Installation / Quick Start Guide	
Product	DASH 1.1 SDK Windows and Linux
Release Number	
Release Date	October 2010

Introduction

DASH SDK provides software in source and binary form to allow management consoles and management applications to support the Distributed Management Task Force DASH interface.

The SDK consists of two top-level components:

- Low-level APIs that expose all DASH-related CIM (Common Information Model) properties and methods. This allows full access to all DASH capabilities.
- High-level APIs for common operations such as discovery, power-on/ff, and basic inventory. The high-level API is an abstraction layer and minimizes the exposure of the console to details of the low-level DASH protocol stack.

This document describes how build and get started with DASH SDK for Windows and Linux Operating Systems.

Source Build Process

Requirements

Windows Environment

What other software must be installed first?

- Windows XP/Vista/2003 (x86 PC)
- Get Microsoft Visual C++ 2008 express edition.
  - Download the installer from the link below:
    http://www.microsoft.com/express/download/
  - Follow the installer's instructions.
- Get Perl
  - Download and Install ActivePerl from the below link:
    http://www.activestate.com/Products/activeperl/index.mhtml
  - Other versions of perl might work as well, but are not officially supported.
  - During the setup, ActivePerl asks whether it should be added to the PATH environment variable; activate this checkbox
  - If you didn't check this, add the directory manually to the PATH environment:
    Control Panel -> System -> Advanced -> Enviornment Variable -> click on "PATH" -> Edit
    Add the path of the bin directory of the perl installation to end of existing path.
- Get Putty
  - Download putty from below link. (Only needed for text redirection).
    http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.
  - Set the path to putty.exe as below.
    Control Panel -> System -> Advanced -> Enviornment Variable -> click on "PATH" -> Edit
    Add the path (directory where the putty.exe is located) to end of existing path.
- Get NSIS, the NullSoft Installer
  - Download NSIS from the link below. (Only needed to build the installer)
    http://nsis.sourceforge.net/Main_Page
  - Install the NSIS package
  - Add NSIS install directory to the PATH environment variable (see in the Putty section above)
  - copy inetc.dll to nsis/plugin folder: http://nsis.sourceforge.net/Inetc_plug-in

- extract EnvVarUpdate plugin to nsis include folder: http://nsis.sourceforge.net/Environmental_Variables:_append%2C_prepend%2C_and_remove_entries#Function_Code

Linux Environment

What other software must be installed first?

- x86 PC with any distribution of Linux installed. (Currently only Suse Enterprise Edition 11.1/Redhat/Fedora is verified).

- Install development package.(gcc,g++ and other development tools like autoconf, automake etc).

- Install libxml2, libcurl, openssl, python development packeges.

  Development Packages required for Suse Enterprise Edition 11.1
  ---
  1. Python-devel
  2. libopen-ssl-devel
  3. zlib-devel
  4. readline-devel
  5. ncurses-devel
  6. tack
  7. libxml2-devel

- Install ssh client and telnet client (Only needed for text redirection).

Building Source under Windows

Please follow the steps outlined below:

1	Download the latest release of DASH SDK.
2	Open the solution. Start the VC++ IDE and open the solution file: File -> Open -> Project/Solution, select: *windows/dash/dash.sln* The DASH SDK includes the following projects: ➢ **dashcli** (Application) ➢ dashapi (C++ library) ➢ dashapic (C library) ➢ wsman_client_lib (library) ➢ Openwsman (library) ➢ eventsink (library) ➢ openssl (library) ➢ libxml2 ➢ mof2oal (tool to generate C++ files from mof file) ➢ installer (the Windows installer) *Note - These projects can also be built individually using the project specific solution file.*
3	Build the solution - Click *Build-> Batch Build*, select all projects (Select All button) and press Build. That builds all projects (application and library) and copies the executable to "*windows/win32pkg*" directory. This includes all application; library and dependent dll's concerning release build as well as debug build. - Please note that you don't select "installer" if you don't want to build the Windows Installer. - Please note that the installer requires that all other projects are built in both "Debug" and "Release" configuration - The installer is copied into the "*installer*" directory *Note – The above process builds both release and debug builds*

Debugging under Windows

Please follow the steps outlined below:

1	Open the *windows/dash/dash.sln* solution file in *VC++ IDE* (like described above)
2	Build the solution if necessary (like described above).
3	Start debugging Select a specific project, and Right Click -> Set as startup project and press debugging keys *F10, F11, or F5* to start

Building Source under Linux

Please follow the steps outlined below:

1	Make sure to download the latest released DASH SDK from *sourceforge*
2	From the command prompt go to the root directory where the DASH SDK is stored
3	Type the following command • *#./configure --prefix=<root DASH SDK directory>/install_root* *Ex:- if DASH SDK root directory path is : "/home/testuser/DASHSDK" then command looks like –* *#./configure --prefix= /home/testuser/DASHSDK/install_root* • *# make* • *#make install* The following components will be built: ➢ **dashcli** (Application) ➢ mof2oal (tool to generate C++ files from mof file) ➢ dashapi (C++ library) ➢ dashapic (C library) ➢ wsman_client_lib (library) ➢ Openwsman (library) ➢ wseventsink (library) ➢ webserver

Getting Started

How can I quickly get started using *DASH 1.0 SDK*?

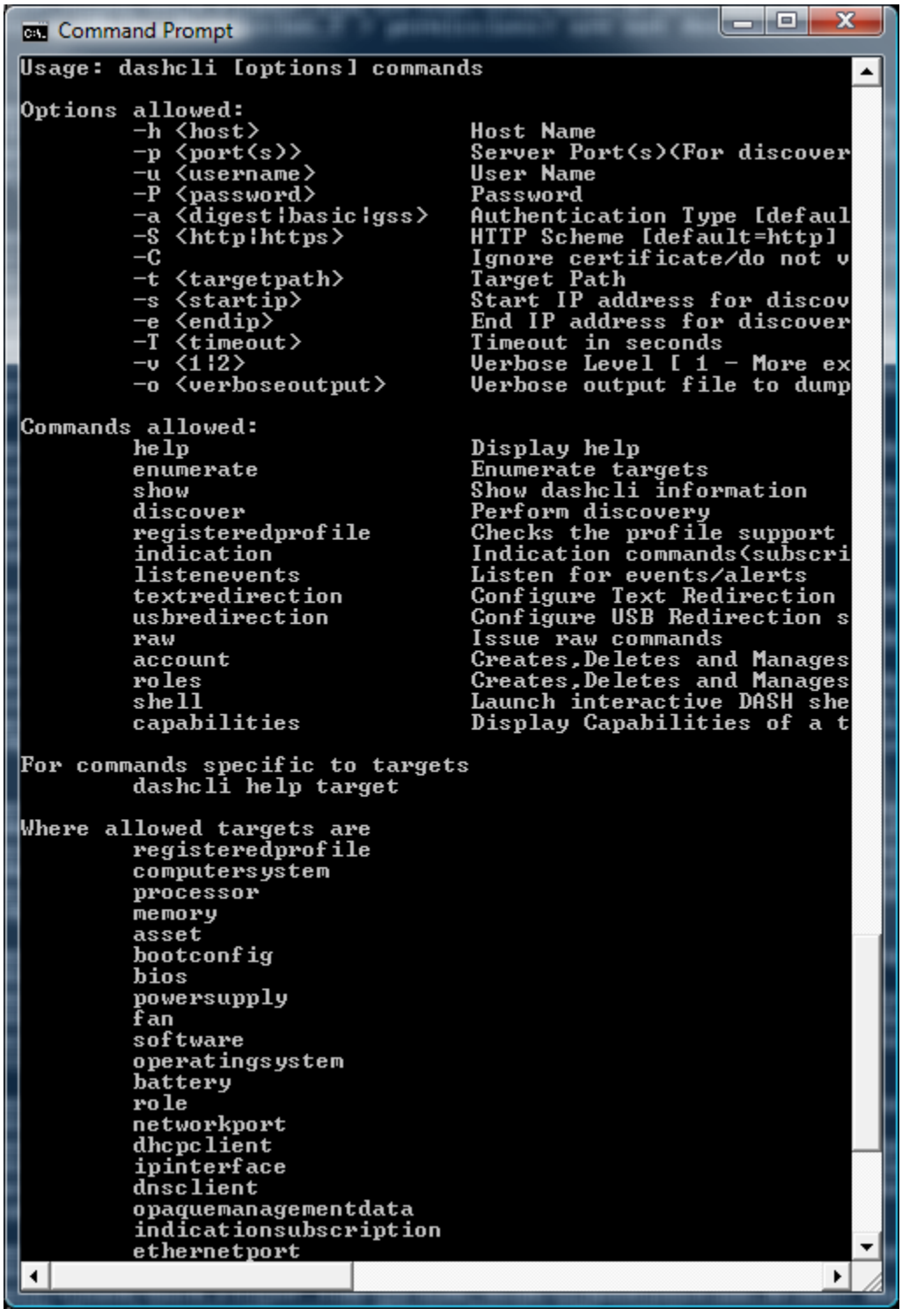
WINDOWS:
Please follow the steps outlined below for getting started in windows:

1	The quickest way to test the SDK library is to use the DASH CLI *Note – DASHCLI uses the SDK library to communicate with a DASH enabled system.*
2	After the source is built successfully, go to the release directory where the Window package is stored *example - \trunk\windows\win32pkg\release>*
3	From the above command prompt, run the DASHCLI to start using and testing SDK library.
4	To write an application using the DASH SDK, see the DASH 1.0 SDK API reference document

LINUX:
Please follow the steps outlined below for getting started in Linux:

1	The quickest way to test the SDK library is to use the DASH CLI *Note – DASHCLI uses the SDK library to communicate with a DASH enabled system.*
2	After the source is built successfully, r*un the DASHCLI to start using and testing SDK library.* To start dashcli type from the command prompt *#dashcli*
3	To write an application using the DASH SDK, see the DASH 1.0 SDK API reference document

Command prompt sample screen is shown below –


Command Prompt
Usage: dashcli [options] commands
Options allowed:
-h <host> Host Name
-p <port(s)> Server Port(s)(For discover
-u <username> User Name
-P <password> Password
-a <digest|basic|gss> Authentication Type [defaul
-S <http|https> HTTP Scheme [default=http]
-C Ignore certificate/do not v
-t <targetpath> Target Path
-s <startip> Start IP address for discov
-e <endip> End IP address for discover
-T <timeout> Timeout in seconds
-v <1|2> Verbose Level [1 - More ex
-o <verboseoutput> Verbose output file to dump
Commands allowed:
help Display help
enumerate Enumerate targets
show Show dashcli information
discover Perform discovery
registeredprofile Checks the profile support
indication Indication commands(subscri
listenevents Listen for events/alerts
textredirection Configure Text Redirection
usbredirection Configure USB Redirection s
raw Issue raw commands
account Creates,Deletes and Manages
roles Creates,Deletes and Manages
shell Launch interactive DASH she
capabilities Display Capabilities of a t
For commands specific to targets
dashcli help target
Where allowed targets are
registeredprofile
computersystem
processor
memory
asset
bootconfig
bios
powersupply
fan
software
operatingsystem
battery
role
networkport
dhcpclient
ipinterface
dnsclient
opaquemanagementdata
indicationsubscription
ethernetport