UEFI Shell Application Examples

Finnbarr P. Murphy
(fpm@fpmurphy.com)

The three examples listed in the EFI Shell Developers Guide (the latest version is 0.91 dated June 27, 2005) have never been updated to reflect the TianoCore UEFI development environment. I have modified the sources for the three examples so that they now successfully build in the TianoCore EDK. This post assumes that you are familiar with UEFI and the EFI Developers Kit. It does not attempt to explain the content of the files. Please read the EFI Shell Developers Guide if you need a detailed explanation.

Example 1 – HelloWorld

```
//
//  HelloWorld.h
//
#ifndef _EFI_HELLOWORLD_H_
#define _EFI_HELLOWORLD_H_
#define EFI_HELLOWORLD_GUID {0x6435f523, 0x5cd3, 0x444f, 0xa4, 0x23, 0x83, 0x4f, 0xbb, 0x7
1, 0x29, 0xc7}
#endif
```

```
//
// HelloWorldStrings.uni
//
/=#
#langdef eng "English"
#string STR_HELLO      #language eng  "Hello World\n"
#string STR_GOODBYE    #language eng  "Goodbye World\n"
```

```
//
//  HelloWorld.c
//
#include "EfiShellLib.h"
#include "HelloWorld.h"
#include STRING_DEFINES_FILE
extern UINT8 STRING_ARRAY_NAME[];
EFI_HII_HANDLE HiiHandle;
EFI_GUID       HelloGuid = EFI_HELLOWORLD_GUID;
EFI_BOOTSHELL_CODE(
  EFI_APPLICATION_ENTRY_POINT(HelloMain)
)
EFI_STATUS
HelloMain ( IN EFI_HANDLE       ImageHandle,
            IN EFI_SYSTEM_TABLE *SystemTable)
{
   EFI_SHELL_APP_INIT (ImageHandle, SystemTable);
   EFI_SHELL_STR_INIT (HiiHandle, STRING_ARRAY_NAME, HelloGuid);
   PrintToken (STRING_TOKEN(STR_HELLO), HiiHandle);
   PrintToken (STRING_TOKEN(STR_GOODBYE), HiiHandle);
   LibUnInitializeStrings();
```

For personnal use only

 Copyright 2004-2013 Finnbarr P. Murphy. All rights reserved.

```
   return EFI_SUCCESS;
}
```

```
#
# HelloWorld.inf
#
# Component description file
#
[defines]
   BASE_NAME      = HelloWorld
   FILE_GUID      = 0B2E45BA-3539-4a2a-B853-3723E3CAAB1B
   COMPONENT_TYPE = APPLICATION
[sources.common]
   HelloWorldStrings.uni
   HelloWorld.c
[includes.common]
   .
   ..\Inc
   ..\Library
   $(EDK_SOURCE)\Foundation
   $(EDK_SOURCE)\Foundation\Core\Dxe
   $(EDK_SOURCE)\Foundation\Efi
   $(EDK_SOURCE)\Foundation\Efi\Include
   $(EDK_SOURCE)\Foundation\Framework
   $(EDK_SOURCE)\Foundation\Framework\Include
   $(EDK_SOURCE)\Foundation\Include
   $(EDK_SOURCE)\Foundation\Include\IndustryStandard
   $(DEST_DIR)\
[libraries.common]
   EfiShellLib
[nmake.common]
   IMAGE_ENTRY_POINT = HelloMain
   C_STD_FLAGS=$(C_STD_FLAGS) /DSTRING_ARRAY_NAME=$(BASE_NAME)Strings
   C_STD_FLAGS=$(C_STD_FLAGS) /DSTRING_DEFINES_FILE=\"$(BASE_NAME)StrDefs.h\"
   C_STD_FLAGS=$(C_STD_FLAGS) /DEFI_BOOTSHELL
```

Example 2 – UseArg

```
//
// UseArg.c
//
#include "EfiShellLib.h"
#ifdef EFI_BOOTSHELL
EFI_APPLICATION_ENTRY_POINT (UseArgMain)
#endif
EFI_STATUS
UseArgMain ( IN EFI_HANDLE       ImageHandle,
             IN EFI_SYSTEM_TABLE *SystemTable)
{
   UINTN Index;
   EFI_SHELL_APP_INIT (ImageHandle, SystemTable);
   Print (L"The arguments are:\n");
   for (Index = 0; Index < SI->Argc; Index++) {
      Print (L"%s\n", SI->Argv[Index]);
   }
   return EFI_SUCCESS;
}
```

For personal use only

 Copyright 2004-2013 Finnbarr P. Murphy. All rights reserved.

```
#
# UseArg.inf
#
# Component description file
#
[defines]
   BASE_NAME       = UseArg
   FILE_GUID       = 181157FA-77B2-4798-B3F7-003756ECE6ED
   COMPONENT_TYPE  = APPLICATION
[sources.common]
   UseArg.c
[includes.common]
      .
   ..\Inc
   ..\Library
   $(EDK_SOURCE)\Foundation
   $(EDK_SOURCE)\Foundation\Core\Dxe
   $(EDK_SOURCE)\Foundation\Efi
   $(EDK_SOURCE)\Foundation\Efi\Include
   $(EDK_SOURCE)\Foundation\Framework
   $(EDK_SOURCE)\Foundation\Framework\Include
   $(EDK_SOURCE)\Foundation\Include
   $(EDK_SOURCE)\Foundation\Include\IndustryStandard
   $(DEST_DIR)\
[libraries.common]
   EfiShellLib
[nmake.common]
   IMAGE_ENTRY_POINT=UseArgMain
   C_STD_INCLUDE=
   C_STD_FLAGS=$(C_STD_FLAGS) /DEFI_BOOTSHELL
```

Example 3 - UseVar

```
//
// UseVar.h
//
#ifndef _EFI_USEVAR_H_
#define _EFI_USEVAR_H_
#define VAR_NAME L"SampleVar"
#define EFI_USEVAR_GUID {0xe35ab40b, 0x1116, 0x43a3, 0x91, 0x41, 0xa3, 0xcd, 0xa8, 0x85, 0
x1, 0xb7}
#define VAR_GUID {0x11299421, 0xa2a5, 0x4206, 0xb1, 0x93, 0x9c, 0xf3, 0xe8, 0x21, 0x29, 0x
5c}
#endif
```

```
//
// UseVar.c
//
#include "EfiShellLib.h"
#include "UseVar.h"
#include STRING_DEFINES_FILE
extern UINT8 STRING_ARRAY_NAME[];
// prototypes
BOOLEAN  IsQuoted (void);
EFI_STATUS OutputVar (EFI_HII_HANDLE);
EFI_STATUS WriteVar (CHAR16 *, EFI_HII_HANDLE);
#ifdef EFI_BOOTSHELL
EFI_APPLICATION_ENTRY_POINT (UseVarMain)
#endif
```

For personnal use only

 Copyright 2004-2013 Finnbarr P. Murphy. All rights reserved.

```
EFI_STATUS
UseVarMain ( IN EFI_HANDLE       ImageHandle,
             IN EFI_SYSTEM_TABLE *SystemTable)
{
   EFI_HII_HANDLE HiiHandle;
   EFI_STATUS     Status  = EFI_SUCCESS;
   EFI_GUID       StrGuid = EFI_USEVAR_GUID;
   EFI_SHELL_APP_INIT (ImageHandle, SystemTable);
   EFI_SHELL_STR_INIT (HiiHandle, STRING_ARRAY_NAME, StrGuid);
   switch (SI->Argc) {
      case 1:
         Status = OutputVar (HiiHandle);
         break;
      case 2:
         if (!StriCmp (SI->Argv[1], L"-?") &amp;&amp; !IsQuoted ()) {
            PrintToken (STRING_TOKEN(STR_USAGE), HiiHandle);
            break;
         }
         Status = WriteVar (SI->Argv[1], HiiHandle);
         break;
      default:
         PrintToken (STRING_TOKEN(STR_USAGE), HiiHandle);
         break;
   }
   LibUnInitializeStrings ();
   return Status;
}
EFI_STATUS
OutputVar (EFI_HII_HANDLE HiiHandle)
{
   EFI_STATUS  Status;
   EFI_GUID    VarGuid = VAR_GUID;
   UINTN       BufferSize = 0;
   VOID        *Buffer = NULL;
   // Pass in a zero size buffer to find the required buffer size.
   Status = RT->GetVariable (VAR_NAME,
                             &amp;VarGuid,
                             NULL,
                             &amp;BufferSize,
                             Buffer);
   if (Status == EFI_BUFFER_TOO_SMALL) {
      // Allocate the buffer to hold the content of variable
      Status = BS->AllocatePool (EfiBootServicesData,
                                 BufferSize,
                                 &amp;Buffer);
      if (EFI_ERROR (Status)) {
         return Status;
      }
      //
      // Read variable into the allocated buffer.
      //
      Status = RT->GetVariable (VAR_NAME,
                                &amp;VarGuid,
                                NULL,
                                &amp;BufferSize,
                                Buffer);
      if (EFI_ERROR (Status)) {
          return Status;
      }
   } else if (EFI_NOT_FOUND == Status) {
      PrintToken (STRING_TOKEN(STR_NOT_FOUND), HiiHandle);
      return Status;
   }
   PrintToken (STRING_TOKEN(STR_TITLE), HiiHandle);
   PrintToken (STRING_TOKEN(STR_VAR), HiiHandle, Buffer);
   BS->FreePool (Buffer);
   return Status;
}
```

For personal use only

 Copyright 2004-2013 Finnbarr P. Murphy. All rights reserved.

```
EFI_STATUS
WriteVar (CHAR16          *Str2Write,
          EFI_HII_HANDLE  HiiHandle)
{
   EFI_STATUS  Status  = EFI_SUCCESS;
   EFI_GUID    VarGuid = VAR_GUID;
   UINTN       VarSize;
   if (NULL == Str2Write) {
      return EFI_INVALID_PARAMETER;
   }
   // Calculate the size of the string to write
   // including the end NULL
   VarSize = (StrLen (Str2Write) + 1) * sizeof (CHAR16);
   Status = RT->SetVariable (VAR_NAME,
                             &amp;VarGuid,
                             EFI_VARIABLE_BOOTSERVICE_ACCESS|EFI_VARIABLE_RUNTIME_ACCESS|E
FI_VARIABLE_NON_VOLATILE,
                             VarSize,
                             Str2Write);
   if (EFI_ERROR (Status)) {
       PrintToken (STRING_TOKEN(STR_WRITE_FAIL), HiiHandle);
   }
   return Status;
}
BOOLEAN
IsQuoted (void)
{
   EFI_SHELL_ARG_INFO *ArgInfo;
   // Attribute of ARG_IS_QUOTED means that at least part of the
   // argument was quoted in a pair of quotation marks.
   // Attribute of ARG_PARTIALLY_QUOTED means that not all the
   // chararcters of the argument were quoted.
   ArgInfo = SI->ArgInfo;
   if (ArgInfo[1].Attributes &amp; ARG_IS_QUOTED
       &amp;&amp; !(ArgInfo[1].Attributes &amp; ARG_PARTIALLY_QUOTED)) {
          return TRUE;
   }
   return FALSE;
}
```

```
/
// UseVarStrings.uni
//
/=#
#langdef   eng "English"
#langdef   fra "Français"
#string STR_TITLE
#language eng  "%HThe content of variable 'SampleVar' is:%N\n"
#language fra  "%HThe content of variable 'SampleVar' is:%N\n"
#string STR_VAR
#language eng  "%s\n"
#language fra  "%s\n"
#string STR_NOT_FOUND
#language eng  "Variable %H'SampleVar'%N not found!\n"
#language fra  "Variable %H'SampleVar'%N not found!\n"
#string STR_WRITE_FAIL
#language eng  "Fail to write variable %H'SampleVar'%N.\n"
#language fra  "Fail to write variable %H'SampleVar'%N.\n"
#string STR_USAGE
#language eng  "USEVAR [string | -?]\n"
               "\n"
               "      string - String will be written in variable\n"
               "               named 'SampleVar'.\n"
               "      -?     - Display the help information.\n"
```

For personal use only

 Copyright 2004-2013 Finnbarr P. Murphy. All rights reserved.

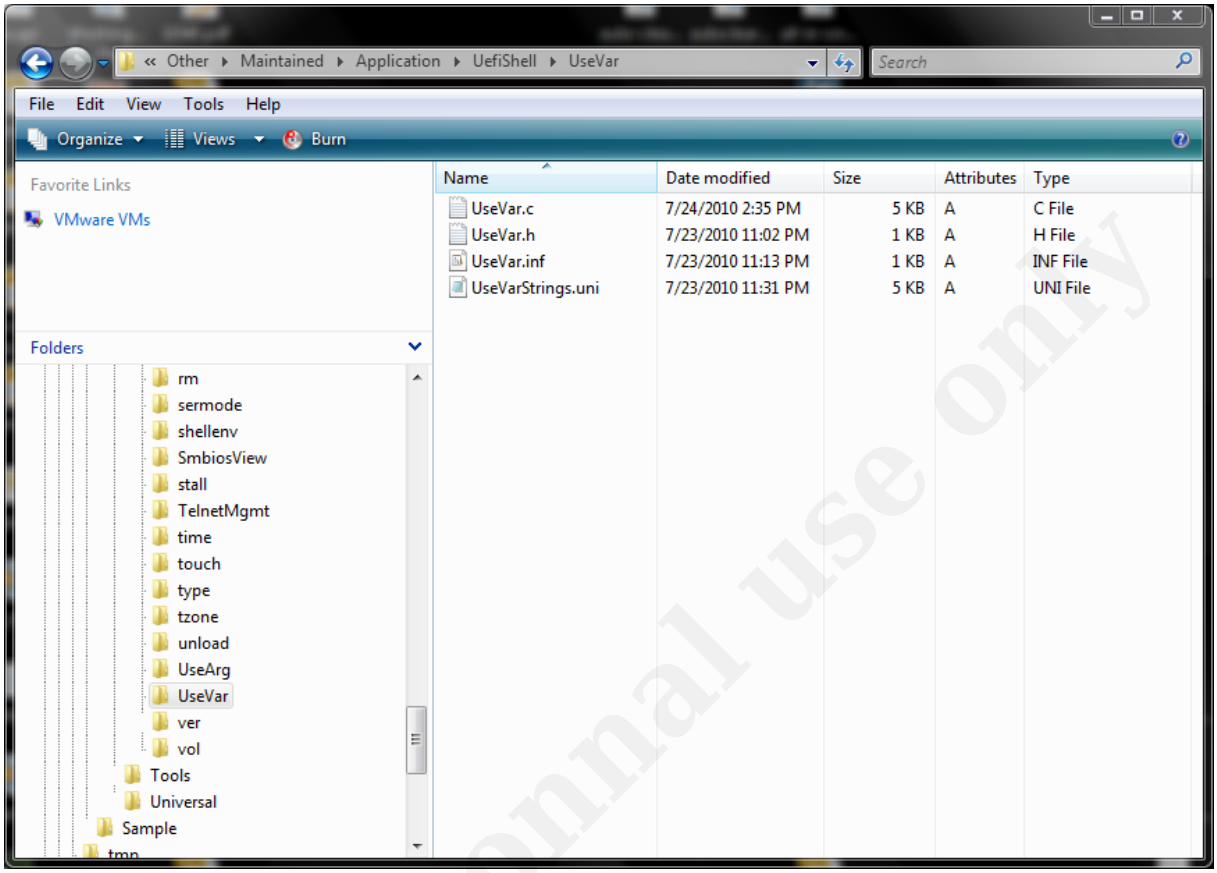
```
                "\n"
                "Examples:\n"
                " *To write a string into variable 'SampleVar':\n"
                "  fs0:\\> usevar \"EFI, One generation ahead!\"\n"
                "\n"
                " *To display the content of variable 'SampleVar':\n"
                "   fs0:\\> usevar\n"
                "   The content of variable 'SampleVar' is:\n"
                "   EFI, One generation ahead!\n"
 #language fra  "USEVAR [string | -?]\n"
                "\n"
                "      string - String will be written in variable\n"
                "               named 'SampleVar'.\n"
                "      -?     - Display the help information.\n"
                "\n"
                "Examples:\n"
                " *To write a string into variable 'SampleVar':\n"
                "  fs0:\\> usevar \"EFI, One generation ahead!\"\n"
               "\n"
                " *To display the content of variable 'SampleVar':\n"
                "  fs0:\\> usevar\n"
                "  The content of variable 'SampleVar' is:\n"
                "  EFI, One generation ahead!\n"
#string STR_SHELLENV_GNC_COMMAND_NOT_SUPPORT
#language eng  "%hs: This command is not supported in the \n"
               "underlaying EFI version, required EFI version \n"
               "should be %hd.%hd or above\n\n"
#language fra  "%hs: This command is not supported in the \n"
               "underlaying EFI version, required EFI version \n"
               "should be %hd.%hd or above\n\n"
```

```
#
# UseVar.inf
#
# Component description file
#
[defines]
   BASE_NAME      = UseVar
   FILE_GUID      = 7BD2613C-4EF8-49f5-876D-DF94A42B9B09
   COMPONENT_TYPE = APPLICATION
[sources.common]
   UseVarStrings.uni
   UseVar.c
[includes.common]
   .
   ..\Inc
  ..\Library
   $(EDK_SOURCE)\Foundation
   $(EDK_SOURCE)\Foundation\Core\Dxe
   $(EDK_SOURCE)\Foundation\Efi
   $(EDK_SOURCE)\Foundation\Efi\Include
   $(EDK_SOURCE)\Foundation\Framework
   $(EDK_SOURCE)\Foundation\Framework\Include
   $(EDK_SOURCE)\Foundation\Include
   $(EDK_SOURCE)\Foundation\Include\IndustryStandard
   $(DEST_DIR)\
[libraries.common]
   EfiShellLib
[nmake.common]
   IMAGE_ENTRY_POINT=UseVarMain
   C_STD_INCLUDE=
   C_STD_FLAGS=$(C_STD_FLAGS) /DSTRING_ARRAY_NAME=$(BASE_NAME)Strings
   C_STD_FLAGS=$(C_STD_FLAGS) /DSTRING_DEFINES_FILE=\"$(BASE_NAME)StrDefs.h\"
   C_STD_FLAGS=$(C_STD_FLAGS) /DEFI_BOOTSHELL
```

For personal use only

 Copyright 2004-2013 Finnbarr P. Murphy. All rights reserved.

The source files for each example should be placed in their own subdirectory under *${EDK_SOURCE}/Other/Maintained/Application/UefiShell*. For example here is the the filesystem layout for the *UseVar* example.



The EDK Shell Developers Guide contents incorrect information regarding configuring a build tip and where the resultant executables are placed by the buidl.

The appropriate build tip description file (*.dsc*) needs to be modified to include the three examples in the build. For example, if you are building DUET, you would modify *${EDK_SOURCE}/Sample/Platform/DUET/build/DUET.dsc* as follows:

```
#
# Shell
#
Other\Maintained\Application\UefiShell\ShellFull.inf  FV=NULL
Other\Maintained\Application\UefiShell\HelloWorld\HelloWorld.inf  FV=NULL
Other\Maintained\Application\UefiShell\UseArg\UseArg.inf FV=NULL
Other\Maintained\Application\UefiShell\UseVar\UseVar.inf FV=NULL
```

Upon successfully building the tip the resultant executable files for the three applications can be found in *${EDK_SOURCE}/bin/DUETuefi32/IA32* for a 32-bit build or *${EDK_SOURCE}/bin/DUETuefi64/IA32* for a 64-bit build.

June 2010 Post updated. TianoCore activity moved to SourceForge in April 2010.

For personal use only

 Copyright 2004-2013 Finnbarr P. Murphy. All rights reserved.

For personnal use only

 Copyright 2004-2013 Finnbarr P. Murphy. All rights reserved.