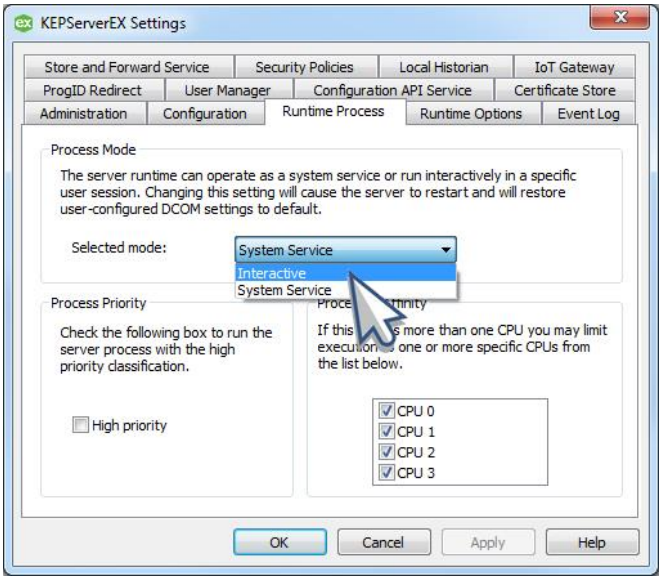
3. Configuring the Server for DDE Connectivity

3.1 Change the Server to Interactive Mode

Although the server's default process mode is System Service, communication between Excel and the server are simpler in Interactive mode. To change the process mode to Interactive mode, follow these steps:



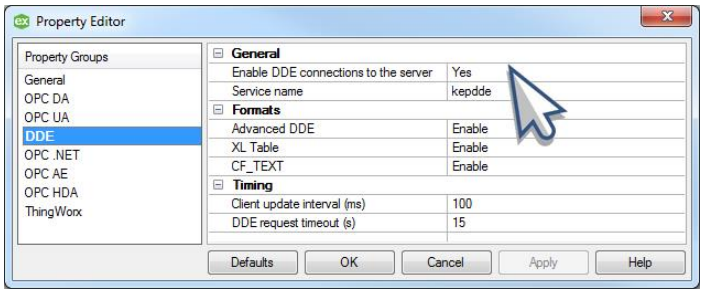
1. Right-click on the **Administration** menu and select **Settings...**.
2. Open the **Runtime Process** tab and locate the **Process Mode** area.
3. Use the Selected Mode drop-down menu to select **Interactive**.
4. Click **OK**.



3.2 Configure the DDE Properties

The server installation disables DDE by default because it can slow the server's performance and cause security issues when it is enabled unnecessarily. It is required for connection to a database, so follow these steps to enable DDE:

1. Choose **File | Project Properties** and select the **DDE** tab.
2. Under **General**, select **Yes** to enable for DDE connections to the server.



- **Note**: If using a local DDE connection, leave Enable Net DDE disabled.

3. In Service Name, keep the default name "kepdde".
4. In the Formats area, select **Enable** for all three before continuing.
5. Leave the remaining properties at the default settings.
6. Click **OK**.
7. Restart the server so the changes take effect.

- *For more information on the DDE parameters, consult the online help system.*

4. Writing and Accessing DDE Data in Excel

4.1 Write and Display DDE Data

A DDE connection in Excel requires an application name, topic, and an item. Only one topic name is required in KEPServerEX, however: all server data can be accessed using "_ddedata" as the DDE topic. This global server topic allows access to every tag or address within the server, regardless of the device or tag group under which it is built. With this topic in a link, a full path identifier must be provided for the intended item. The path identifier must include tag groups, sub groups, channel, and device names.

- **Note**: In this tutorial, Microsoft Office 2007 is used.

To display DDE data in Excel, enter the following formula into each cell:

=<Application>|<Topic>!<Item>

where each bracketed variable is replaced by the specifics in the environment and data.

A valid cell formula for the Simulation Driver Demo project would be:

=kepdde|_ddedata!Channel1.Device1.Tag1

- Long item names can be simplified through the server's alias feature. To create an alias in the Configuration; locate the **Aliases** node in the tree view, right-click and choose **New Alias**.
- *For more information, refer to Reading Arrays from KEPServerEX into Excel.*

4.2 Access Data Dynamically

To dynamically access data, create a cell formula using a device address instead of a tag. Dynamic tags can only be added at the device level; however, dynamic tags added to tag groups are rejected.

- *For more information on dynamic vs. static tags refer to the server help file.*

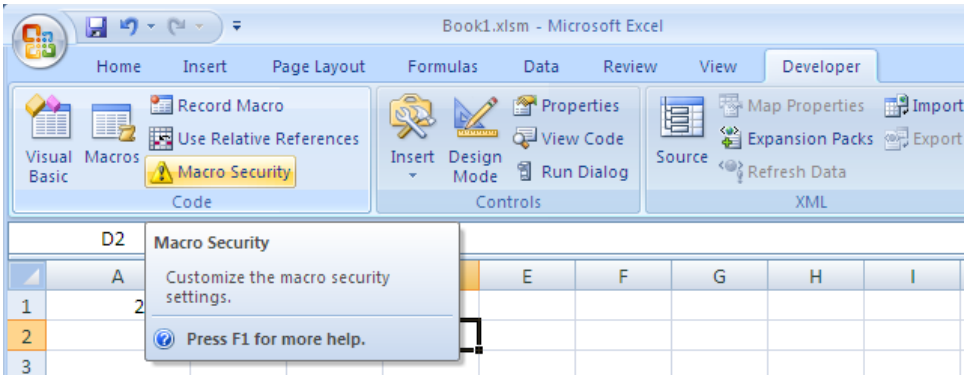
5. Writing Data from Excel to the Server

5.1 Enable Macros in Excel

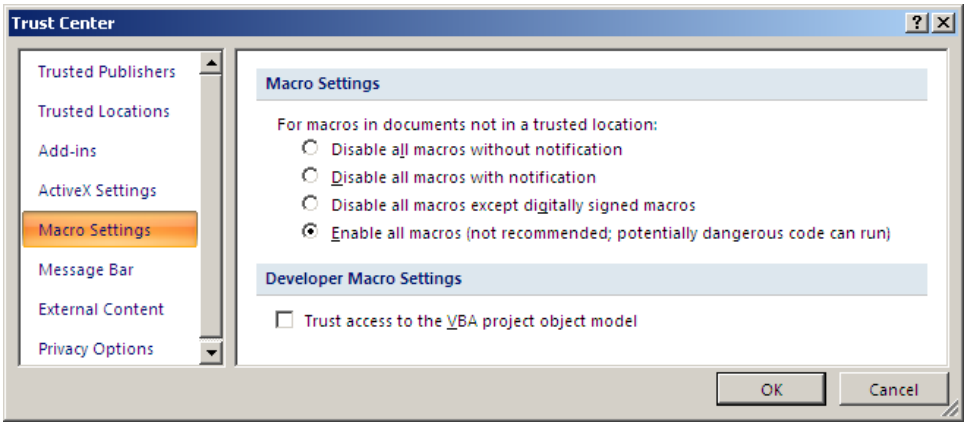
To write data from Excel to the server, an Excel macro must be created that executes a DDE Poke. Since Office 2007 disables Excel macros by default, users must change the parameter before a macro can be created.

- *For more information, refer to the instructions below.*

1. In Excel, open the **Developer** tab.
2. Click **Macro Security** (located in the **Code** ribbon bar group).
3. In Trust Center, select **Macro Settings**.



4. Select **Enable all macros**.
5. Click **OK** to exit.



5.2 Start the Microsoft Excel Project

The following example shows to access two tags from the Simulation Driver Demo. Cells are referred to by row and column. For more information, follow the instructions below.

1. Locate Column A of Sheet1, where you will enter two consecutive DDE read links as shown below.
2. In Cell(A1), enter the following formula for reading server data:

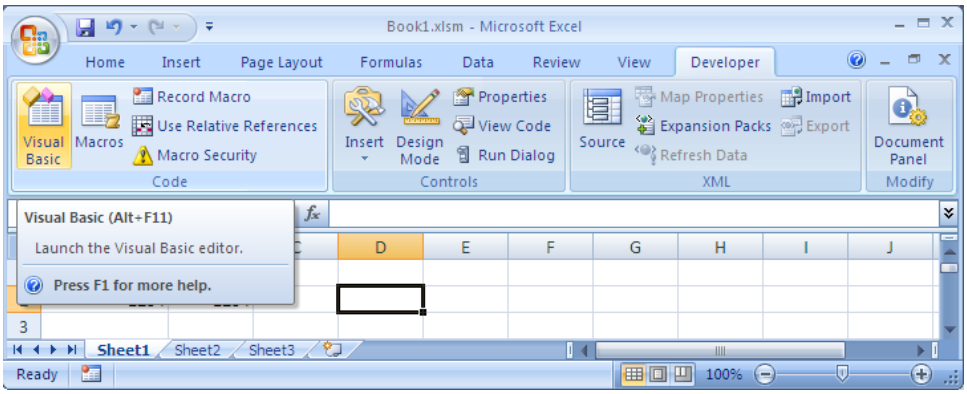
   =kepdde|_ddedata!Channel1.Device1.Tag1

3. In Cell(A2), enter the following formula for reading server data:

   =kepdde|_ddedata!Channel1.Device1.Tag2

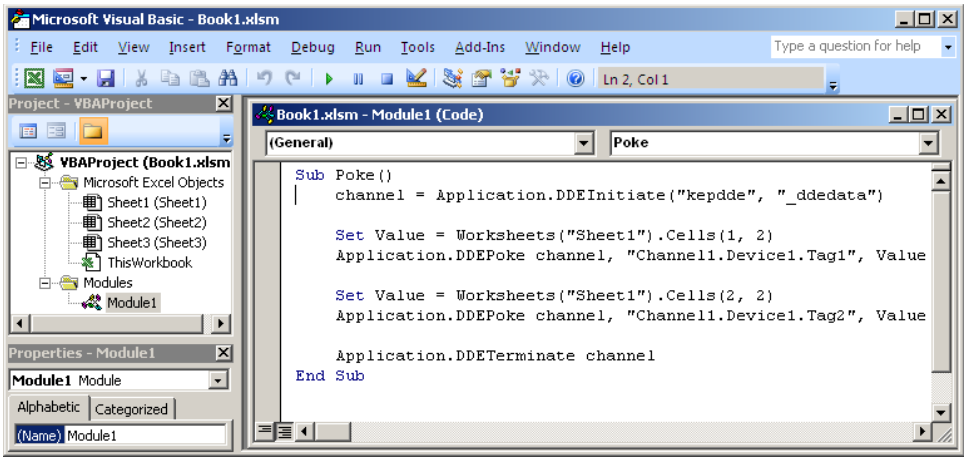
4. Select the **Developer** tab.

5. Open Visual Basic Editor.



6. In a Visual Basic module window, enter the following code:

```
Sub Poke()
    channel = Application.DDEInitiate("kepdde", "_ddedata")
    Set Value = Worksheets("Sheet1").Cells(1, 2)
    Application.DDEPoke channel, "Channel1.Device1.Tag1", Value
    Set Value = Worksheets("Sheet1").Cells(2, 2)
    Application.DDEPoke channel, "Channel1.Device1.Tag2", Value
    Application.DDETerminate channel
End Sub
```

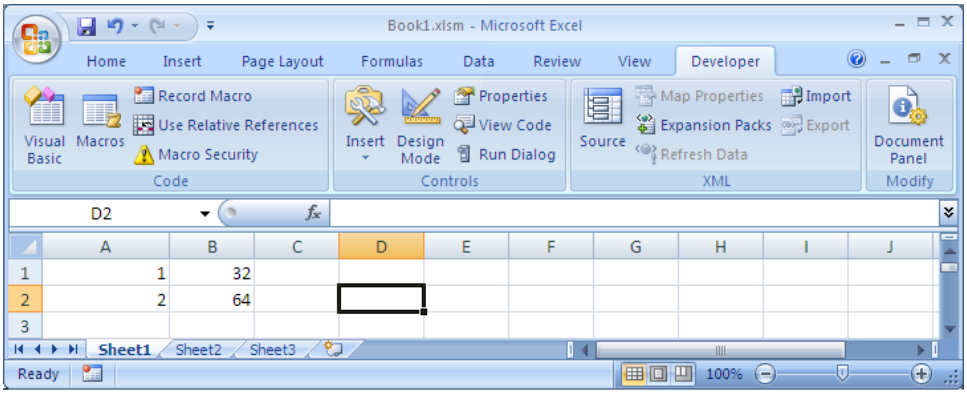


Note: The Visual Basic Command Cells property uses Row and Column number references for example Cell A1 would be referred to in the command with the syntax of Worksheets("Sheet1").Cells.(1,1).

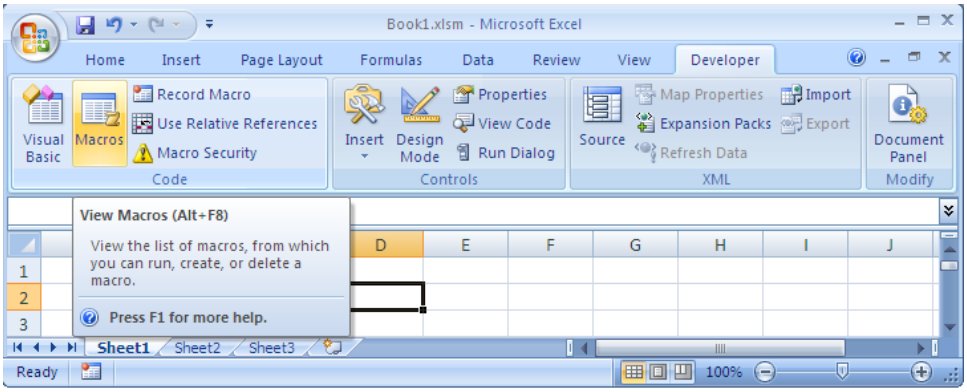
7. Choose **File | Save** and name the code module "Poke".

8. Exit Visual Basic and return to the Excel spreadsheet. The macro is ready to be tested.

9. Enter some numbers in Cell(B1) and Cell(B2). Click on an unused cell to make sure that the last value is set.

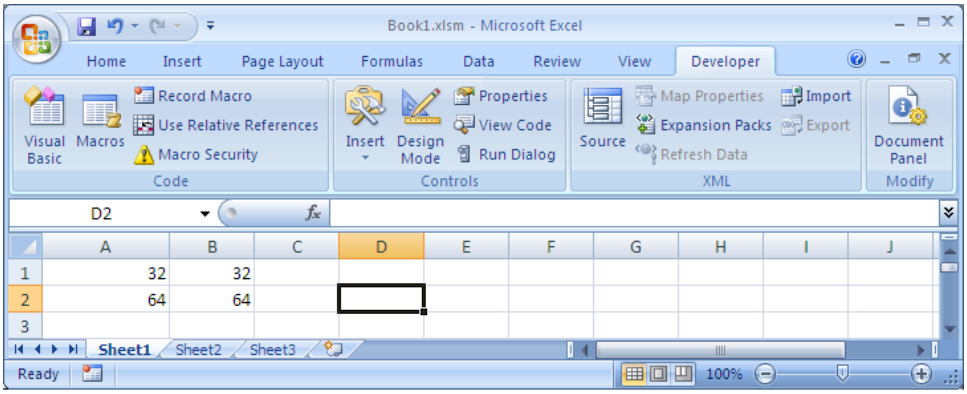


10. In the Developer tab, choose **Macros | View Macros**.



11. In the Macro dialog box, select the "Poke" macro and click **Run**.

12. Verify that the cells in column A contain new values.



6. Reading and Writing Arrays from KEPServerEX with Microsoft Excel

Arrays provide fast data access, especially in comparison to data referenced one address at a time (as in the examples above). Drivers that consecutively order data of the same data type should use arrays to get data into Excel. Once in Excel, the data can be moved and displayed as needed.

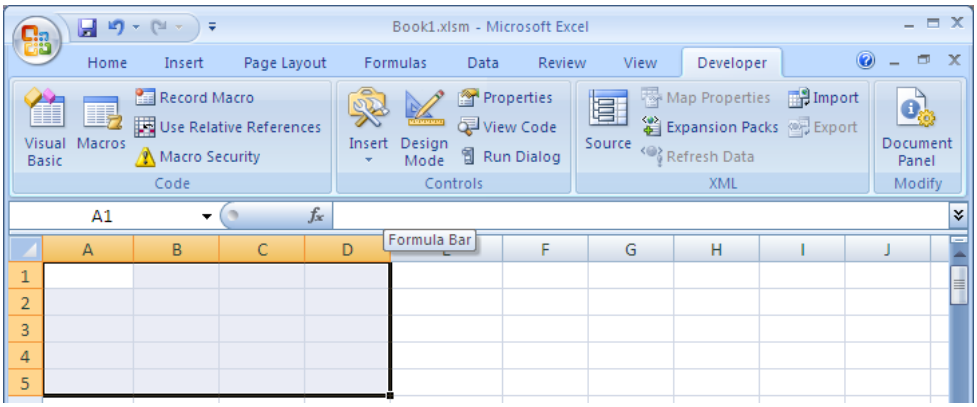
Likewise, it is possible to improve performance of writes by writing an array rather than writing to individual items one at a time.

6.1 Reading an Array from the Server

This example shows how to dynamically access the Simulator tag R100[5][4] through Excel. Follow the instructions below:

- Many of the device drivers available for KEPServerEX support arrays.

1. In Excel, mark a range of cells by left-clicking and dragging the cursor over an area of cells that is equal to the array dimensions. For this tutorial, select cells A1 to D5.



2. Enter the following formula into the Formula Bar.

- Do not enter the formula into a cell.

  =kepdde|_ddedata!'Channel1.Device1.R100[5][4]'

- **Notes**:

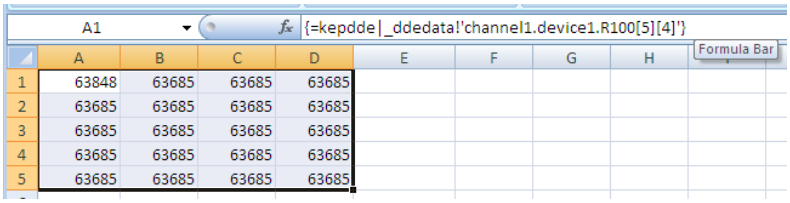
  - In Excel, escape characters must be used around any formula string that contains square brackets or other special characters (such as @). This example uses a dynamic address reference in the server. Therefore, the quotation / foot / tick marks included in the formula above are essential.

  - This example uses dynamic tag addressing, which shows the array dimensions. Static tags can also be created for arrays; the address of the tag in the server specifies the starting address of the array data and its dimensions. When using a static tag that is an array, the dimensions do not need to be specified. If the array tag is named Array5_, the formula would be '=kepdde|_ddedata!Channel1.Device1.Array5_4' for example.

3. Press the keys: Shift, Control (Ctrl), and Enter simultaneously. This combination informs Excel that an array element has been entered.

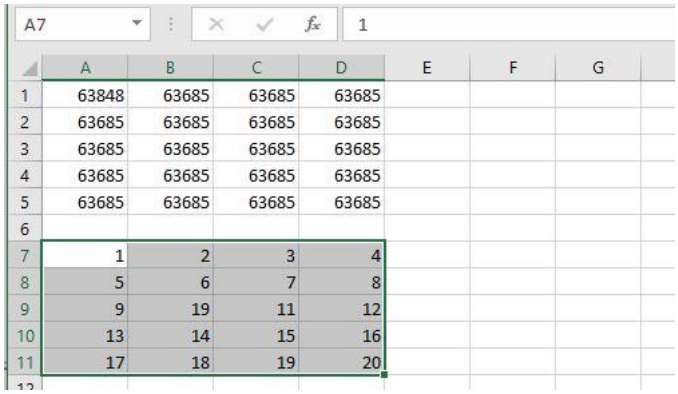
Note: If these keys are not pressed simultaneously, only a single item will be displayed in the first cell.

4. If the formula and command worked correctly, data changes from the server should be visible in the array's cells.



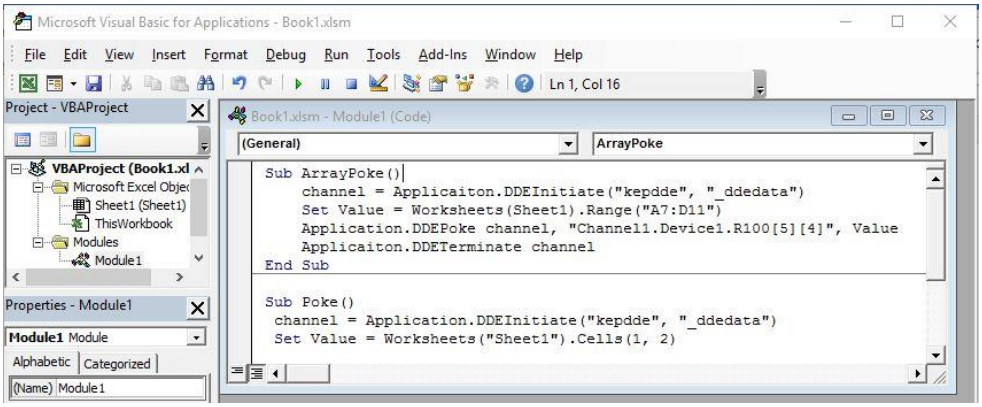
6.2 Writing an Array to the Server

1. In the Excel worksheet select a range of cells that is the same size as the one that was used in the read example and populate the cells with values. In this example a range of cells from A7 to Dll were chosen.

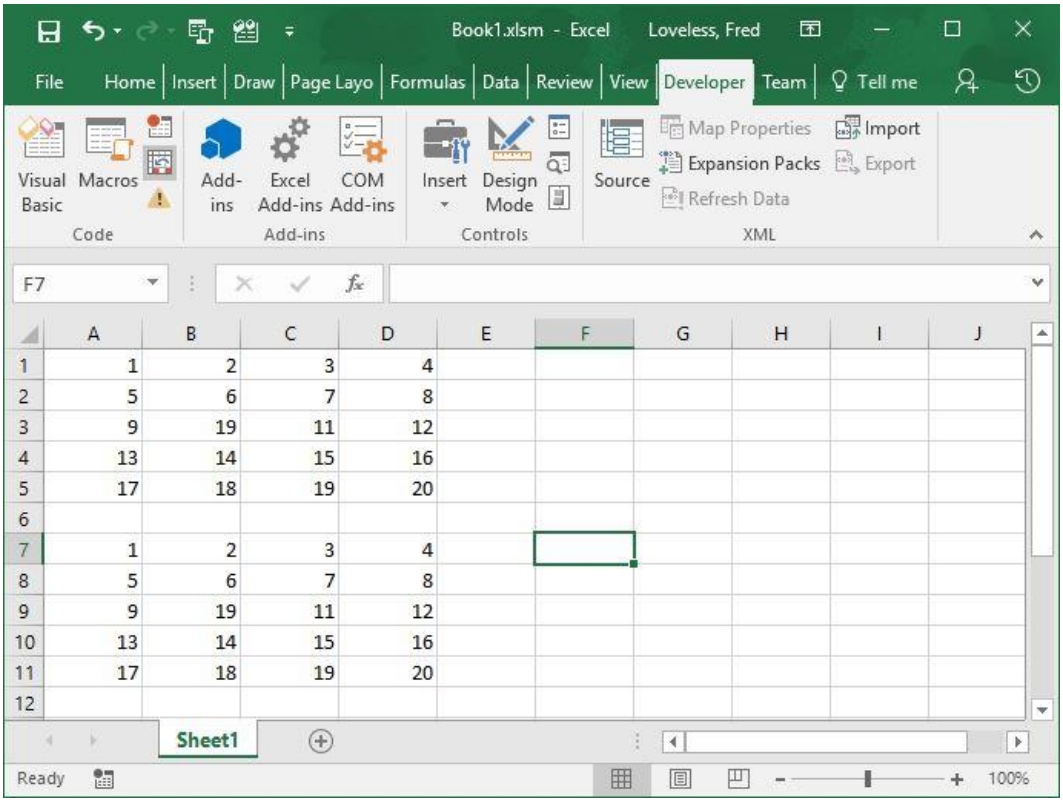


2. Select the **Developer** tab.
3. Open the Visual Basic Editor.
4. In a Visual Basic module window, enter the following code:

```
Sub ArrayPoke()
    channel = Applicaiton.DDEInitiate("kepdde", "_ddedata")
    Set Value = Worksheets(Sheet1).Range("A7:D11")
    Application.DDEPoke channel, "Channel1.Device1.R100[5][4]", Value
    Applicaiton.DDETerminate channel
End Sub
```



5. Choose **File | Save** and save the code module.
6. Exit Visual Basic and return to the Excel spreadsheet. The macro is ready to be tested.
7. In the Developer tab, choose **Macros | View Macros**.
8. In the Macro dialog box, select the “ArrayPoke” macro and click **Run**.
9. Verify that the cells used to display the array when it is read now show the values that were just written.



	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4						
2	5	6	7	8						
3	9	19	11	12						
4	13	14	15	16						
5	17	18	19	20						
6										
7	1	2	3	4						
8	5	6	7	8						
9	9	19	11	12						
10	13	14	15	16						
11	17	18	19	20						
12										