Honeywell

Scanner Management Utility

User Guide

Disclaimer

Honeywell International Inc. ("HII") reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult HII to determine whether any such changes have been made. The information in this publication does not represent a commitment on the part of HII.

HII shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of HII.

Microsoft®, Windows®, Visual C++® are registered trademarks of Microsoft corporation.

©2021 Honeywell International Inc. All rights reserved.

Other product names or marks mentioned in this document may be trademarks or registered trademarks of other companies and are the property of their respective owners.

Web Address: www.honeywellaidc.com

Patents

For patent information, please refer to www.honeywellaidc.com\patents

TABLE OF CONTENTS

Customer Support

If you need assistance installing or troubleshooting your product, contact us by using one of these methods:

Technical Assistance

To search our knowledge base for a solution or to log in to the Technical Support portal and report a problem, go to https://support.honeywellaidc.com.

For our latest contact information, see www.honeywellaidc.com/locations.

Web Form

You can contact our technical support team directly by filling out our online support form at https://support.honeywellaidc.com. Enter your contact details and the description of the question or problem.

SCANNER MANAGEMENT UTILITY

Overview

The Honeywell command line Scanner Management Utility (SMU) assists with the staging, deployment, and management of barcode scanners.

The key features of Scanner Management Utility are:

- Firmware updates
- Configuration updates
- Golden image and sending end device model specific commands
- Details of available features and work flows

System Requirements

Before you proceed, you should check that your computer meets the minimum system requirements to install and run the Scanner Management Utility.

- Operating System: 32 or 64 bit version of Microsoft® Windows® 10, 8, 7, POSReady 7, or POSReady 2009.
- Installed Microsoft Visual C++® 2010 Redistributable Package.

Note: Honeywell recommends keeping your operating system version updated to the latest release.

Software Installation

To download the Scanner Management Utility

1. Go to www.honeywellaidc.com.
2. Select **Resources** > **Downloads** > **Software**.
3. Click on the Technical Support Downloads Portal link https://hsmftp.honeywell.com/.
4. Create an account if you have not already created one. You must log-in to download the software.

5. Navigate to **Software** > **Barcode Scanners** > **Software** > **Tools and Utilities**.

6. Download the zip file to your PC and extract the files. The zip file contains the SMU.exe file, NULL Driver, Release Notes, User Guide, and Flash DLLs.

7. Click on **SMU.exe** file to run the application and follow the commands.

8. During the installation process, run the **nulldrvinstall.bat** batch file with administrative privileges to install the NULL driver on Windows 8.1 and Windows 10 to associate Honeywell scanner devices configured as USB HID barcode or USB Keyboard scanner.

Note: NULL Driver is available inside SMU package under NULL_DRIVER folder.

Supported Scanner Models

For supported scanner models, please refer to SMU release notes.

Start Scanner Management Utility

Scanner Management Utility (SMU) is a command line tool.

To execute the Scanner Management Utility:

1. Open the command prompt from the Windows Start menu.
2. Navigate to the folder containing the SMU.exe file.
3. Execute the command using the syntax below.

The syntax is:

smu.exe {[/exmset <file>] | [/exmget <file>] | [/upg <file>] | [/cmd <file>] | [/cmdy <file>]} [/timeout <0...>] [/port <0,1,2...>] [/model <string>] [/loglevel <0-4>]

Mandatory Arguments

When creating a work flow, one of the five Mandatory Arguments shown below must be specified.

Mandatory Arguments	
Syntax	**Description**
/exmset	Downloads the specified EXM file to the scanner.
/exmget	Retrieves the scanner configuration to the given.
/upg	Upgrades the scanner firmware with the given MOC.
/cmd	Sends the given string as a SYN-M menu command. Scanner response is dumped to standard output.
/cmdy	Sends the given string as a SYN-Y menu command. Scanner response is dumped to standard output.

Optional Arguments

Optional Arguments	
Syntax	**Description**
/timeout	Maximum time to wait for a scanner reply, in milliseconds. Default=5000ms.
/port	Indicates where to look for the scanner. 0=Parallel Auto Detection (default) for the following interfaces: • USB ReM Interface • USB ReM legacy Interface • USB HID Barcode Scanner • USB Serial, RS232 and BT SPP ports (in this order, for each COM present where at least one modem signal is active). 1,2,3… = Any other fixed COM port. For example USB Serial, RS232 or BT SPP.
/interface	Indicates which interface to look for when multiple devices are found. SMU runs the command on the fist device found. If **/port is 0 or no /port specified**, then SMU searches for the /interface value. • USB HID POS • USB Serial • USB ReM • ReM legacy (The second interface for USB HID Key board device). • RS232 • BT SPP If the **/port value is 1,2,3 . . .,** then SMU ignores the /interface defined.
/model	Device's name (SYN-Y P_NAME) must match the provided string.
/loglevel	Logging Level [0-4] Default=0. "*Create Log*" on page 7 for detailed information on log level values 0 to 4 and their descriptions.

Return Codes

Upon execution, the SMU returns exit codes according to the specified parameters.

If multiple arguments are specified, the SMU returns ACK only if all the arguments return ACK. Otherwise, the SMU sets the last NAK or ENQ as the return code.

/cmd and /cmdy Exit Codes		
Value	Name	Description
0	ACK	Success: Scanner replied with ACK
1	ENQ	Error: Menu command not recognized
2	NAK	Error: Menu command recognized but rejected
3	TIMEOUT	Error: Timeout awaiting a device response

/exmset, /exmget, and /upg Exit Codes		
Value	Name	Description
0	Success	Success: The operation (exmset, exmget or upg) is successful
4	No_Device	Error: Did not find a scanner
5	Timeout_Invalid	Error: Timeout value is invalid
6	ParseCmdLine_ Error	Error: Command line arguments are invalid
[7,13]	SDK_DC_INTER_ ERROR	Error: SDK discovery internal error
[14,19]	SDK_DC_INTER_ ERROR	Error: SDK data connection internal error
[20,23]	SDK_ CONNECTION_ INTER_ERROR	Error: SDK data internal connection error
[24,41]	SDK_UPG_ INTER_ERROR	Error: SDK flash firmware internal error
[42,2141]	DSM_INTER_ ERROR	Error: DSM internal error

Supported Workflows and Examples

Create a configuration "Golden Image"

The EXM configuration file is termed as the "Golden Image".

To create a configuration:

1. Define configuration as desired with EZConfig.
2. Use the SMU to retrieve golden.exm.

Example: *smu.exe /exmget golden.exm*

Update a configuration with "Golden Image"

The SMU can update the configuration on single device or multiple devices, when used with PowerShell or other scripting language with logging scripts. Push the golden.exm settings to the first autodetected scanner or the scanner connected to COM2.

Update to auto detect scanner

Example: *smu.exe /exmset golden.exm*

Update to specific port

Example: *smu.exe /exmset golden.exm /port 2*

Update Firmware

SMU can update a single device or multiple devices on the same PC, when used with PowerShell or other scripting language with looping scripts.

Upgrade specific scanner:

Upgrade on a specific PC example below uses xxxxxxx on port 21.

Example: *smu.exe /upg BJ000180AAA.moc /port 21 /model 1900*

Upgrade several scanners on a PC:

Upgrade to auto detect scanner example below uses xxxxxxx on port 21.

Example: *smu.exe /upg BJ000180AAA.moc*

Create Log

Use SMU to create file smu.log with detailed logging.

Example: *smu.exe /exmget myconfig.exm /loglevel 4*

Total of five logging levels are supported. The description of each level is mentioned below.

Log Level	Description	Remarks
0	LogLevelNone	No information is logged.
1	LogLevelData	Data communication between the host and the device is logged in the file. Both hexadecimal vales and corresponding ASCII character values of each byte of transferred data is logged.
2	LogLevelError	Debug statements are logged during error conditions.
3	LogLevelInfo	Debug statements with generic information and communication data are logged during any successful operation.
4	LogLevelDebug	All debug statements from the underlying Scanning SDK library along with communication data are logged in the file. The file size generated is significantly larger than the other levels since it includes complete debugging information.

Batch Operations

The real power of the SMU is extracted from a server using PowerShell scripts or other scripting language with looping scripts. An administrator can upgrade many scanners in an organization at the same time and log the results of each update activity.

PowerShell scripts can be used to:

- Find all scanners in network
- Upgrade all scanners on a list
- Find all scanners online

PowerShell Script Example

```powershell
region: User Defined Functions

function RunCmd_AND_Log {

Param(


[Parameter(mandatory=$true]

[String]$cmdString,

[Parameter(mandatory=$true]

[String]$logFile,

[Parameter(mandatory=$false]

[String]$params,

[Parameter(mandatory=$false]

[String]$outputFile = ""

)

if($outputFile -ne ""){

    #Run Input Command

    Start-Process -filepath $cmdString -ArgumentList $params -NoNewWindow
  -RedirectStandardOutput $outputFile -Wait

}

else{

    Start-Process -filepath $cmdString -ArgumentList $params -NoNewWindow -Wait

}

#Depending on output, Log if cmd Succeeded or failed

switch($?)

{

    $true{Add-Content $logFile $($cmdString+ $params +" Ran.")}

    $false{Add-Content $logFile $($cmdstring +$params +" Failed to Run.")

}

#endregion: User Defined Functions

set-location c:\yourfolder\yoursubfolder

#region: Setup

#Create Log File
```

```
$EZDLogFile = New-Item -ItemType file -Path $(".\FullEZDInstall_$
(get-date -f yyyy-MM-dd_hhmmss).log")

#Test if posapp is running, if so quit install

$posapp = Get-Process -name "posapp" -ErrorAction SilentlyContinue

if($posapp){

    Add-Content $EZDLogFile "posapp is still running! Install Terminated."

    exit

}

#first test for smu.exe, without it we cannot proceed with anything

if((Test-Path -path .\smu.exe)-eq $false){

    Add-Content $EZDLogFile "Folder does not contain smu.exe! Install Terminated."

    exit

}

#Need to check if plugin is already installed

c:\yourfolder\yoursubfolder\smu.exe /cmd plgdir. | out-file .\plugindir.txt -force

$EasyDLPlugin = select-string -path .\plugindir.txt -pattern "EasyDL_2_0.plugin"

if($EasyDLPlugin){

add-content $EZDLogFile "EasyDL Plugin Already Installed! Install Terminated."

exit

}

$missingSerialFilePath = ".\MissingSerial.txt"


# Check to see if missing serial file exists if so delete

if((Test-Path $missingSerialFilePath)){

Remove-Item -Path $missingSerialFilePath

}

#endregion: Setup

#region: Check Firmware

do{

$reRun = $false

#get rev info

If (Test-Path C:\yourfolder\yoursubfolder\rev.txt) {

Remove-Item C:\yourfolder\yoursubfolder\rev.txt -force
```

```
}

C:\yourfolder\yoursubfolder\smu.exe /cmd revinf.
| out-file C:\yourfolder\yoursubfolder\rev.txt -Force

$DeviceNotFound = Select-String -path .\rev.txt -Pattern "No Device Found"

if($DeviceNotFound){

Add-Content $EZDLogFile "Scanner Not Found! Install Terminated!"

exit

}

$serial = (gc c:\yourfolder\yoursubfolder\rev.txt)[10]

$ser = $serial.Substring(15)

$firmware = (gc c:\yourfolder\yoursubfolder\rev.txt)[6]

$fw = $firmware.Substring(22)

#if firmware not upgraded, run upgrade and rerun code

If ($fw -ne " CB000107BAA" ) {


   Add-Content $EZDLogFile "Firmware needs to be upgraded.."


       if(test-path -path .\ CB000107BAA.moc)

       {

          c:\posapps\honeywell\smu.exe /upg CB000107BAA.moc
          | out-file C:\yourfolder\yoursubfolder\fwlog.txt -Force

          Add-Content $EZDLogFile "Firmware Upgraded."

          $reRun -eq $true

       }


       else

       {

          Add-Content $EZDLogFile "Firmware Upgrade File Not Found! Install Terminated."

          Exit

       }

   }

   else{Add-Content $EZDLogFile "Firmware Version CB000107BAA Confirmed"}


}until($reRun -eq $false)
```

```
#endregion: Firmware

#region : EZDL Install

#Check to make sure all required files exists

$EasyDlExists = Test-Path -Path ".\EasyDL.moc"

$ssLKExists = test-path -path ".\ss_lk.txt"

$EZDLConfigExists = test-path ".\EZDL_GS1Parse_Config.exm"

#Search for Serial in license file

$SerialFound = Select-String -Path c:\yourfolder\yoursubfolder\ss_lk.txt -Pattern
$ser -quiet

if($EasyDlExists -eq $false){

    Add-Content $EZDLogFile "EasyDL.moc Missing."

}

if($ssLKExists -eq $false){

    Add-Content $EZDLogFile "ss_lk.txt Missing."

}

if($EZDLConfigExists -eq $false){

    Add-Content $EZDLogFile "EZDL_GS1Parse_Config.exm Missing."

}

if($SerialFound -eq $false){


    Add-Content $EZDLogFile "Serial Not Found in ss_lk.txt."

    #add missing serial number to text file

    Add-Content $missingSerialFilePath $ser

}

$ExePath = "c:\yourfolder\yoursubfolder\smu.exe"

#if serial is found and all reqired files exist, proceed

If ($SerialFound -and ($EasyDlExists -and $ssLKExists -and $EZDLConfigExists)) {


    Add-Content $EZDLogFile "Starting EZDL Install..."

        #install ezdl

    RunCmd_AND_Log -cmdString $ExePath -params "/upg EasyDL.moc" -logFile $EZDLogFile

    #license ezdl

    RunCmd_AND_Log -cmdString $ExePath -params "/manifest ss_lk.txt" -logFile $EZDLogFile
```

```
#reset scanner

RunCmd_AND_Log -cmdString $ExePath -params "/cmd reset_." -logFile $EZDLogFile

#send config

RunCmd_AND_Log -cmdString $ExePath -params "/cmd 9902A0022[remainderofserialnumber]."
-logFile $EZDLogFile

# 9902A0022[remainderofserialnumber].

#send exm settings

RunCmd_AND_Log -cmdString $ExePath -params "/exmset EZDL_GS1Parse_Config.exm" -logFile
$EZDLogFile

#verify plugin

RunCmd_AND_Log -cmdString $ExePath -params "/cmd plginf." -logFile $EZDLogFile
-outputFile ".\license.txt"

Add-Content $EZDLogFile "EZDL Install Complete."

}

Else {

    #Add-Content $EZDLogFile "Plugin Not Done;Only exm applied."

    #send exm settings

    RunCmd_AND_Log -cmdString $ExePath -params "/exmset C:\yourfolder\yoursubfolder
    \EZDL_GS1Parse_Config.exm" -logFile $EZDLogFile

    #& C:\yourfolder\yoursubfolder\smu.exe /exmset C:\yourfolder\yoursubfolder\
    EZDL_GS1Parse_Config.exm

    }
#endregion : EZDL Install
```

Update Firmware and License/Install EZDL

The PowerShell script example shown checks:

- If the POS application is running
- Presence of the SMU
- Firmware version and updates if necessary
- License file for the serial number

The PowerShell script also:

- Installs EZDL, if necessary
- Applies any config updates

Honeywell
9680 Old Bailes Road
Fort Mill, SC 29707

www.honeywellaidc.com