

Trusted ePlatform Services



Software Protection

Version 1.0

User Manual



Advantech Co. Ltd.

No. 1, Alley 20, Lane 26,
Rueiguang Road, Neihu District,
Taipei 114, Taiwan, R. O. C.

www.advantech.com

Copyright Notice

This document is copyrighted, 2008, by Advantech Co., Ltd. All rights reserved. Advantech Co., Ltd. Reserves the right to make improvements to the products described in this manual at any time. Specifications are thus subject to change without notice.

No part of this manual may be reproduced, copied, translated, or transmitted in any form or by any means without prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd., assumes no responsibility for its use, or for any infringements upon the rights of third parties which may result from its use.

All the trade marks of products and companies mentioned in this data sheet belong to their respective owners.

Copyright © 1983-2009 Advantech Co., Ltd. All Rights Reserved

Version History

Date	Version	Author	Description
2009/05/1	1.0	CL/Wilson	New release

Table of Contents

Introduction.....	4
Security ID Structure	5
How to protect your Application?	6
Product Features	6
Environments	7
System Architecture	7
Support BIOS Type	7
Support OS.....	7
Tutorial of Using Software Protection	8
Step1. Install the Utility and Library	8
Step2. Security BIOS is necessary	8
Step3. Check the Security Status “Red”	8
Input the Vendor ID	9
Step4. Input the Customer ID.....	10
Step5. Double check the IDs.....	11
Step6. Check the Security Status “Green”	12
Step7. Write application	13
Software Protection Program.....	14
Installation.....	14
How to write the Security ID?	15
How to check the Security ID?	19
SUSI API Programmer’s Documentation	20
[Initialize Module:]	20
(1) bool EPF_InitializeOpen	20
(2) bool EPF_InitializeClose	20
[Secure Module:].....	21
(3) bool EPF_SetCustomerIDData.....	21
(4) bool EPF_SetSecureVendorIDData.....	21
(5) bool EPF_CheckSecureID	22
Appendix.....	23
Supported BIOS Description.....	23

Introduction



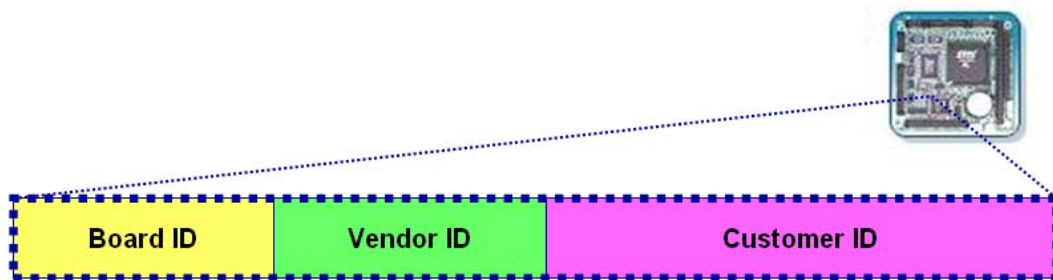
The embedded application is the most important property of a system integrator. It contains valuable intellectual property, design knowledge and innovation, but it is easy to be copied! An unscrupulous competitor only needs to purchase one system from the market to copy the embedded application and run it on a similar hardware platform—voila! The illegal copy is made.

Advantech Embedded Core Services developed a Software Protection utility which provides reliable security functions for customers to secure their application data within embedded BIOS. We've designed three unique Security ID functions inside the BIOS, consisting of Board ID, Vendor ID, and Customer ID. The Security ID is a unique string defined by the customer and encrypted by using hash function SHA-1. The system Integrators' embedded application can then access the Security ID by calling Advantech Application Programming Interface (API), and if the Security IDs are not correct, the application will stop executing and send a message over the LAN for further action depending on customers' application design—security is assured!

Security ID Structure

The Security ID consists of Board ID, Vendor ID, and Customer ID stored in a special area in the BIOS. The Board ID will store the first MAC address of the board or system, so each board or system will have a unique Board ID, which will be factory installed and read only. Vendor ID is a unique string for each customer or project; this is pre-defined by the customer or can be input at the factory. The last is Customer ID; a unique string defined by the customer and input by customer using our utility or the customer's own application calling our API. The three IDs provide a triple level protection for the customer's system.

The Security ID is encrypted by using hash function SHA-1. (See Note1)



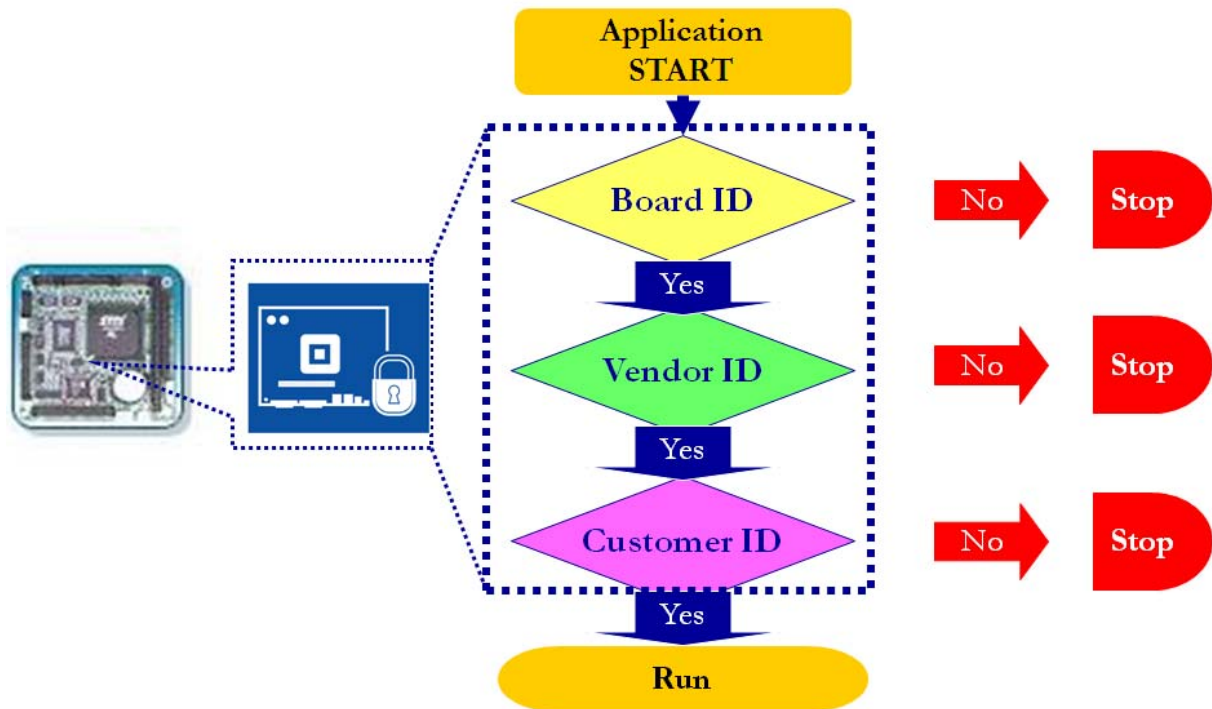
1. **Board ID**: unique string for each board, ready in Factory, Read Only
2. **Vendor ID**: unique string for each customer or project, input by customer
3. **Customer ID**: unique string defined by customer, input by customer

Note1: The SHA hash functions are a set of cryptographic hash functions designed by the National Security Agency (NSA) and published by the NIST as a U.S. Federal Information Processing Standard. SHA stands for Secure Hash Algorithm. The three SHA algorithms are structured differently and are distinguished as SHA-0, SHA-1, and SHA-2. The SHA-2 family uses an identical algorithm with a variable key size which is distinguished as SHA-224, SHA-256, SHA-384, and SHA-512.

SHA-1 is the best established of the existing SHA hash functions, and is employed in several widely used security applications and protocols.

How to protect your Application?

System Integrators' embedded application can access the Security ID by calling our API (Application Programming Interface). There are 3 IDs for checking, see below, if the Security IDs are not correct, it can stop to run and send a message back from LAN for further action depending on your application design.



Product Features

- Protection by Security ID via BIOS
- Security ID consist of Board ID, Vendor ID & Customer ID
- Security ID is stored in special area in BIOS
- Security ID is encrypted using hash function SHA-1
- Utility and API for fast implementation of custom applications

Environments

System Architecture

X86 Systems.

Support BIOS Type

Flash Size (1M, 2M, 4M, 16M)KB

Flash Type (1M ROM)

Note: The Standard BIOS don't contain Security ID space, please contact Advantech local sales to request a customize BIOS file.

Support OS

1. Windows XP Professional
2. Windows XP Embedded Standard
3. Windows Embedded Standard

Note: For Other OS support, we will go by project, please contact Advantech local sales.

Tutorial of Using Software Protection

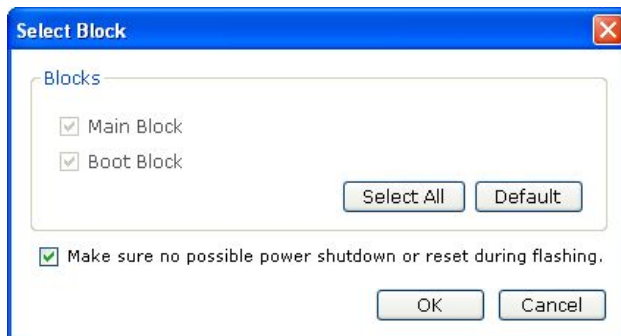
Step1. Install the Utility and Library

Install the Software Protection Utility and Library on an Advantech ePlatform device. The OS must be Window XP-Professional or Windows XP Embedded.

Step2. Security BIOS is necessary

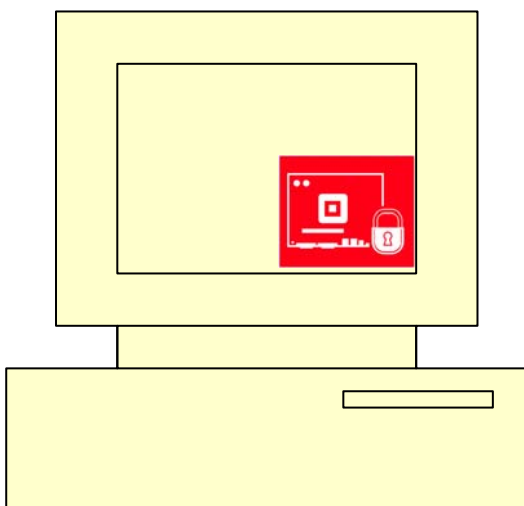
Please flash the Software Protection version image to the BIOS, You can find some evaluation version in the CD "BIOS file" folder. If you don't find the image for your platform, please contact us.

You must flash the whole Software Protection version image (boot block + main block) to the BIOS. If you use BiosFlash to flash the BIOS, you have to choose follow options like this.



Step3. Check the Security Status "Red"

After installation, you can execute an application that is called "CheckSIDStatus.exe", At first times, It will appear a **red** icon on right lower side (system tray), the red mean there are no any Security IDs, Vendor ID or Customer ID, inside in your BIOS.



Input the Vendor ID

Run the Utility “Software Protection”, input the “Authentication No”.

At the first times, the application will take a longer time for execute it, because it will register the hardware information into BIOS.

Click tab “Vendor ID”, input the previous “Vendor ID”, then input the New “Vendor ID”, re-type the New “Vendor ID” again to confirm.

Click “Write”,

Then you can see the writing on status bar.

Software Protection Utility

ADVANTECH

Vendor ID | Customer ID | ID Verification

Vendor ID Update (10 bytes max)

Previous ID: *****

New ID: ***

Confirm New ID: ***

Write

Platform Security Status

SID	Attribute	Status
Board ID	Auto-detect	OK
Vendor ID	Auto-detect	
Customer ID	User-input	

Sample... About... Exit

Note. If you forget the previous ID, you have to re-flash BIOS file. **Vendor ID** default value is “AdvPRJ001”.

Step4. Input the Customer ID

Run the Utility “Software Protection”, input the “Authentication No”.

Click tab “Customer ID”, input the previous “Customer ID”, then input the New “Customer ID”, re-type the New “Customer ID” again to confirm.

Click “Write”.

Then, you can see the writing on status bar.

Software Protection Utility

ADVANTECH

Vendor ID Customer ID ID Verification

Customer ID Update (12 bytes max)

Previous ID: *****

New ID: ***

Confirm New ID: ***

Write

Platform Security Status

SID	Attribute	Status
Board ID	Auto-detect	
Vendor ID	Auto-detect	
Customer ID	User-input	OK

Sample... About... Exit

Note. If you forget the previous ID, you have to re-flash BIOS file. **Customer ID** default value is “AdvCID001”.

Step5. Double check the IDs

After input all IDs, go to tab "ID Verification", type your "Vendor ID" and "Customer ID". Click "Check All", you can check it on status bar.

Software Protection Utility

ADVANTECH

Vendor ID Customer ID **ID Verification**

Check for existing IDs

Vendor ID: (10 Bytes max)

Customer ID: (12 Bytes max)

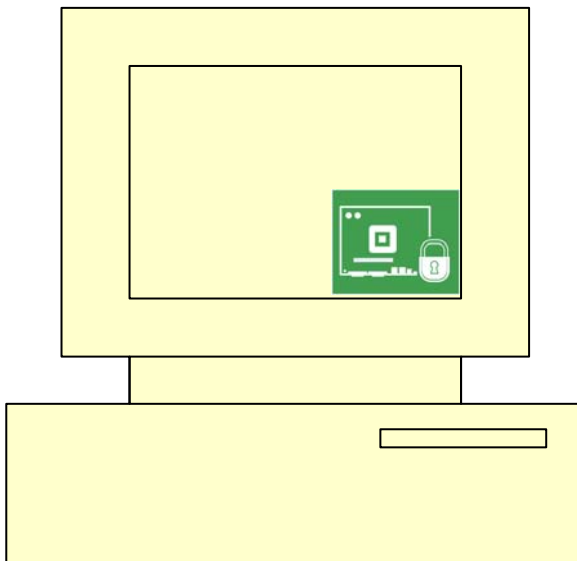
Platform Security Status

SID	Attribute	Status
Board ID	Auto-detect	Pass
Vendor ID	Auto-detect	Pass
Customer ID	User-input	Pass

Step6. Check the Security Status “Green”

After input all IDs, then you can execute “CheckSIDstatus.exe” again, you will see a **green** icon on right lower side, the green mean the Security IDs are ready for application accessing.

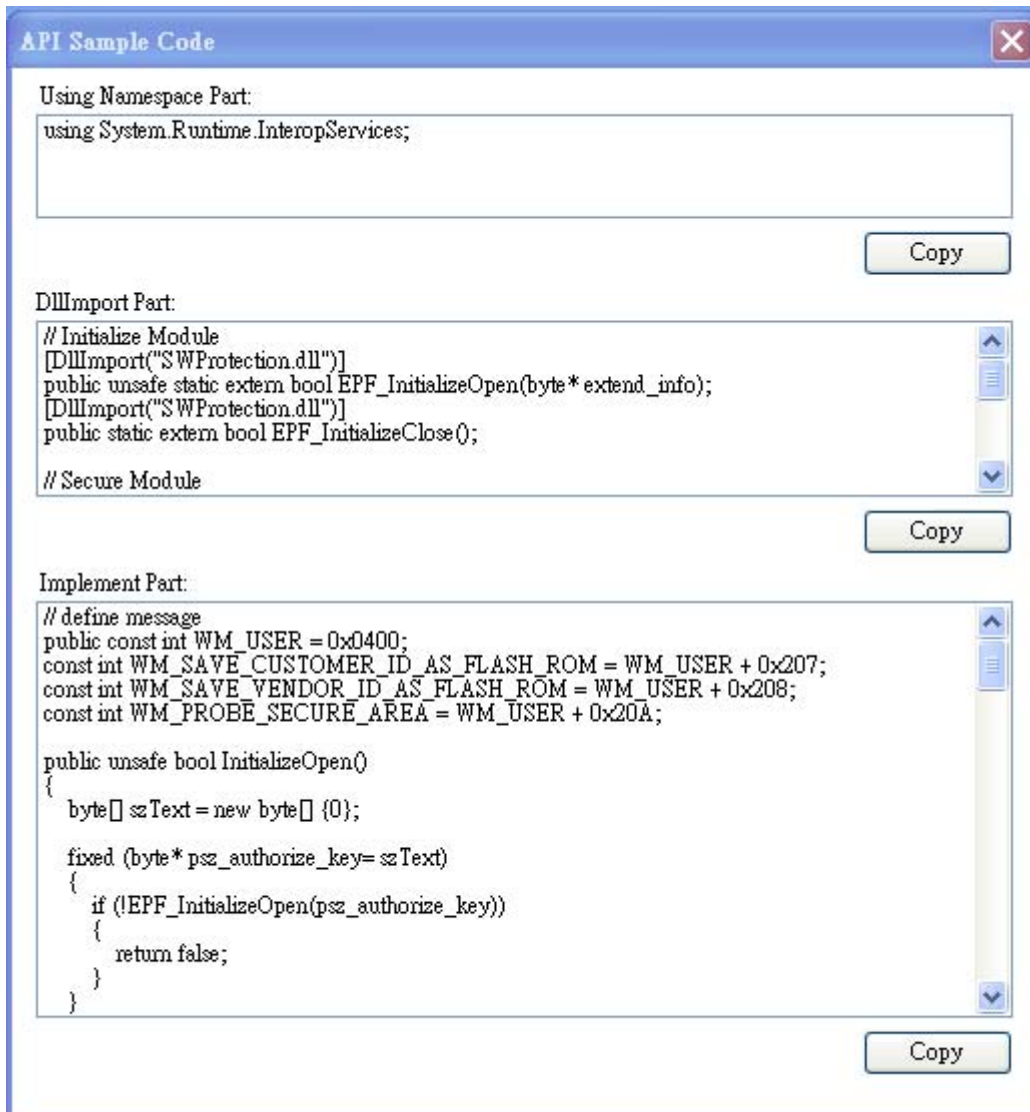
If you double click it, you will execute the SWProtection utility tool directly.



Step7. Write application

Run the Utility “Software Protection”, input the “Authentication No”.

Click tab “Sample Code”, you can get the sample code of how to access Security IDs. Please copy and paste to Microsoft Visual Studio 2005, click the “Build” and then run it on an Advantech ePlatform device.



Congratulations! You have successfully completed this tutorial and created a custom application to access the Security ID on your device, your application now are protected with the device.

Software Protection Program

Installation

Software Protection installation is a **setup** file, please click the setup.exe to do the installation, follow the steps to complete the process.

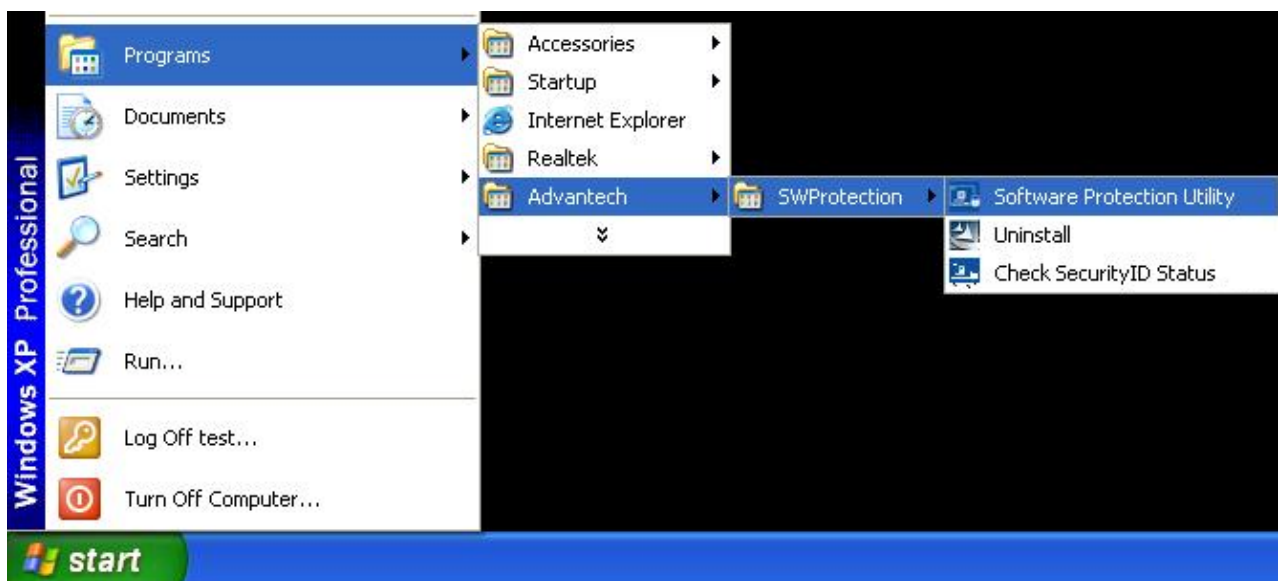
After the installation, you will see

Software Protection Utility,

Uninstall,

Check Security ID Status

in Advantech SWProtection folder.



How to write the Security ID?

1. Run “**Software Protection Utility**”

2. Then Input the Authentication Number

Note: The Authentication Number is on cover of the CD, it is required whenever you launch the application, please don't lose it.

3. Input the Vendor ID

Click tab “Vendor ID”, input the previous “Vendor ID”, then input the New “Vendor ID”, re-type the New “Vendor ID” again to confirm.

Click “Write”, you can see the writing on status bar.

Software Protection Utility

ADVANTECH

Vendor ID Customer ID ID Verification

Vendor ID Update (10 bytes max)

Previous ID: *****

New ID: ***

Confirm New ID: ***

Write

Platform Security Status

SID	Attribute	Status
Board ID	Auto-detect	OK
Vendor ID	Auto-detect	
Customer ID	User-input	

Sample... About... Exit

4. Input the Customer ID

Click tab "Customer ID", input the previous "Customer ID", then input the New "Customer ID", re-type the New "Customer ID" again to confirm.

Click "Write", you can see the writing on status bar.

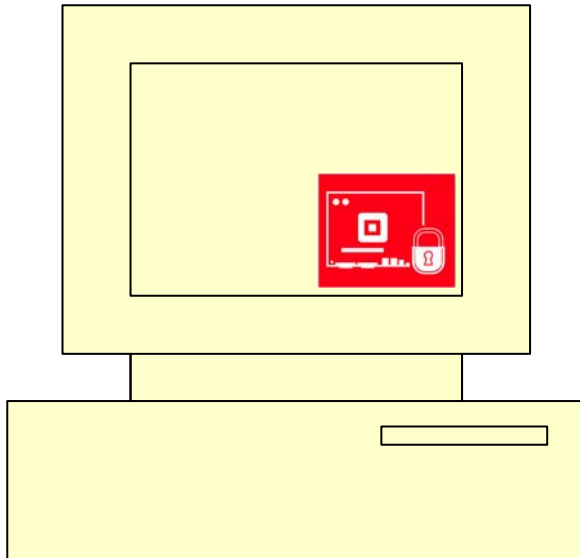
The screenshot shows the 'Software Protection Utility' window with the 'Customer ID' tab selected. The window has a blue title bar and a dark blue header with the 'ADVANTECH' logo. Below the logo are three tabs: 'Vendor ID', 'Customer ID' (selected), and 'ID Verification'. The main content area is divided into two sections. The top section, titled 'Customer ID Update (12 bytes max)', contains three input fields: 'Previous ID:' with a masked value '*****', 'New ID:' with a masked value '***', and 'Confirm New ID:' with a masked value '***'. To the right of these fields is an icon of a server rack. Below the input fields is a 'Write' button. The bottom section, titled 'Platform Security Status', contains a table with three columns: 'SID', 'Attribute', and 'Status'. The table lists three items: 'Board ID' (Auto-detect), 'Vendor ID' (Auto-detect), and 'Customer ID' (User-input). The 'Status' column shows 'OK' for the 'Customer ID' row. At the bottom of the window are three buttons: 'Sample...', 'About...', and 'Exit'.

SID	Attribute	Status
Board ID	Auto-detect	
Vendor ID	Auto-detect	
Customer ID	User-input	OK

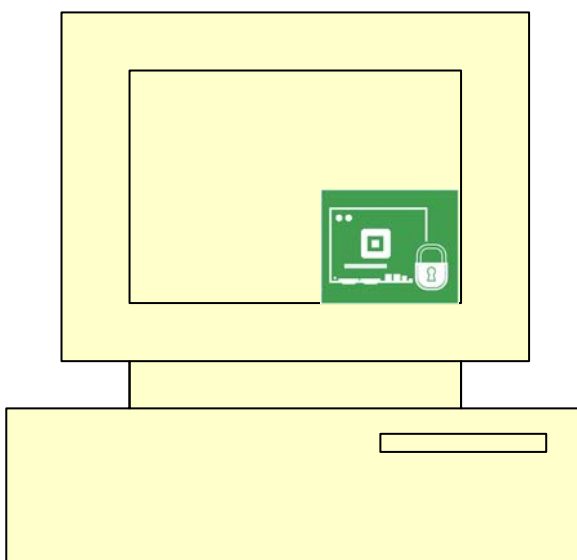
5. Check Security ID Status?

Run **Check Security ID Status**

After installation, It will appear a **red** icon on right lower side (system tray), the red mean there are no any Security IDs, Vendor ID or Customer ID, inside in your BIOS.



After input all IDs, you can execute "**CheckSIDstatus.exe**" again, then you can see a **green** icon on right lower side, the green mean the Security IDs is ready for application accessing.



If you double click it, you will execute the SWProtection utility tool directly.

Note 1: The Board ID will be the first LAN MAC address, it will be written when you first time run the Software Protection Utility. If the board doesn't have MAC ID, we will use "FFFFFFFFFFFF" string to write it to Board ID.

Note 2: Vendor ID: usually for customer to input project name. Ex, you have one project with 100 devices, you can use project 1 in Vendor ID.

Vendor ID default value is "**AdvPRJ001**".

Vendor ID max length: **10 byte**

Note 3: Customer ID: usually for customer to input flow number to identify all available devices. Ex, project 1 with 100 devices, in Customer ID, you can input "device001" for the first devices, "device 002" for second device..."device100" for 100th device.

Customer ID default value is "**AdvCID001**".

Customer ID max length: **12 byte**

How to check the Security ID?

Run the Utility “Software Protection Utility”, input the “Authentication No”.

After input all IDs, go to tab “ID Verification”, type your “Vendor ID” and “Customer ID”.

Click “Check All”, you can check it on status bar.

You need to type the correct ID and it will only show “*****”

Software Protection Utility

ADVANTECH

Vendor ID Customer ID **ID Verification**

Check for existing IDs

Vendor ID: *** (10 Bytes max)

Customer ID: *** (12 Bytes max)

Check All

Platform Security Status

SID	Attribute	Status
Board ID	Auto-detect	Pass
Vendor ID	Auto-detect	Pass
Customer ID	User-input	Pass

Sample... About... Exit

SUSI API Programmer's

Documentation

The library is place on CD\API folder, for detail; please check the "API introduction.txt" and "How to use library.txt".

All APIs return the BOOL data type except Susi*Available and some special cases that are of type int. If any function call fails, i.e. a BOOL value of FALSE, or an int value of -1, the error code can always be retrieved by an immediate call to SusiGetLastError.

[Initialize Module:]

(1) bool EPF_InitializeOpen

bool EPF_InitializeOpen (void extend_info1)*

Description: Initialize

Parameter: extend_info1, [OUT], a reserved parameter, you can set a empty string.

Return: true (1), false (0)

(2) bool EPF_InitializeClose

bool EPF_InitializeClose ()

Description: Un-Initialize

Parameter: None

Return: true (1), false (0)

[Secure Module:]

(3) **bool EPF_SetCustomerIDData**

```
bool EPF_SetCustomerIDData (char* secure_string,
                             char* old_secure_string,
                             HWND hWnd,
                             UINT msgID);
```

Description: Set user define string into this field. (less than 12 characters)

Parameter: secure_string, [OUT], set secure string into BIOS.

old_secure_string, [OUT], Which customer id be insided.

hWnd, [Out], Assign the progress bar on which windows handle.

msgID, [Out], Assign one kind of action.

MsgID: 0x0400 + 0x207

Return: true (1), false (0)

(4) **bool EPF_SetSecureVendorIDData**

```
bool EPF_SetSecureVendorIDString (char* vendor_id_string,
                                   char* old_vendor_id_string,
                                   HWND hWnd,
                                   UINT msgID);
```

Description: Set user define string into this field. (less than 10 characters)

Parameter: secure_string, [OUT], set secure string into BIOS.

old_secure_string, [OUT], Which vendor id be insided.

hWnd, [Out], Assign the progress bar on which windows handle.

msgID, [Out], Assign one kind of action.

MsgID: 0x0400 + 0x208

Return: true (1), false (0)

(5) bool EPF_CheckSecureID

```
bool EPF_SetSecureVendorIDString (char* chk_result,
                                  char* vid_io_buf,
                                  char* cid_io_buf,
                                  HWND hWnd,
                                  UINT msgID);
```

Description: Check the all of securityID wheher valid or not.

Parameter: chk_result, [IN], The check result of security area, The value mean is showing below.

XXXXXXX1 : BOARD ID is fail

XXXXXX1X : Vendor ID is fail

XXXXX1XX : Customer ID is fail

X is don't care, you can mask these bits.

vid_io_buf, [OUT], Which vendor id be insided.

cid_io_buf, [OUT], Which customer id be insided.

hWnd, [Out], Assign the progress bar on which windows handle.

msgID, [Out], Assign one kind of action.

Message ID: 0x0400 + 0x20A

WM_VERIFY_BIN_FILE_AND_FLASH_ROM = 0x0400 + 0x206;

Return: true (1), false (0)

Appendix

Supported BIOS Description

Flash Size (1M,2M,4M,16M)KB

Flash Type (1M ROM)

(AMD, ATMEL, CSI, INTEL, MOSEL, MX_AP, MX_P, SST, AMIC, WIN)

```
{AMD_CHIP_ID,      "AMD 29F010 /5V      "},
{ATMEL_CHIP_ID_1,  "ATMEL 29C010A /5V      "},
{ATMEL_CHIP_ID_3,  "ATMEL 49F001T /5V      "},
{CSI_CHIP_ID,      "CSI CAT28F001P /12V     "},
{INTEL_CHIP_ID,     "INTEL 28F001BX-T /12V   "},
{MOSEL_1M_CHIP_ID, "MOSEL V29C51001T /5V   "},
{MX_AP_CHIP_ID,     "MXIC 28F1000AP /12V     "},
{MX_P_CHIP_ID,      "MXIC 28F1000P /12V      "},
{MXIC_29F001T_ID,  "MXIC 29F001T /5V       "},
{SST_CHIP_ID,       "SST 28EE010 & 28EE011 /5V"},
{SST_CHIP_ID_1,     "SST 29EE010/5V         "},
{SST_39SF010_CHIP_ID, "SST 39SF010 /5V       "},
{AMIC_A29001_ID,    "AMIC A29001 /5V        "},
{WIN_CHIP_ID,       "WINBOND 29EE011 /5V     "},
```

(AMD, AMIC, ATMEL, BM, CSI, EN, GTK, HY, IMT, INTEL, MOSEL, WINBOND, EFST, WIN, SST, PMC, ST, MXIC, PMC, TI)

Flash Type (2M ROM)

```
{AMD_2M_CHIP_ID,   "AMD 29F002(N)T /5V     "},
{AMIC_A29002_ID,    "AMIC A29002 /5V        "},
{ATMEL_2M_1_CHIP_ID, "ATMEL 49F002T /5V      "},
{ATMEL_2M_2_CHIP_ID, "ATMEL 29LV020 /3V      "},
{ATMEL_2M_CHIP_ID,  "ATMEL 29C020 /5V       "},
{BM_2M_CHIP_ID,     "BRIGHT BM29FS020 /5V   "},
{CSI_2M_CHIP_ID,     "CSI CAT28F002T /12V     "},
{EN_29F002_ID,      "EN EN29F002NT /5V      "},
{GTK_020_CHIP_ID,   "ARF35LV020             "},
{GTK_022_CHIP_ID,   "AVF35LV020             "},
{HY_29F002T_ID,     "HYUNDAI HY29F002T /5V  "},
{IMT_2M_CHIP_ID,    "IMT IM29F002T /5V      "},
{INTEL_2M_CHIP_ID,   "INTEL 28F002BX-T /12V   "},
```

```

{MOSEL_2M_CHIP_ID, "MOSEL V29C51002T /5V"},
{MOSEL_2M_V29LC51002_ID, "MOSEL V29LC51002T /5V"},
{WINBOND_49F002T_CHIP_ID, "WINBOND 49F002U /5V"},
{WINBOND_39L020_CHIP_ID, "WINBOND 29L020 /3.3V"},
{EFST_F49B002UA_CHIP_ID, "EFST F49B002UA /5V"},
{WIN_49V002_CHIP_ID, "WINBOND 49V002 /3.3V"},
{SST_49LF020_CHIP_ID, "SST 49LF020 LPC /3.3V"},
{SST_49LF020A_CHIP_ID, "SST 49LF020A LPC /3.3V"},
{PMC_49LP002_Chip_ID, "PMC Pm49LP002 LPC /3.3V"},
{PMC_Pm49FL002T_Chip_ID, "PMC Pm49FL002T LPC/FWH"},
{ST_M50FW002_ID, "ST M50FW002 FWH"},
{ST_M50LPW002_ID, "ST M50LPW002 LPC"},
{WIN_49V002F_ID, "WINBOND 49V002F /3.3V"},
{ATMEL_AT49LL020_ID, "ATMEL AT49LL020 2Mb LPC"},
{SST_49LF003A_CHIP_ID, "SST 49LF003A 3Mb /3.3V"},
{SST_49LF030A_CHIP_ID, "SST 49LF030A 3Mb /3.3V"},
{MXIC_2000PPC_ID, "MXIC 28F2000PPC /12V"},
{MXIC_2000TPC_ID, "MXIC 28F2000TPC /12V"},
{MXIC_2M_2_CHIP_ID, "MXIC 28F002TTC /12V"},
{MXIC_29F002T_ID, "MXIC 29F002(N)T /5V"},
{MXIC_29F022T_ID, "MXIC 29F022(N)T /5V"},
{PMC_2M_CHIP_ID, "PMC PM29F002T /5V"},
{PMC_39F020_CHIP_ID, "PMC PM39F020 /5V"},
{SST_2M_CHIP_ID, "SST 29EE020 /5V"},
{SST_2M_1_CHIP_ID, "SST 29LE020 /3V"},
{SST_39SF020_CHIP_ID, "SST 39SF020 /5V"},
{SST_39VF020_CHIP_ID, "SST 39VF020 /3.3V"},
{SST_49LF002_CHIP_ID, "SST 49LF002A /3.3V (2Mb)"},
{ST_2M_CHIP_ID, "ST M29F002T /5V"},
{TI_2M_CHIP_ID, "INTEL/TI TMS28F020 /12V"},
{WINBOND_2M_CHIP_ID, "WINBOND 29C020 /5V"},

```

Flash Type (4M ROM)

(AMD, HY, ATMEL, GTK, BM, PMC, BMB, MOSEL, MXIC)

```

{AMD_4M_CHIP_ID, "AMD 29F400BT /5V"},
{HY_29F040A_ID, "HYUNDAI HY29F040A /5V"},
{AMD_16M_CHIP_ID, "AMD 29F160D /5V"},
{BMB_16M_CHIP_ID, "MBM 29F160 /5V"},
{ATMEL_29C040_ID, "ATMEL 29C040A /5V"},

```

```

{GTK_040_CHIP_ID,"AVF35LV040"},
{BM_29F040_ID, "BRIGHT BM29FS040 /5V"},
{Bright_BM29F040_ID, "BRIGHT BM29F040 /5V"},
{PMC_39F040_ID, "PMC PM39F040 /5V"},
{PMC_PM29F004T_ID, "PMC Pm29F004T /5V"},
{ATMEL_AT49F040T_ID, "ATMEL 49F040T /5V"},
{EN_29F040_ID, "EN EN29F040 /5V"},
{BMB_29F040_ID, "Fujitsu BMB29F040C /5V"},
{MOSEL_29C51004_ID, "MOSEL 29C51004T /5V"},
{MXIC_29F004_CHIP_ID,"MXIC 29F004T /5V"},
(AMD, HY, ATMEL, GTK, BM, PMC, BMB, MOSEL, MXIC, INTEL, SST, WINBOND, ST, MegaWin, AMIC,
IMT)
{INTEL_E8280AD_ID, "INTEL E82802AB /3.3V(4Mb)"},
{INTEL_E82F400B5T_ID, "INTEL E82F400B5"},
{SST_49LF004_CHIP_ID, "SST 49LF004 /3.3V"},
{SST_49LF004A_CHIP_ID, "SST 49LF004A/B /3.3V"},
{Winbond_FWH_W39V040A_Chip, "Winbond W39V040FA (4Mb)"},
{Winbond_LPC_W39V040AP_Chip, "Winbond W39V040AP (4Mb)"},
{PMC_Pm49FL004T_Chip_ID, "PMC Pm49FL004T LPC/FWH"},
{ATMEL_AT49LW040_ID, "ATMEL AT49LW040 4Mb FWH"},
{ST_M29W040B_ID, "ST M29W040B /3V"},
{ST_M29F040B_ID, "ST M29F040B /5V"},
{ATMEL_AT49LL040_ID, "ATMEL AT49LL040 4Mb LPC"},
{SST_49LF040_CHIP_ID, "SST 49LF040A LPC /3.3V"},
{SST_28SF040A_ID, "SST 28SF040A /5V"},
{ST_M29F400T_ID, "ST M29F400T /5V"},
{WINBOND_29C040_ID, "WINBOND 29C040 /5V"},
(AMD, HY, ATMEL, GTK, BM, PMC, BMB, MOSEL, MXIC, INTEL, SST, WINBOND, ST, MegaWin, AMIC,
IMT)
{WINBOND_39L040_CHIP_ID, "WINBOND 29L040 /3.3V"},
{MegaWin_MM29F040_ID, "AMD AM29F040B /5V"},
{MegaWin_MM29LF040_ID, "MEGAWIN MM29LF040 /3.3V"},
{MXIC_MX29F040_ID, "MXIC MX29F040 /5V"},
{AMIC_A29040_ID, "AMIC A29040 /5V"},
{ST_M50FW040_ID, "ST M50FW040 /3V"},
{ST_M50LPW040_ID, "ST M50LPW040 /3V"},
{ST_M50LPW041_ID, "ST M50LPW041 /3V"},
{SST_39SF040_ID, "SST 39SF040 /5V"},
{SST_39SF040P_ID, "SST 39SF040P /5V"},

```

```

{SST_39VF040P_ID, "SST 39VF040P /5V      "},
{IMT_4M_CHIP_ID,   "IMT IM29F004T /5V      "},
{INTEL_E8280AC_ID,  "INTEL E82802AC /3.3V(8Mb)"},
{SST_49LF008_CHIP_ID, "SST 49LF008A /3.3V      "},
{SST_49LF080A_CHIP_ID, "SST 49LF080A /3.3V      "},
(AMD, HY, ATMEL, GTK, BM, PMC, BMB, MOSEL, MXIC, INTEL, SST, WINBOND, ST, MegaWin, AMIC,
IMT)
{ST_M50LPW080_ID,    "ST M50LPW080 8Mb LPC /3V "},
{ST_M50FW080_ID,     "ST M50FW080 8Mb FWH /3V  "},
{ATMEL_AT49LW080_ID, "ATMEL AT49LW080 8Mb FWH  "},
Flash Type (16M ROM)
{0x25bf, "SST 25VF016B 16Mb SPI    ", SPI}}

```