



Configuring CitectSCADA SNMP projects with MIB2CIT

A reference for CitectSCADA Customers

support@citect.com

Revision 1.00

Created 28-12-2009

Document revision history

Please update this table whenever you make an update to the document. Don't forget to increment the Cover Page.

Revision author	Changes made	Date	New version No.

Customer guide to basic CitectSCADA SNMP configuration

Introduction

As well as the monitoring and control of traditional IO on a SCADA network, CitectSCADA can be configured to communicate with SNMP Version 1¹ enabled network devices such as switches, routers, servers, and even PLCs /RTUs with SNMP capability.

To assist with the configuration of CitectSCADA SNMP projects, Citect provide a tool called MIB2CIT which guides the customer through creating variable tags, device IDs and unique device addresses.

¹ At the time of writing this document, Citect does not support SNMP versions II & III

This document is made up of the following sections:

1. Creating a simple 2 tag test project using CitectSCADA
2. Using an application called MIB2CIT to configure your test SNMP project
3. Using an alternative tool to MIB2CIT called GetIF to accomplish the same results
4. Terms used in this document
5. Useful downloads

Intended Audience

This document has been written for CitectSCADA customers to help them set up CitectSCADA SNMP projects.

Table of Contents

Table of Contents.....	5
1 Building a simple SNMP test project in CitectSCADA	6
1.1 Preparing the CitectSCADA project.....	6
1.2 Installing & configuring MIB2CIT.....	7
1.3 Verifying SNMP is working correctly before configuring CitectSCADA.....	8
1.4 Creating CitectSCADA variable tags with MIB2CIT	11
1.4.1 Assigning tag suffixes to our OIDs and adding tags to our CitectSCADA Project.....	12
1.4.2 Verifying your tags have been exported to CitectSCADA using SNMPVARS.DBF and VARIABLE.DBF	13
1.4.3 Testing your tags in CitectSCADA.....	14
2 Useful Terminology	16

Part I: Using MIB2CIT to set up and configure a test SNMP project

1 Building a simple SNMP test project in CitectSCADA

We'll create a CitectSCADA SNMP project with two simple variable tags. We will use our own PC's as SNMP test devices and configure CitectSCADA to read two pieces of information, called objects, from our PC's using the CitectSCADA SNMPIO driver.

The two objects we will monitor are listed below:

1. The name of our PC (SysName)
2. The duration our PC has been running for (SysUptime)

You can think of SNMP object identifiers, or OIDs, like point addresses in a PLC or RTU.²

1.1 Preparing the CitectSCADA project

Follow the steps below to install the CitectSCADA SNMPIO driver from the Citect [website](#) and create your SNMP device (your PC) in your CitectSCADA project.

1. Install the latest SNMPIO driver from the [Driver Web](#)
2. Create a new project called SNMPIOProject using the CitectSCADA Explorer
3. Save your SNMPIOProject project in your root directory
4. Go to the Express Wizard in the CitectSCADA project builder and select the SNMPIO driver. Enter the IP address (127.0.0.1) and Community String (public) of your test machine, 127.0.0.1 public.

In the real world these values will be your actual I/O device IP address and community string.

5. That's it!

Notice anything odd? We haven't created any CitectSCADA variable tag information. Our friend, MIB2CIT, in the following sections, will be used to create these for us.

² For more information on SNMP OIDs, click here

1.2 Installing & configuring MIB2CIT

We can use MIB2CIT to retrieve information on a device object such as System Uptime or System Name before we even touch the CitectSCADA configuration – this is handy for testing SNMP is working in the first place.

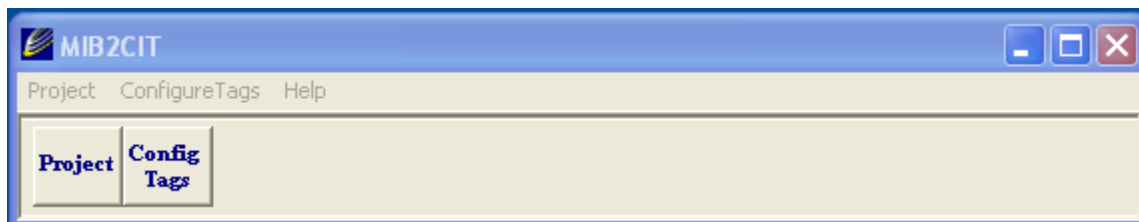
- Extract and install MIB2CIT to your root directory C:\
 - **Pathname lengths must be less than 99 characters**
One of the bugs with MIB2CIT is KB'd in [Q5564](#). The MIB2CIT.ini file references your project path name, go ahead and open it up in a text editor, you will see the following:

```
[Project]
ProjDir=C:\SNMPProj
```

The path name must be less than 99 characters or MIB2CIT. This is why we saved our project to the root directory.

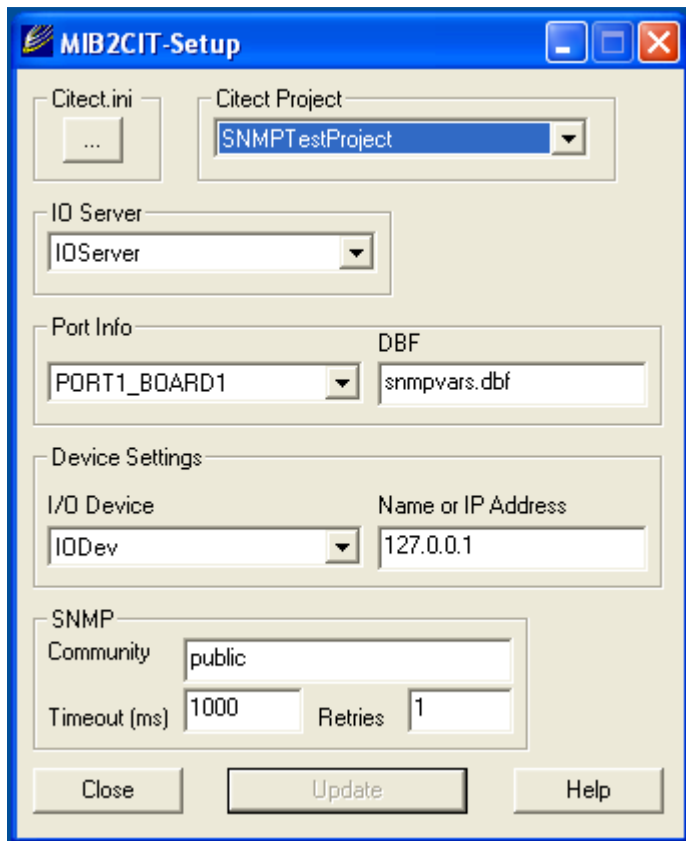
- Launch the MIB2CIT.exe and click “**Project**”.

Figure 1: Showing MIB2CIT landing page



- Click Citect.INI in the top left and browse to your Citect.ini file

Figure 2: MIB2CIT Set Up



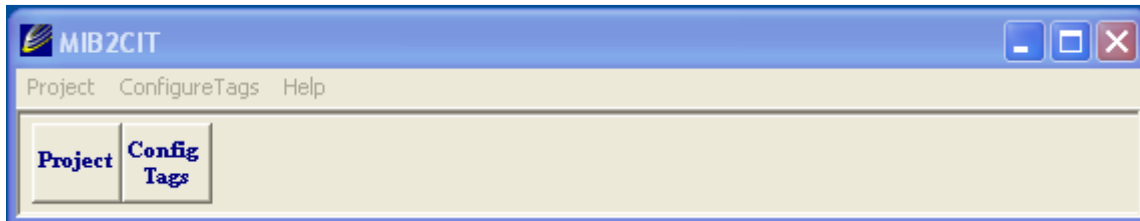
- Next, select your project from the drop down **Citect Project** box
- The details you entered in to your CitectSCADA **SNMPProject** earlier will be populated, together with reference to the SNMPVARS.dbf file - we'll discuss what this file does later on.
- Click **Close**

1.3 Verifying SNMP is working correctly before configuring CitectSCADA

Next, we will run a simple test to read (GET) the System Uptime of your desktop PC using SNMP. This test has nothing to do with CitectSCADA, but rather it is used to test connectivity at a network layer and that SNMP polling is working before we start testing CitectSCADA.

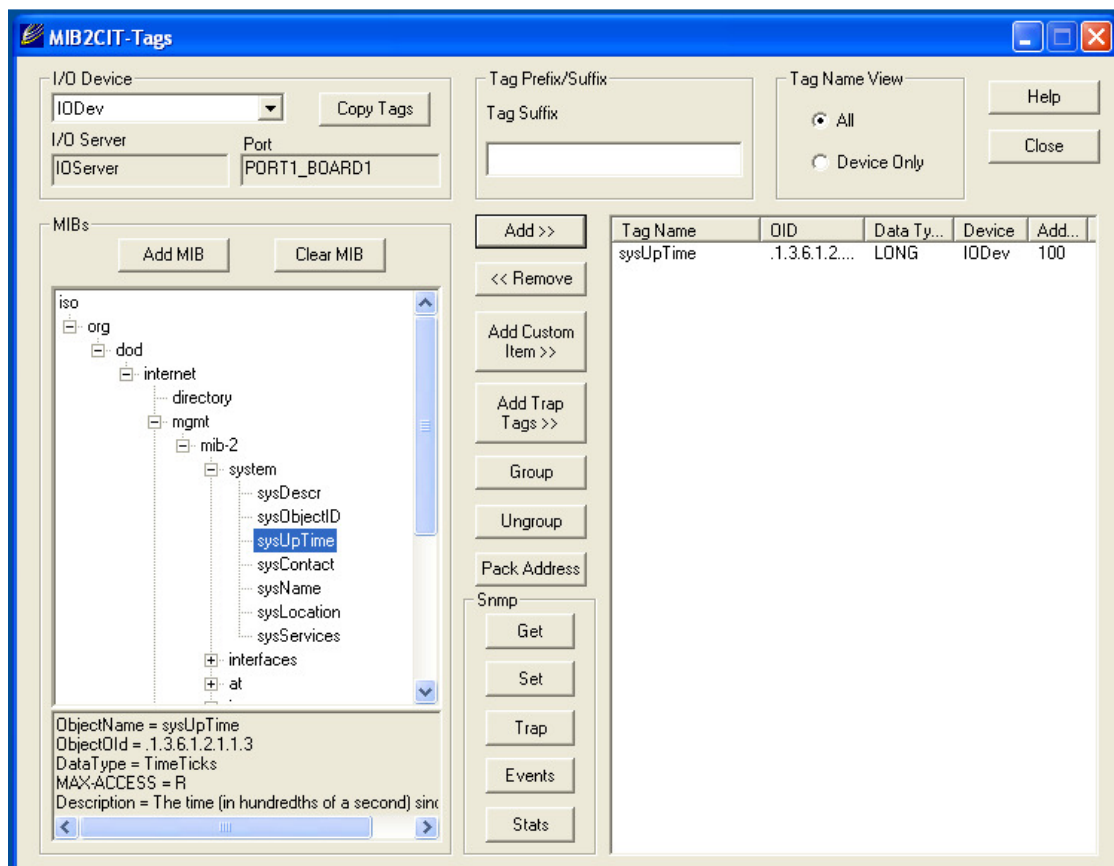
- Click on **Config Tags** on the main screen and select your project's IODevice

Figure 3: MIB2CIT Landing Page



- Navigate the MIB tree in the left hand pane to:
 - ISO > ORG > DOD > INTERNET > MGMT > MIB-2 > SYSTEM > SysUptime
 - Click **GET**. A new popup will appear with the result of your **GET** request.

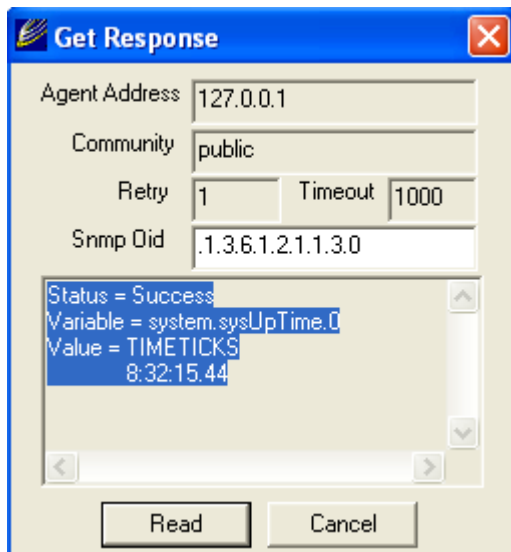
Figure 4: Showing Tag Configuration Page



How cool is that? 😊

You can see the value of the OID as well; in this case the OID value for SysUptime is 1.3.6.1.2.1.1.3. We just configured a simple SNMP GET request using MIB2CIT!

Figure 5: Showing the response to a GET query sent from the CitectSCADA IO Server



This popup shows us that SNMP is working correctly; we can see our machine has been up for 8hours 32 minutes.

In the popup, you will also see the SNMP OID. If you were to open the SNMPVARS.DBF, you would see that MIB2CIT has populated the OID fields and appended the suffix to the **TagName** field as shown in the table below.

This is why we needed to configure our CitectSCADA project first, before setting up MIB2CIT.

Figure 6: Showing the contents of SNMPVARS.DBF

INDEX	DEVNAME		TAGNAME	CTYPE	SNMPNAME	OID
100	IODev		sysUpTime_TestPC1	LONG	sysUpTime	.1.3.6.1.2.1.1.3
100	IODev		sysName_TestPC1	STRING	sysName	.1.3.6.1.2.1.1.5

1.4 Creating CitectSCADA variable tags with MIB2CIT

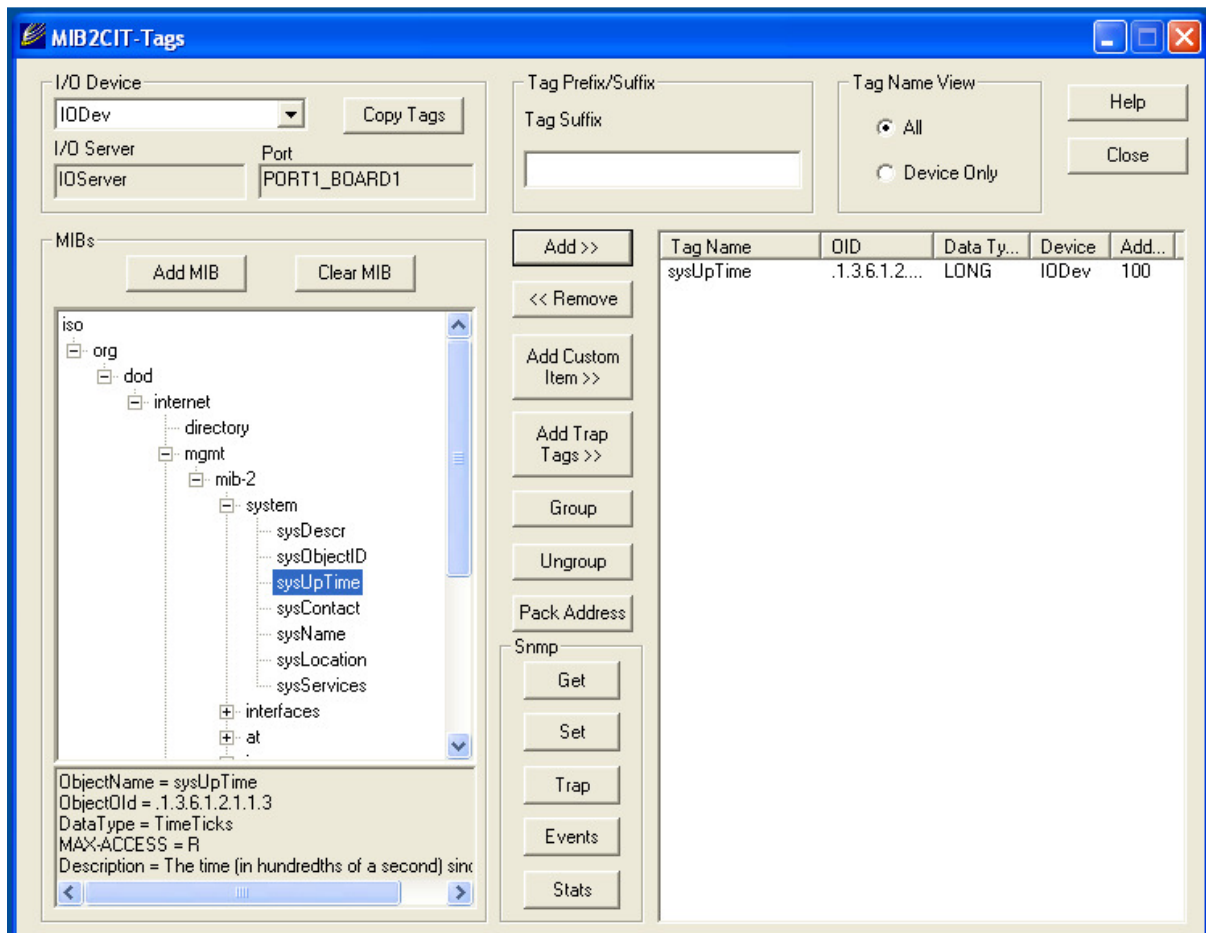
What is the relationship between CitectSCADA variable tags, and SNMP OIDs in the SNMP device we wish to monitor?

Each **IO point** being monitored in an RTU needs to have a corresponding variable tag set up in the CitectSCADA project in order for CitectSCADA to know what to monitor.

This is where MIB2CIT comes in really handy. Up until now MIB2CIT hasn't done much except browse our MIB Tree and fire off a **GET** command.

MIB2CIT will now create our CitectSCADA variable tags for us and cross-reference them to the device OIDs we wish to monitor in the device.

Figure 7: Tag configuration screen



1.4.1 Assigning tag suffixes to our OIDs and adding tags to our CitectSCADA Project

In the **Tag Suffix** field of the Tags screen, give your device a name such as `_TestPC1`. We need a suffix to determine which device the OID belongs to.

Think of a suffix like an IO Device name, a way of identifying a network device of which there are many of the same type. You could have 3 Cisco 2900 series switches to monitor so the MIB (and associated OIDs) would be the same on each switch. How would CitectSCADA know which switch the OID you want to poll belongs too? Well, we need to give each switch a unique Suffix.

Navigate down the MIB tree to SysUptime:

- ISO > ORG > DOD > INTERNET > MGMT > MIB-2 > SYSTEM > SysUptime

Click **Add>>**

Do the same for:

- ISO > ORG > DOD > INTERNET > MGMT > MIB-2 > SYSTEM > SysName

Click **Add>>**

Click **Close** when finished adding your tags.

Your Tag information should now export to the CitectSCADA DBF files, magic!

1.4.2 Verifying your tags have been exported to CitectSCADA using SNMPVARS.DBF and VARIABLE.DBF

CitectSCADA uses two files in the CitectSCADA project to store SNMP and traditional CitectSCADA Tag information, these files are:

- SNMPVARS.DBF
- VARIABLE.DBF

Both files are located in your project directory. You can open up the files using the DBF addin for Excel. You can see your CitectSCADA tag names, corresponding OID number as well as some other information shown below:

Figure 8: Taken from the SNMPVARS.DBF file

INDEX	DEVNAME	TAGNAME	CTYPE	SNMPNAME	OID
100	IODev	sysUpTime_TestPC1	LONG	sysUpTime	.1.3.6.1.2.1.1.3

Figure 9: Taken from the VARIABLE.DBF file showing the N100³ ADDR field

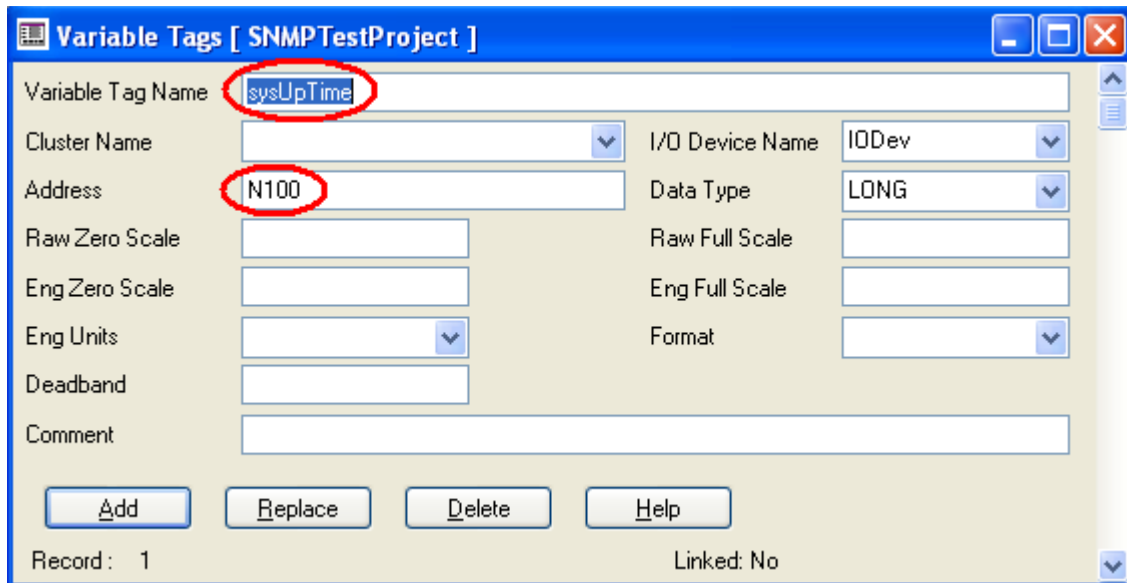
NAME	TYPE	UNIT	ADDR
sysUpTime_TestPC1	LONG	IODev	N100

³ Note: N100 in the ADDR field of VARIABLE.DBF stands for Numeric and DA is Alpha Numeric- see SNMPII Data Types in the online help for further explanation of these annotations.

In your CitectSCADA project, you can see the assigned address for that tag in your CitectSCADA Project Builder:

- Project Builder > Tags > Variable Tags

Figure 10: CitectSCADA Variable Tags



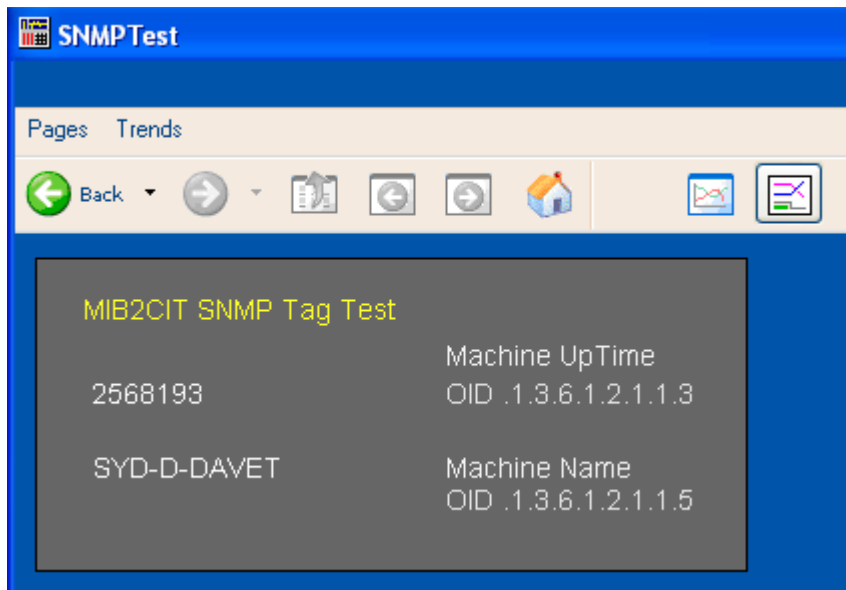
1.4.3 Testing your tags in CitectSCADA

Now, we know that MIB2CIT can read our SysUptime OID and we have verified that the Tags have been created in SNMPVARS.DBF and the addressing for the Tags are in VARIABLES.DBF so we should be able to read these values in CitectSCADA.

Create a new page in the Graphics Builder within your SNMPPProject. Place a text object on to your page and add the Tag "SysUptime" in the Display Value Tab. Do the same for SysName.

Save your page, compile and run your project.

Figure 11: CitectSCADA RUNTIME showing our SysUptime and SysName we configured with MIB2CIT



You should see something similar to the above, your Machine Name and Machine Uptime should be displayed.

This information has been retrieved using SNMP.

Additional Information

2 Useful Terminology

- **MIB: Message Information Block.**
 - A MIB is a text file written in ASN.1 (adapted subset), which defines the data objects available from an SNMP "manageable entity". MIB2CIT and GetIF use the MIB to navigate to an OID
- **OID: Object Identifier**
 - A numeric string that is used to uniquely identify an object within a MIB.
- **Community String**
 - A clear text password (in SNMPV1 & 2) sent by the SNMP manager in order to Read (**Get**) or Write (**Set**) values in an **OID**.