Importing Netbiter Argos data

into Microsoft Excel

APPLICATION NOTE

SCM-1202-041-EN 2.0 ENGLISH

Connecting Devices™

Important User Information

Liability

Every care has been taken in the preparation of this document. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the USA and other countries.

eWON®, Netbiter®, and Netbiter Argos® are registered trademarks of HMS Industrial Networks AB. All other trademarks mentioned in this document are the property of their respective holders.

Table of Contents

Page

1 Preface 3

1.1 Document history 3

1.2 Document Conventions 3

2 Importing Data into Microsoft Excel 4

2.1 Creating an API Access Key 4

2.2 Finding the API Calls for Aggregated Log History 4

2.3 Importing Aggregated Log Data into Microsoft Excel 6

3 Using VBA Macros to Import Data 7

3.1 Displaying the Developer Tab in Microsoft Excel 7

3.2 Enabling Macros in Microsoft Excel 7

3.3 Creating Macros with the VBA Editor 8

3.4 VBA Code Examples 9

3.5 Connecting a Macro to a Toolbar Button 12

This page intentionally left blank

1 Preface

This document describes how to import data from Netbiter Argos into a Microsoft Excel spreadsheet using the the Netbiter Argos REST API.

For more information about the Netbiter Argos API, please visit www.netbiter.com/support.

For more information about how to work with Visual Basic for Applications (VBA) macros and how to create graphs and charts from data in spreadsheets, please refer to the documentation for Microsoft Excel and related resources on the Internet.

1.1 Document history

Version	Date	Description
1.10	2013-05-03	General update
1.20	2013-12-09	Added chapter on VBA macros
2.0	2018-02-14	Updated for Excel 2013 and current Argos version New brand and document ID

1.2 Document Conventions

Bold typeface is normally used to indicate user interaction.

```
Monospaced text is used to indicate program code and other
kinds of data input/output such as configuration scripts.
```

Cross-reference within the document: *Document Conventions, p. 3*

External link (URL): www.hms-networks.com

This is additional information which may facilitate installation and/or operation.

This instruction must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk.

Caution
This instruction must be followed to avoid a risk of personal injury.

WARNING
This instruction must be followed to avoid a risk of death or serious injury.

2 Importing Data into Microsoft Excel

To carry out the examples in this document you need administrator access to a Netbiter Argos account with the Manage and Analyze service, and a remote system which is logging data and has some configured alarms and visualization parameters.

The examples are based on the US English version of Microsoft Excel 2013. You may have to refer to the documentation for Microsoft Excel if using a different version.

2.1 Creating an API Access Key

1. Log in as administrator to the Argos account and select **Account ►External data access**.
2. Click on **Add** to create an *API access key*. You will be prompted to set the scope and access level of the key. For these examples the default settings should work.
3. Click on **Save** to save the new access key.

You can now select and copy the access key to the clipboard to use it in the next step.

2.2 Finding the API Calls for Aggregated Log History

1. Enter the following URL in a web browser: `https://api.netbiter.net/operation/v1/rest/xml/system/`**`[systemid]`**`/log/config?accesskey=`**`[accesskey]`**

   Replace **[systemid]** with the System ID of the remote system, and **[accesskey]** with the API access key that you created in the previous step.

> *The System ID can be found under **Management – All systems** in Netbiter Argos.*

   The browser window will now display the configuration of all available log parameters for the system in XML format.

2. Locate the parameter that you wish to read data from and note its **<id>** attribute.

Fig. 1 Available log parameters

3. Enter the following URL in the browser: `https://api.netbiter.net/operation/v1/rest/xml/system/`**`[systemid]`**`/log/`**`[parameterid]`**`/aggregated/hour?accesskey=`**`[accesskey]`**`&limitrows=24`

   Replace **[parameterid]** with the value from the **<id>** attribute.

   Replace **[systemid]** and **[accesskey]** in the URL as explained in the previous steps.

   Adding `limitrows=24` at the end of the query means that only the last 24 hours of average data will be imported.

4. The response from this URL should be an XML page similar to this:

Fig. 2 Aggregated log data

This URL will be used in the following example to import data into Microsoft Excel.

2.3 Importing Aggregated Log Data into Microsoft Excel

1. Open a new spreadsheet and select the **Data** tab, then click on **From Web**.
2. Paste the URL from the previous example into the address field of the Web Query browser in Microsoft Excel and press **Enter**.

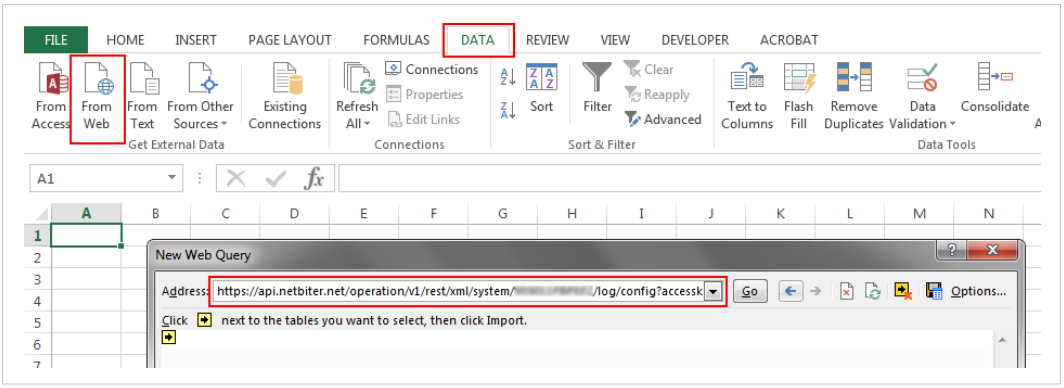

Fig. 3 Entering the URL query

3. The response from Argos will be presented in XML format.

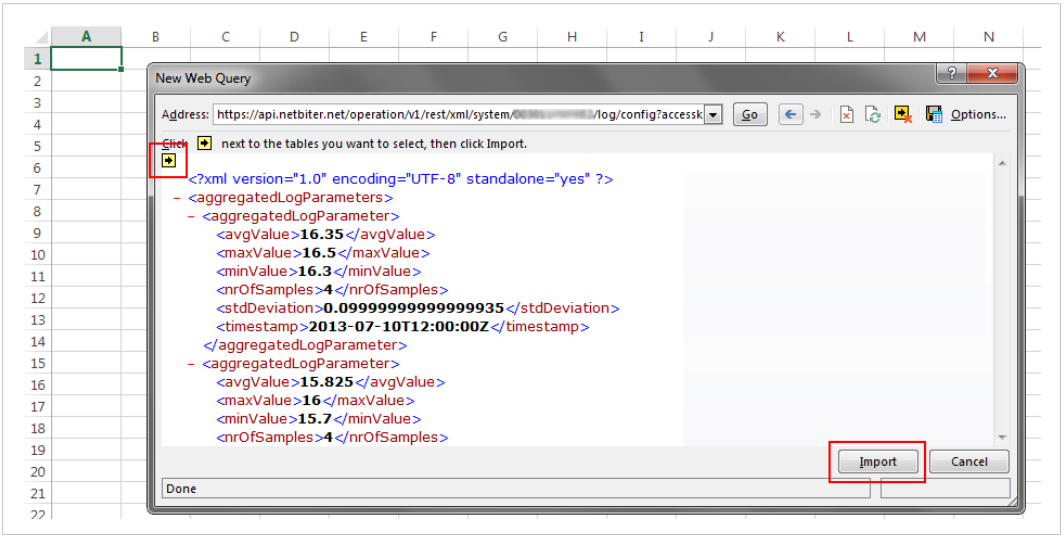

Fig. 4 Argos response in XML format

4. Click on the arrow in the top left corner of the browser page to select the complete XML structure. The arrow will change into a check mark.
5. Click on **Import**, then click on **OK** in the following dialogs to complete the operation. The spreadsheet should now contain the data from Netbiter Argos.

avgValue	maxValue	minValue	nrOfSamples	stdDeviation	timestamp
16,35	16,5	16,3	4	0,1	2016-12-28-10T12:00:00Z
15,825	16	15,7	4	0,15	2016-12-28-10T11:00:00Z
16,225	16,3	16	4	0,15	2016-12-28-10T10:00:00Z
16,5	16,8	16,3	4	0,21602469	2016-12-28-10T09:00:00Z
17,05	18	16,5	4	0,714142843	2016-12-28-10T08:00:00Z
17,125	17,5	17	4	0,25	2016-12-28-10T07:00:00Z
17,4	17,6	17,2	4	0,163299316	2016-12-28-10T06:00:00Z
17,125	17,5	16,8	4	0,330403793	2016-12-28-10T05:00:00Z
16,575	16,8	16,2	4	0,287228132	2016-12-28-10T04:00:00Z
15,45	15,8	15,2	4	0,264575131	2016-12-28-10T03:00:00Z

Fig. 5 Imported log data

3 Using VBA Macros to Import Data

Tasks in Microsoft Excel can be automated using macros that are created with Visual Basic for Applications (VBA). This chapter describes how to create a macro that imports Netbiter Argos data into a spreadsheet, and how to assign this macro to a toolbar button.

3.1 Displaying the Developer Tab in Microsoft Excel

If the **Developers** tab in Microsoft Excel is not visible, do as follows:

1. Go to **File** – **Options** – **Customize Ribbon**.
2. In the **Customize the Ribbon** dropdown, select **Main Tabs**.
3. Check the **Developer** checkbox and click on **OK**.

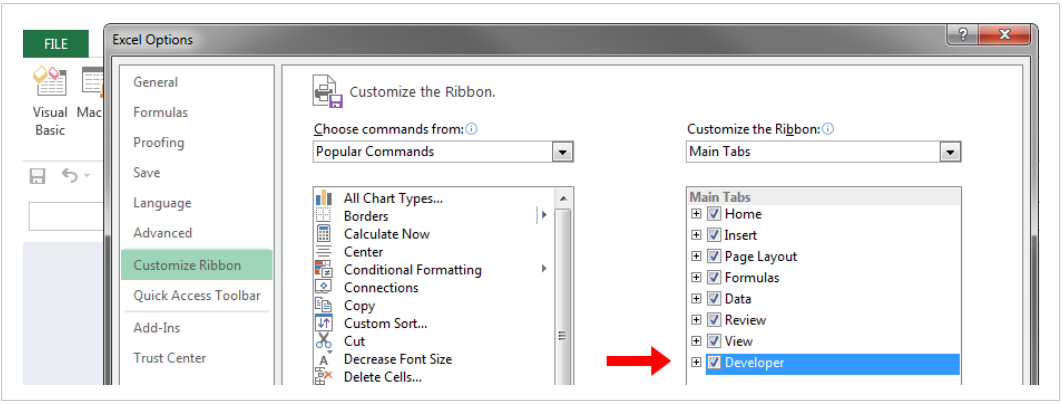

Fig. 6 Excel Options dialog

3.2 Enabling Macros in Microsoft Excel

1. On the **Developer** tab, click on **Macro Security**.

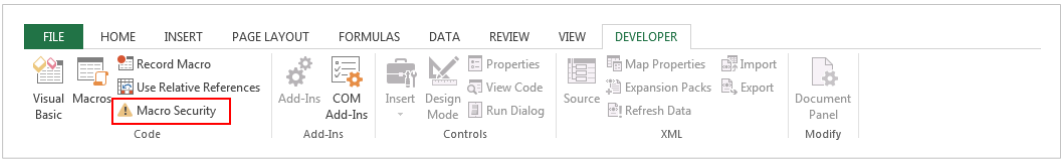

Fig. 7 Developer tab

2. In **Macro Settings**, select **Enable all macros** and click on **OK**.

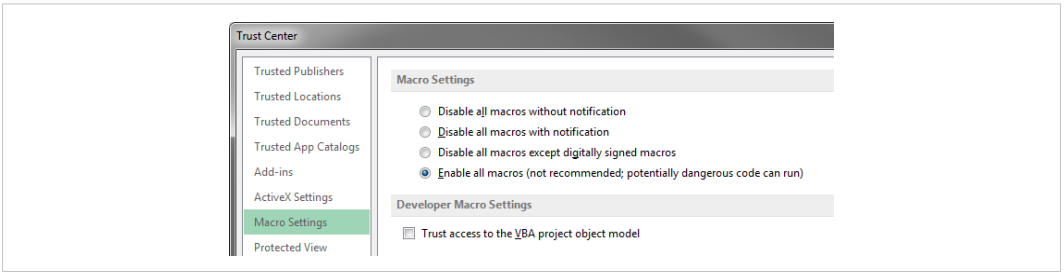

Fig. 8 Macro Settings

> **!** This option makes Microsoft Excel less secure and should not be enabled when you are not working with macros.

3.3 Creating Macros with the VBA Editor

> The functionality of the VBA editor depends on the version of Microsoft Excel and the installed libraries. See the documentation for Microsoft Excel.

The preparations to make an API call from a macro are the same as when importing XML data manually into Excel. See *Importing Data into Microsoft Excel, p. 4*.

Basic Steps

1. Open a new spreadsheet in Microsoft Excel.
2. On the **Developer** tab, click on **Visual Basic**.

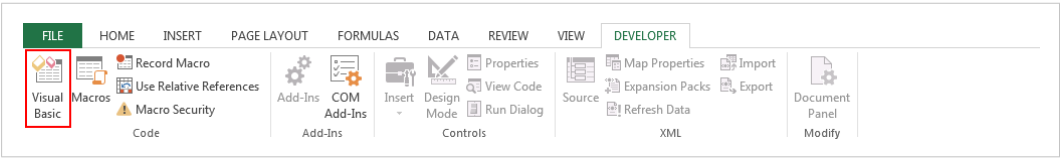

Fig. 9 Developer tab

3. On the **Insert** menu in the VBA editor window, select **Module**.

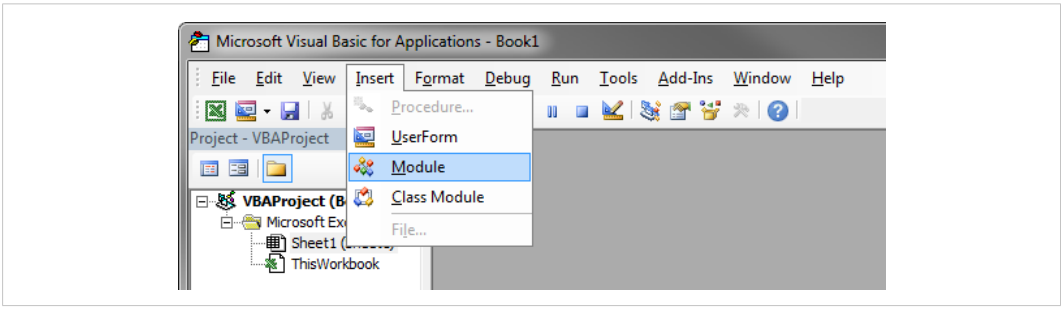

Fig. 10 Visual Basic editor

4. Write or paste the code to use for this macro into the editor.
   See *VBA Code Examples, p. 9*.
5. Click on the ► button or press **F5** to run the macro.
6. On the **File** menu, select **Close and return to Microsoft Excel**.

3.4 VBA Code Examples

3.4.1 Using GET and POST to Access Data

This macro will get the name of all the alarms that have occurred since a specific date. Replace the *accesskey* and *startdate* values in the code example with the actual access key for your system and the desired start date.

```
Sub GetAllUpdatedAlarms()

Dim zip As String
Dim query As String
Dim apiResult As New MSXML2.DOMDocument
Dim apiService As New MSXML2.XMLHTTP

zip = Range("B1").Text

'assemble query string
  query = "https://api.netbiter.net/operation/v1/rest/xml/updated/
  alarms?accesskey=123456789ABCDEF&startdate=2016-12-01"

'create HTTP request to query URL - make sure to have
'that last "False" there for synchronous operation
  apiService.Open "GET", query, False

'send HTTP request
  apiService.send

'parse result
  apiResult.LoadXML (apiService.responseText)

'extract result contents into appropriate cells
  Set xmlNodeList = apiResult.getElementsByTagName("alarms")

  j = 1
  For Each Node In xmlNodeList
    Cells(j, 1).Value = Node.SelectSingleNode("name").Text
    j = j + 1
  Next
End Sub
```

We can replace the *accesskey* string in the query above with a call to a function which retrieves the access key from the system if the user is authorized.

```
https://api.netbiter.net/operation/v1/rest/xml/updated/
alarms?accesskey="& accessKey &"&startdate=2016-12-01
```

The following VBA code will retrieve the access key and put it into the variable **`accessKey`** after the user has entered the correct login credentials:

```
Sub GetAccessKey()

Dim strName As String
Dim myHTTP As MSXML2.XMLHTTP
Dim myDom As MSXML2.DOMDocument
Dim myXML As String
Dim apiResult As New MSXML2.DOMDocument
Dim accessKey As String
Dim URL As String

  strName = InputBox(Prompt:="Enter username.", Title:="Username")
  passName = InputBox(Prompt:="Enter password.", Title:="Password")

  If strName = vbNullString Or _
    passName = vbNullString Then

    Exit Sub

  Else

    Set myHTTP = CreateObject("msxml2.xmlhttp")

    URL = "https://api.netbiter.net/operation/v1/rest/xml/user/
authenticate"

    myXML = "<userCredentials><userName>" &
    strName & "</userName><password>" & passName &
"</password></userCredentials>"

    myHTTP.Open "post", URL, False
    myHTTP.setRequestHeader "Content-Type", "application/xml"
    myHTTP.send (myXML)

    If myHTTP.Status = 200 Then
      apiResult.LoadXML (myHTTP.responseText)
      Set usersList = apiResult.getElementsByTagName("userAccess")

      For Each Node In usersList
      accessKey = Node.SelectSingleNode("accessKey").Text
      MsgBox accessKey
      Next
    Else
      MsgBox myHTTP.responseStream
    End If

  End If
End Sub
```

3.4.2 Using PUT to Set a Value from Microsoft Excel

The following example shows how to use a macro to control the relay output on a Netbiter gateway via the API by setting a value in Microsoft Excel. Replace the *system* and *accesskey* values in the code example with the actual values for your system.

In this example we have defined a tag for the parameter to use in the API call. You can also use the database ID for the parameter.

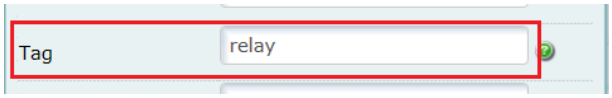

Fig. 11 Adding a tag name

```
Sub SetValue()

Dim strName As String
Dim myHTTP As MSXML2.XMLHTTP
Dim myXML As String
Dim apiResult As New MSXML2.DOMDocument
Dim accessKey As String
Dim URL As String

strValue = InputBox(Prompt:="Enter a value.", Title:="Set value")

  If strValue = vbNullString Then

    Exit Sub

  Else

  Set myHTTP = CreateObject("msxml2.xmlhttp")

URL = "https://api.netbiter.net/operation/v1/rest/xml/system/
003011FBFC18/live?accesskey=123456789ABCDEF"

myXML = "<writeLiveParameters><writeLiveParameter><id>relay</id>
<value>" & strValue & "</value></writeLiveParameter>
</writeLiveParameters>"

    myHTTP.Open "put", URL, False
    myHTTP.setRequestHeader "Content-Type", "application/xml"
    myHTTP.send (myXML)
    Data = myHTTP.responseText

    If myHTTP.Status = 200 Then
      MsgBox "OK"
    Else
      MsgBox myHTTP.responseStream
    End If

  End If

End Sub
```

3.5 Connecting a Macro to a Toolbar Button

Macros in Microsoft Excel can be connected to objects in a spreadsheet or to interface objects such as toolbar buttons and menus. This example shows how to assign the macro you created in the previous example to a new button in the Quick Access Toolbar.

1. Go to **File** – **Options** – **Quick Access Toolbar**.
2. In the **Choose commands from** dropdown, select **Macros**.
3. Select the macro in the list of macros (left column) and click on **Add**.
4. Select the added macro in the list of commands (right column) and click on **Modify Button**.

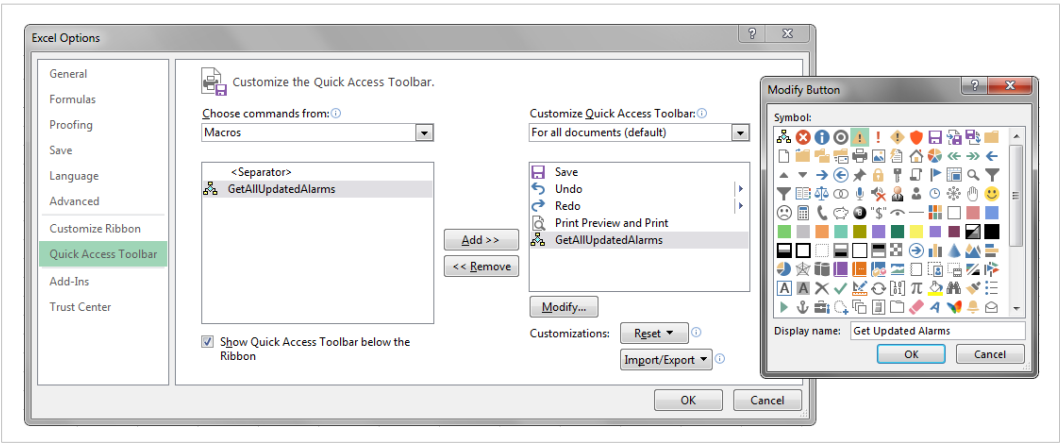

Fig. 12 Excel Options dialog

5. Enter a **Display name** and choose a **Symbol** (icon) for the macro, then click **OK**.

The Quick Access Toolbar will now contain a new button that will run the macro when clicked. The name of the macro will be shown when the mouse cursor is over the button.

Fig. 13 New button in Quick Access Toolbar

This page intentionally left blank

© 2018 HMS Industrial Networks AB
Box 4126
300 04 Halmstad, Sweden

info@hms.se
SCM-1202-041-EN 2.0.7426 / 2018-02-14