UEFI——VFR simple use

原创 Practitioner xxl Modified on 2024-10-15 08:32:57 Read 1.6k Collection 20 Likes 15

Article Tags: UEFI bios

1. Introduction to VFR and HFR

VFR (Visual Forms Representation) is a user interface component. In UEFI, there are four main user interface components: Font, Image, Form, and String. Its structure diagram is as follows:

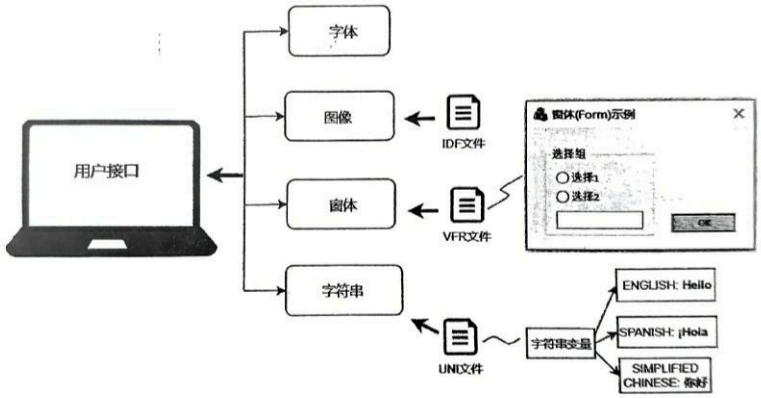
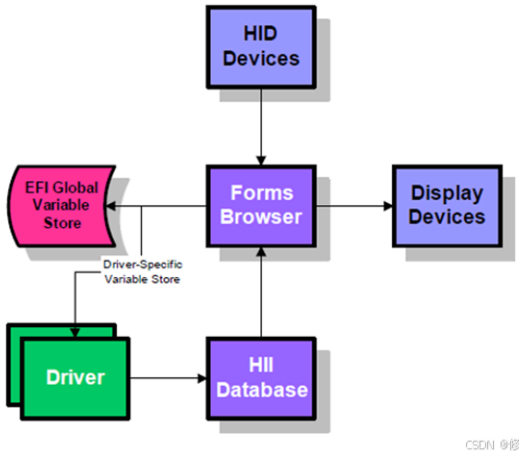


图 3-7 用户接口的组件结构

String is provided by UNI file, and Form is provided by VFR file.

Here are some common concepts used in UEFI image display development:



HII (Human Interface Infrastructure) is a key component of UEFI. It provides a set of standardized APIs and tools for creating, managing, and displaying UEFI graphical user interfaces. The HII database mainly provides interfaces for users to install, uninstall, and use various resources such as strings , fonts, and images.

The Forms Browser acts as a user interface manager in the firmware, allowing users to interact with the UEFI firmware's user interface. It is responsible for displaying information and collecting input and output information and presenting this information to the user.

Drivers and applications install elements such as fonts, strings, images, and forms into the HII database , which acts as a central repository for the entire platform. The Forms Browser uses these elements to render the user interface on the display device and receives information from the user through the HID device . After that, the changes made by the user in the Form Browser will be saved elsewhere, and the control of storage is in the Driver, at the Driver's discretion.

Hii Database consists of Packagelist. Packagelist has a header and contains multiple packages. Package consists of various types of binary files, such as fonts, strings, images, forms, etc.

2. Simple use of VFR files

1. Modify FrontPageStings.uni and FrontPageVfr.Vfr

In UEFI development, the FrontPageStrings.uni file is a file containing string resources for localization of the UEFI interface. The FrontPageVfr.Vfr file is used to define the elements of the UEFI user interface.

Add string resources to MdeModulePkg/Application/UiApp/FrontPageStrings.uni

```
//Add CSDN UI resources
#string STR_CSDN_BANNER_LEFT           #language en-US  "Csdn Bannder Left" //将字符串“Csdn Bannder Left”用STR_BANNER_LEFT表示，语言为美国英语
#string STR_CSDN_BANNER_RIGHT          #language en-US  "Csdn Bannder Right"

#string STR_CSDN_STRING                #language en-US  "CsdnText"
                                       #language fr-FR  "CsdnText"
#string STR_CSDN_STRING_HELP           #language en-US  "CsdnText the current setting."
                                       #language fr-FR  "CsdnText the current setting."

#string STR_CSDN_RADIO_STRING          #language en-US  "CsdnRdioBtn"
                                       #language fr-FR  "CsdnRdioBtn"
#string STR_CSDN_RADIO_STRING_HELP     #language en-US  "CsdnRadio the current setting."
                                       #language fr-FR  "CsdnRadio the current setting."

```

```
#string STR_CSDN_STRING_RADIO_1        #language en-US  "Csdn Radio Button No 1"
#string STR_CSDN_STRING_RADIO_2        #language en-US  "Csdn Radio Button No 2"
#string STR_CSDN_STRING_RADIO_3        #language en-US  "Csdn Radio Button No 3"
#string STR_CSDN_STRING_RADIO_4        #language en-US  "Csdn Radio Button No 4"
#string STR_CSDN_STRING_RADIO_5        #language en-US  "Csdn Radio Button No 5"
#string STR_CSDN_STRING_RADIO_6        #language en-US  "Csdn Radio Button No 6"
#string STR_CSDN_STRING_RADIO_7        #language en-US  "Csdn Radio Button No 7"
```

收起 ^

Modify the Label of FrontPageVfr.Vfr

MdeModulePkg/Application/UiApp/FrontPageVfr.Vfr

AI generated projects 登录复制

```
    banner //横幅
      title = STRING_TOKEN(STR_CSDN_BANNER_LEFT), //横幅的标题为STR_CSDN_BANNER_LEFT所代表的字符串
      line  4, //横幅在屏幕上的垂直距离，在这里横幅位于第4行
      align left; //水平方向左对齐

    banner
      title = STRING_TOKEN(STR_CSDN_BANNER_RIGHT),
      line  4,
      align right;
```

2、修改 FrontPageCustomizedUi.c 与 FrontPageCustomizedUiSupport.c

Add Oneof(RadioButton) menu

MdeModulePkg/Application/UiApp/FrontPageCustomizedUi.c (this file will provide details on what exactly the application does and how to customize the UI)

Oneof is a user interface element that allows the user to select one of a set of predefined options. It is often used to create radio button groups where the user can select only one of the options.

AI generated projects 登录复制

```
/**
  Customize menus in the page.

  @param[in]  HiiHandle             The HII Handle of the form to update.
  @param[in]  StartOpCodeHandle     The context used to insert opcode.
  @param[in]  CustomizePageType     The page type need to be customized.

**/
VOID
UiCustomizeFrontPage (
  IN EFI_HII_HANDLE  HiiHandle,
  IN VOID            *StartOpCodeHandle
  )
{
  //
  // Create "Select CSDN" menu with Oneof opcode.
  //
  UiCreateCsdnRadioMenu(HiiHandle, StartOpCodeHandle);

  ...

  //
  // Create csdn menu.
  //
  UiCreateCsdnMenu(HiiHandle, StartOpCodeHandle);
}

```

收起 ^

The specific implementation of creating CsdnRatioMenu and CsdnMenu is in the FrontPageCustomizedUiSupport.c file:

The specific steps for each operation are included in the MdeModulePkg/Application/UiApp/FrontPageCustomizedUiSupport.c file.

Generates an EFI_IFR_ACTION_OP opcode, which is part of the UEFI Firmware User Interface Framework (IFR).

AI generated projects 登录复制

```
/*
    产生一个EFI_IFR_ACTION_OP操作码，用于定义用户界面的一个问题。当用于与界面元素交互时会被触发
    return: 返回一个指向UINT8数组的指针，这个数组代表了新创建的操作码。
*/
UINT8 *
EFIAPI
HiiCreateActionOpCode (
  IN VOID             *OpCodeHandle, //操作码缓冲区的句柄
  IN EFI_QUESTION_ID  QuestionId, //要定义的问题的ID
  IN EFI_STRING_ID    Prompt, //用于指定问题提示，当问题在用户界面中呈现时，这个提示被显示给用户。
  IN EFI_STRING_ID    Help, //指定问题帮助信息
  IN UINT8            QuestionFlags, //定义问题的各种属性
  IN EFI_STRING_ID    QuestionConfig //用于指定与问题相关的配置数据
  );
```

收起 ^

You can use HiiAllocateOpCodeHandle to allocate a new opcode, and finally use HiiFreeOpCodeHandle() to release the opcode.

Use EFI_GET_VARIABLE GetVariable; to get the value of a variable

AI generated projects 登录复制

```
/*
    返回变量的值，EFI变量是固件运行时可以存储和访问的键值对数据

*/
typedef
EFI_STATUS
(EFIAPI *EFI_GET_VARIABLE)(
  IN     CHAR16                      *VariableName, //一个以NULL结尾的字符串的指针，它是要检索的变量的名称
  IN     EFI_GUID                    *VendorGuid, //指向EFI_GUID的指针，用于标识变量的所有者，这个GUID用于确保正确访问变量
  OUT    UINT32                      *Attributes     OPTIONAL, 输出参数、如果不是 NULL，则指向内存位置的指针，以返回变量的属性位掩码。
  IN OUT UINTN                       *DataSize, //输出参数，Data缓冲区的大小
  OUT    VOID                        *Data           OPTIONAL //输出参数，用于返回变量内容的缓冲区。可以是 NULL 且 DataSize 为零，以便确定所需的缓冲区大小。
  );
```

收起 ^

Use HiiCreateOneOfOptionOpCode to create a radio button of type EFI_IFR_ONE_OF_OPTION_OP. The code prototype is:

AI generated projects 登录复制

```
/**
    创建一个Oneof的操作码（可以理解为单选按钮组）
**/
UINT8 *
EFIAPI
HiiCreateOneOfOptionOpCode (
  IN VOID    *OpCodeHandle, //操作码缓冲区的句柄
  IN UINT16  StringId, //单选按钮的字符串ID，即单选按钮的名称
  IN UINT8   Flags, //单选按钮的属性，是否只读、由默认选择等
  IN UINT8   Type, //单选按钮的类型，按钮的不同行为或外观
  IN UINT64  Value //单选按钮的值，用于唯一标识单选按钮
  );
```

收起 ^

The properties of the radio button group are

```

/**
  Create Csdn menu in the front page.
  在首页上创建一个DSCN菜单

  @param[in]    HiiHandle           The hii handle for the Uiapp driver.
  HII表示一个高层接口句柄，是UEFI用于用户处理界面资源的一种机制
  @param[in]    StartOpCodeHandle   The opcode handle to save the new opcode.
  StartOpCodeHandle是一个特殊的句柄，用于管理UEFI固件中的操作码（OpCode）

**/
VOID
UiCreateCsdnMenu(
  IN EFI_HII_HANDLE              HiiHandle,
  IN VOID                        *StartOpCodeHandle
  )
{
  HiiCreateActionOpCode ( //产生一个操作码
    StartOpCodeHandle, //指向已经创建的EFI_HII_HANDLE
    FRONT_PAGE_KEY_CSDN, //要定义的问题的ID
    STRING_TOKEN (STR_CSDN_STRING), //问题的提示信息为STR_CSDN_STRING
    STRING_TOKEN (STR_CSDN_STRING_HELP), //问题的帮助信息
    EFI_IFR_FLAG_CALLBACK, //用于定义问题的属性
    0
    );
}

/**
  Create CSDN menu in the front page with oneof opcode.

  @param[in]    HiiHandle           The hii handle for the Uiapp driver.
  @param[in]    StartOpCodeHandle   The opcode handle to save the new opcode.

**/
VOID
UiCreateCsdnRadioMenu (
  IN EFI_HII_HANDLE              HiiHandle,
  IN VOID                        *StartOpCodeHandle
  )
{
  CHAR8                       *CsdnText;
  VOID                        *OptionsOpCodeHandle;
  UINTN                        BufferSize;
  EFI_STATUS                   Status;

  CsdnText = AllocatePool (22); //分配EfiBootSerVicesData类型的缓冲区
  ASSERT (CsdnText != NULL);

  Status = gRT->GetVariable (//gRT指向运行时服务
                L"CSDNRadioButtonText", //检索一个名为CSDNRadioButtonText的EFI变量
                &gEfiIfrFrontPageGuid, //这个GUID是变量的拥有者
                NULL, //表示不需要获取变量的属性
                &BufferSize, //输出参数，变量值的大小
                CsdnText //输出参数，指向一个大小为BufferSize的缓冲区
                );

```

```
  if (!EFI_ERROR (Status)) {
    DEBUG((EFI_D_INFO,"[csdn] UiCreateCsdnRadioMenu = %a\n",CsdnText));
  }else{
    DEBUG((EFI_D_INFO,"[csdn] UiCreateCsdnRadioMenu = %r\n",Status));
  }

  OptionsOpCodeHandle = HiiAllocateOpCodeHandle (); //分配一个新的操作码
  ASSERT (OptionsOpCodeHandle != NULL);

  HiiCreateOneOfOptionOpCode ( //创建一个单选按钮组
        OptionsOpCodeHandle, //新创建的操作码
        STRING_TOKEN (STR_CSDN_STRING_RADIO_1), //按钮组的名称string ID
        EFI_IFR_OPTION_DEFAULT, //单选按钮的属性
        EFI_IFR_NUMERIC_SIZE_1, //单选按钮组的不同行为或外观
        0 //单选按钮组的值，唯一标识单选按钮组
        );

  HiiCreateOneOfOptionOpCode (
        OptionsOpCodeHandle,
        STRING_TOKEN (STR_CSDN_STRING_RADIO_2),
        EFI_IFR_OPTION_DEFAULT,
        EFI_IFR_NUMERIC_SIZE_1,
        1
        );

  HiiCreateOneOfOptionOpCode (
        OptionsOpCodeHandle,
        STRING_TOKEN (STR_CSDN_STRING_RADIO_3),
        EFI_IFR_OPTION_DEFAULT,
        EFI_IFR_NUMERIC_SIZE_1,
        2
        );

  HiiCreateOneOfOptionOpCode (
        OptionsOpCodeHandle,
        STRING_TOKEN (STR_CSDN_STRING_RADIO_4),
        EFI_IFR_OPTION_DEFAULT,
        EFI_IFR_NUMERIC_SIZE_1,
        3
        );

  HiiCreateOneOfOptionOpCode (
        OptionsOpCodeHandle,
        STRING_TOKEN (STR_CSDN_STRING_RADIO_5),
        EFI_IFR_OPTION_DEFAULT,
        EFI_IFR_NUMERIC_SIZE_1,
        4
        );

  HiiCreateOneOfOptionOpCode (
        OptionsOpCodeHandle,
        STRING_TOKEN (STR_CSDN_STRING_RADIO_6),
        EFI_IFR_OPTION_DEFAULT,
        EFI_IFR_NUMERIC_SIZE_1,
        5
        );

  HiiCreateOneOfOpCode (
    StartOpCodeHandle,
    FRONT_PAGE_KEY_CSDN_RADIIO,
    0,
    0,
    STRING_TOKEN (STR_CSDN_RADIO_STRING),
    STRING_TOKEN (STR_CSDN_RADIO_STRING_HELP),
    EFI_IFR_FLAG_CALLBACK,
    EFI_IFR_NUMERIC_SIZE_1,
    OptionsOpCodeHandle,
    NULL
    );
}
```


Use EFI_SET_VARIABLE SetVariable to set the value of a variable


```
typedef
EFI_STATUS
(EFIAPI *EFI_SET_VARIABLE)(
  IN  CHAR16                        *VariableName, //变量的名称
  IN  EFI_GUID                      *VendorGuid, //变量所有者的GUID
  IN  UINT32                        Attributes, //定义变量的属性
  IN  UINTN                         DataSize, //指定在Data参数中提供的缓冲区的大小
  IN  VOID                          *Data //一个指向viod类型的指针，指向包含要设置的数据的缓冲区
  );
```

```
// 定义 CallBack
EFI_STATUS
CsdnRadioChangeHandler (
  IN  EFI_IFR_TYPE_VALUE                    *Value
  )
{
  EFI_STATUS                    Status;

  CHAR8  *CsdnText = "Csdn Radio Button No 1";
```

```
  DEBUG((EFI_D_INFO,"[csdn] CsdnRadioChangeHandler value=%x\n",Value->u8));

  Status = gRT->SetVariable (
                  L"CSDNRadioButtonText",
                  &gEfiIfrFrontPageGuid,
                  EFI_VARIABLE_NON_VOLATILE | EFI_VARIABLE_BOOTSERVICE_ACCESS | EFI_VARIABLE_RUNTIME_ACCESS,
                  22,
                  CsdnText
                  );
  if (EFI_ERROR (Status)) {
    DEBUG((EFI_D_INFO,"[csdn] CsdnRadioChangeHandler status=%r\n",Status));
    return EFI_DEVICE_ERROR;
  }
  return EFI_SUCCESS;
}

//触发 CallBack
BOOLEAN
UiSupportLibCallbackHandler (
  IN  EFI_HII_HANDLE                      HiiHandle,
  IN  EFI_BROWSER_ACTION                  Action,
  IN  EFI_QUESTION_ID                     QuestionId,
  IN  UINT8                               Type,
  IN  EFI_IFR_TYPE_VALUE                  *Value,
  OUT EFI_BROWSER_ACTION_REQUEST          *ActionRequest,
  OUT EFI_STATUS                          *Status
  ){
 ...
 switch (QuestionId) {
    case FRONT_PAGE_KEY_CONTINUE:
      //
      // This is the continue - clear the screen and return an error to get out of FrontPage loop
      //
      *ActionRequest = EFI_BROWSER_ACTION_REQUEST_EXIT;
      break;
   ...

    //CallBack 触发入口
    case FRONT_PAGE_KEY_CSDN_RADIIO:
      *Status = CsdnRadioChangeHandler(Value);
      break;

    default:
      break;
    }

  }
```

收起 ^