



Cloud Connector for embedded
Evaluation using Cygwin

1 Document History

Date	Version	Change Description	
12/19/2013	0.1	First draft	
01/15/2014	0.2	FAE and support review	

2 Table of Contents

1	Document History.....	2
2	Table of Contents.....	3
3	Introduction.....	4
3.1	Problem Solved.....	4
3.2	Audience.....	4
3.3	Assumptions.....	4
3.4	Scope.....	4
4	Basics.....	4
4.1	Download Cygwin installer.....	5
4.2	Install Cygwin.....	5
4.3	Select Packages.....	6
4.4	Cygwin terminal.....	7
4.5	Follow now the Cloud Connector Getting Started Guide.....	8
4.5.1	Building a connector sample.....	8
4.5.2	Executing the sample.....	9
5	Conclusion.....	10

3 Introduction

This document describes how to evaluate Cloud Connector on a Windows host machine using Cygwin environment.

3.1 Problem Solved

Etherios Cloud Connector porting kit provides a ready to use ‘Linux platform implementation’ that allows a user to quickly evaluate the product on a Linux host machine before porting to the targeted operating system and platform.

For Windows users not having a Linux machine available, we suggest using a virtual machine emulating Linux... and if that is not possible we describe here how to build and execute provided samples using Cygwin environment.

3.2 Audience

Users that want to do a kit first evaluation by running the samples provided for the host machine but unfortunately don’t have a Linux environment.

3.3 Assumptions

User has already downloaded Cloud Connector for embedded in order to port it to its own hardware or operating system.

User doesn’t have a native Linux environment or a Linux virtual machine.

3.4 Scope

After the Cygwin installation and configuration the user has to move to the ‘Cloud Connector getting started guide’ to start using the Cloud Connector.

4 Basics

Cygwin is:

- A collection of tools which provide a Linux look and feel environment for Windows.
- A DLL (cygwin1.dll) which acts as a Linux API layer providing substantial Linux API functionality.

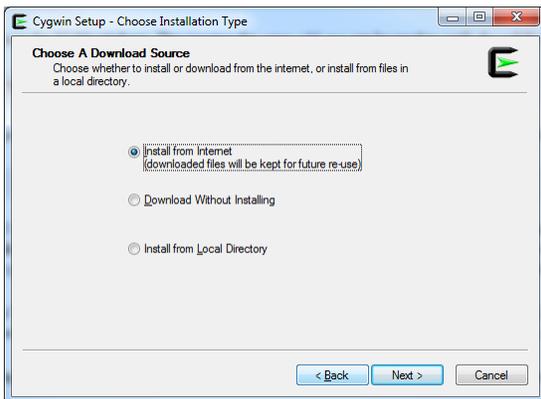
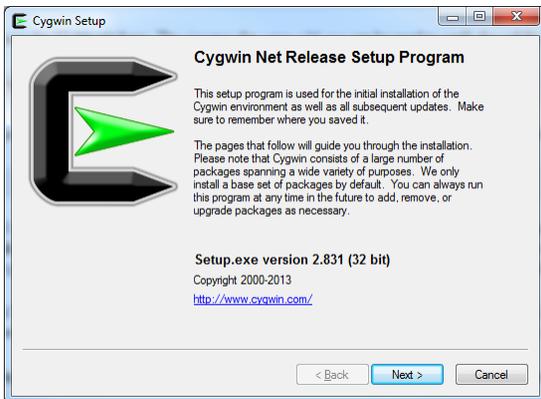
Following entries describe how to install Cygwin on your computer and configure it to include required packets to build the connector.

4.1 Download Cygwin installer

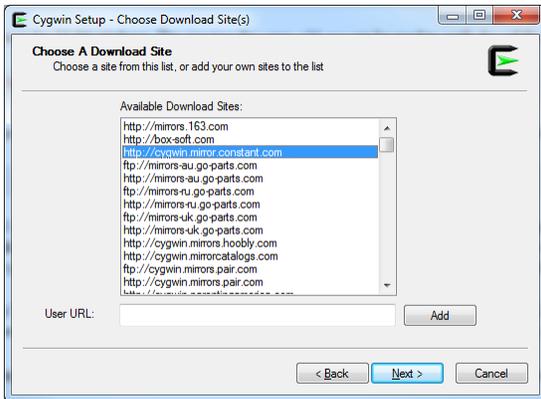
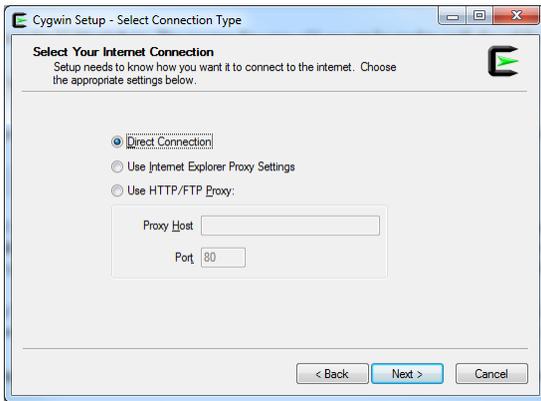
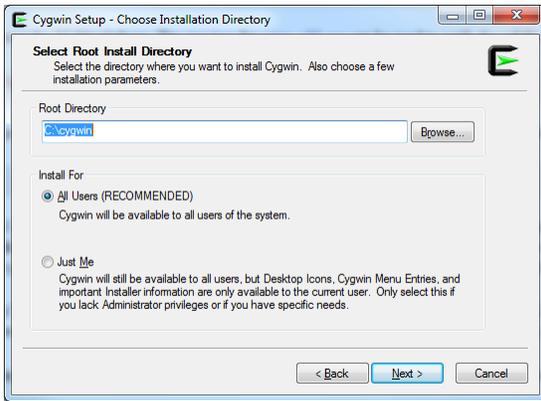
Go to www.cygwin.com , click on ‘Install Cygwin’ and select either ‘Run setup-x86.exe’, or ‘Run setup-x86_64.exe’ depending on your windows version.

4.2 Install Cygwin

Execute and run the installer choosing default options until you get the ‘Select Packages’ screen:



Cloud Connector for embedded evaluation using Cygwin



4.3 Select Packages

'Select Packages' screen has default packages selected. To build the connector you need to add some packages:

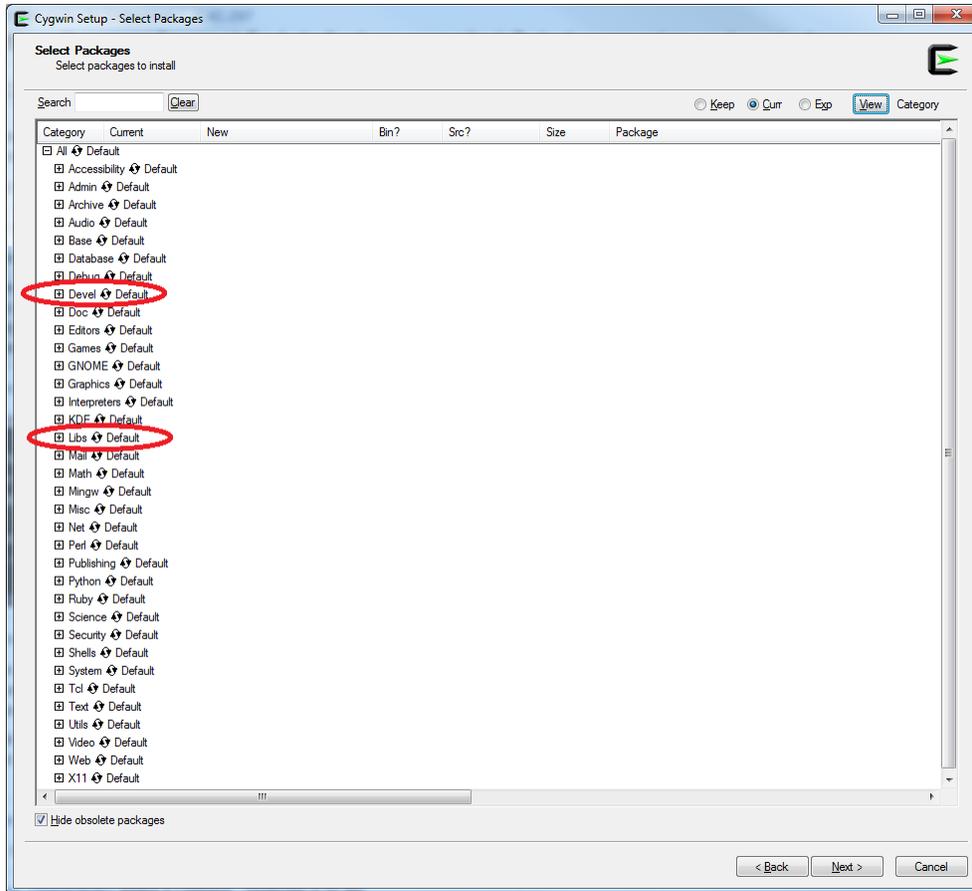
Expand the 'Devel' category, locate following components and click on the 'skip' icon so it changes to a version number:

- make: The GNU version of the 'make' utility
- gcc-core: GNU Compiler Collection (C, OpenMP)

Cloud Connector for embedded evaluation using Cygwin

- openssl-devel: The OpenSSL development environment
Expand the 'Libs' category, locate following components and click on the 'skip' icon so it changes to a version number:

- zlib-devel: Zlib de/compression library (development)

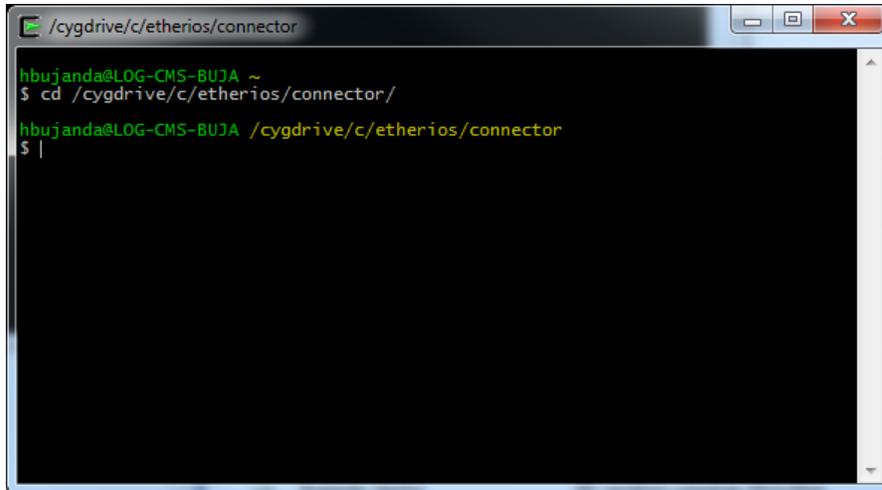


4.4 Cygwin terminal

Once installation finishes, open Cygwin terminal.

Then change to folder where the connector was uncompressed.

Note that Cygwin requires an starting '/cygdrive' before your path... so if you installed the connector at 'c:\etherios\connector', you have to type 'cd /cygdrive/c/etherios/connector'



```
/cygdrive/c/etherios/connector
hbujuana@LOG-CMS-BUJA ~
$ cd /cygdrive/c/etherios/connector/
hbujuana@LOG-CMS-BUJA /cygdrive/c/etherios/connector
$ |
```

4.5 Follow now the Cloud Connector Getting Started Guide

A Linux environment is ready now.

You should now continue with the ‘Getting Started’ chapter of the cloud connector documentation to go on learning on cloud connector.

Next steps are briefly described here:

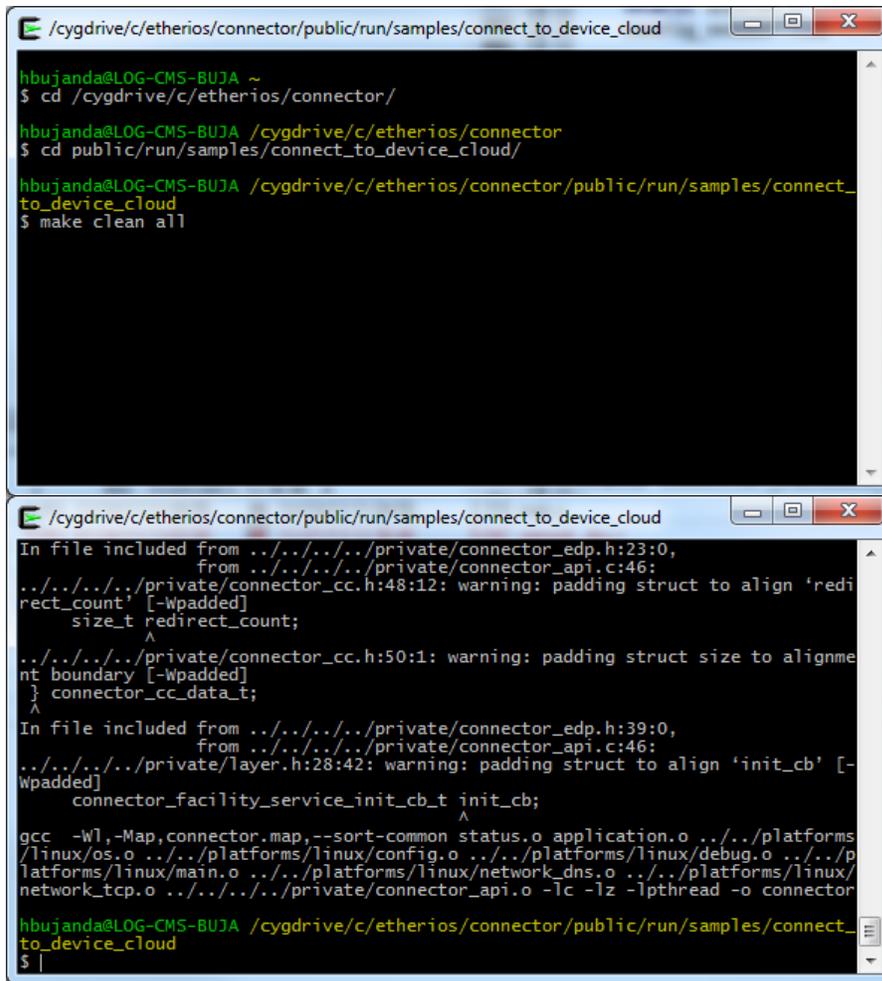
4.5.1 Building a connector sample

Read ‘Setup your Device Cloud Account’ subchapter of the ‘Getting Started’ chapter of the Cloud Connector documentation to learn on how to create a Cloud Connector account and get a Device_ID before continuing.

Read ‘Configuration Routines’ subchapter of the ‘Getting Started’ chapter of the Cloud connector documentation to learn on how to customize public/run/platform/linux/config.c file with your device mac address and your Device Cloud account Device_ID before continuing.

Move to the desired sample ‘public/run/samples/connect_to_device_cloud’ for example and build it using ‘make clean all’ command.

Cloud Connector for embedded evaluation using Cygwin



The image shows two screenshots of a Cygwin terminal window. The top screenshot shows the user navigating to the directory `/cygdrive/c/etherios/connector/public/run/samples/connect_to_device_cloud` and running `make clean all`. The bottom screenshot shows the compilation process, including warnings about padding structs and the final `gcc` command used to build the `connector` executable.

```
hbujuana@LOG-CMS-BUJA ~
$ cd /cygdrive/c/etherios/connector/
hbujuana@LOG-CMS-BUJA /cygdrive/c/etherios/connector
$ cd public/run/samples/connect_to_device_cloud/
hbujuana@LOG-CMS-BUJA /cygdrive/c/etherios/connector/public/run/samples/connect_
to_device_cloud
$ make clean all

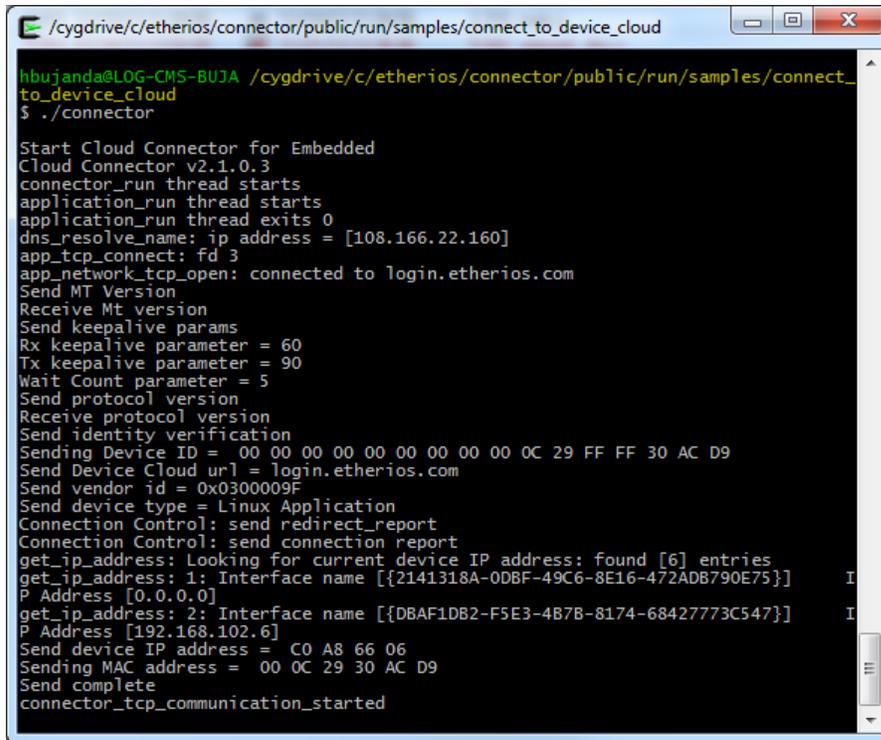
In file included from ../../../../private/connector_edp.h:23:0,
                 from ../../../../private/connector_api.c:46:
../../../../private/connector_cc.h:48:12: warning: padding struct to align 'redi
rect_count' [-Wpadding]
    size_t redirect_count;
    ^
../../../../private/connector_cc.h:50:1: warning: padding struct size to alignme
nt boundary [-Wpadding]
} connector_cc_data_t;
^
In file included from ../../../../private/connector_edp.h:39:0,
                 from ../../../../private/connector_api.c:46:
../../../../private/layer.h:28:42: warning: padding struct to align 'init_cb' [-
Wpadding]
    connector_facility_service_init_cb_t init_cb;
    ^
gcc -Wl,-Map,connector.map,--sort-common status.o application.o ../../platforms
/linux/os.o ../../platforms/linux/config.o ../../platforms/linux/debug.o ../../p
latforms/linux/main.o ../../platforms/linux/network_dns.o ../../platforms/linux/
network_tcp.o ../../../../private/connector_api.o -lc -lz -lpthread -o connector
hbujuana@LOG-CMS-BUJA /cygdrive/c/etherios/connector/public/run/samples/connect_
to_device_cloud
$
```

4.5.2 Executing the sample

Read ‘Add your Device to Device Cloud’ subchapter of the ‘Getting Started’ chapter of the Cloud Connector documentation to learn on how to add your device to the device Cloud account before continuing.

The build process may generate either a ‘connector.exe’ or a ‘connector’ file depending on your environment. You can execute them indifferently by typing ‘./connector’

Cloud Connector for embedded evaluation using Cygwin



```
hbujaanda@LOG-CMS-BUJA /cygdrive/c/etherios/connector/public/run/samples/connect_to_device_cloud
$ ./connector

Start Cloud Connector for Embedded
Cloud Connector v2.1.0.3
connector_run thread starts
application_run thread starts
application_run thread exits 0
dns_resolve_name: ip address = [108.166.22.160]
app_tcp_connect: fd 3
app_network_tcp_open: connected to login.etherios.com
Send MT Version
Receive Mt version
Send keepalive params
Rx keepalive parameter = 60
Tx keepalive parameter = 90
Wait Count parameter = 5
Send protocol version
Receive protocol version
Send identity verification
Sending Device ID = 00 00 00 00 00 00 00 00 0C 29 FF FF 30 AC D9
Send Device Cloud url = login.etherios.com
Send vendor id = 0x0300009F
Send device type = Linux Application
Connection Control: send redirect_report
Connection Control: send connection report
get_ip_address: Looking for current device IP address: found [6] entries
get_ip_address: 1: Interface name [{2141318A-0DBF-49C6-8E16-472ADB790E75}] I
P Address [0.0.0.0]
get_ip_address: 2: Interface name [{DBAF1DB2-F5E3-4B7B-8174-68427773C547}] I
P Address [192.168.102.6]
Send device IP address = C0 A8 66 06
Sending MAC address = 00 0C 29 30 AC D9
Send complete
connector_tcp_communication_started
```

Read ‘Viewing Results on Device Cloud’ subchapter of the ‘Getting Started’ chapter of the Cloud Connector documentation to learn on how to check that your device is connected now to the Device Cloud

5 Conclusion

Cygwin provides a free and easy to install environment to evaluate Cloud Connector for Windows users.