EDK II Meta-data File Specification Document Change Notices (DCNs)

TABLE OF CONTENTS

INTRODUCTION
EDK II DOCUMENT CHANGE NOTICES

Last Updated: 24 March 2016

Proposed changes to EDK II Meta-Data File Specifications

Legal Information

MODULE INFORMATION FILE (INF)
DCNS FOR THE EDK II INF SPECIFICATION

(http://github.com/tianocore-docs/Docs)

Target Revision: 1.26

Target Release Date: February 26, 2016

1. Allow VOID* PCD entries to include Registry or C Format GUID values as well as byte array and string values.

Revision History

1. Revision number must change when there is a change to the specification that includes new content.
2. Changes considered bug fixes to the specification may use the same revision number, with an Errata tag. Errata tags are alpha characters.

Revision Number	Description	Date
1.25	- Revised WORKSPACE wording for updated build system that can handle packages located outside of the WORKSPACE directory tree (refer to the TianoCore.org/EDKII website for additional instructions on setting up a development environment).	January 2016
> 1.26	*> Updates:*	*> Target Date*
	> - Specification revision to 1.26	
	> - Allow Registry Format GUID values as well a C Format GUID values	

Individual DCNs

INF VERSION UPDATE
DOCUMENT CHANGE NOTICE

Change Document: EDK II Module Information (INF) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.25

Target Revision: 1.26

DCN Date: 27 Jan 2016

Summary

Changes the `INF_VERSION` value from `0x00010019` to `0x0001001A` or `1.26`

2.4 [Defines] Section

This is a required section.

The [Defines] section of EDK II INF files is used to define variable assignments that can be used in later build steps. The INF_VERSION of existing INF files does not need to be updated unless content in the file has been updated to match new content specified by this revision of the specification.

Architectural modifiers are not permitted in the [Defines] section.

The parsing utilities process any local symbol assignments defined in this section. The EDK II parsing utilities will use some of this section's information for generating AutoGen.c and AutoGen.h files. Note that the sections are processed in the order listed in the INF file, and later assignments of these local symbols override previous assignments.

`[Defines]`

The format for entries in this section is:

```
Name = Value
```

The following is an example of a driver's `[Defines]` section.

```
[Defines]
~~INF_VERSION        = 0x00010019~~
> INF_VERSION        = 0x0001001A
  BASE_NAME          = DxeIpl
  FILE_GUID          = 86D70125-BAA3-4296-A62F-602BEBBB9081
  VERSION_STRING     = 1.0
  MODULE_TYPE        = PEIM
  ENTRY_POINT        = PeimInitializeDxeIpl
  MODULE_UNI_FILE    = DxeIpl.uni
```

The following is an example of a library's [Defines] section.

```
[Defines]
```

```
~~  INF_VERSION      = 1.26~~
>  INF_VERSION      = 1.26
  BASE_NAME         = BaseMemoryLib
  FILE_GUID         = fd44e603-002a-4b29-9f5f-529e815b6165
  MODULE_TYPE       = BASE
  VERSION_STRING    = 1.0
  LIBRARY_CLASS     = BaseMemoryLib
```

3.4 [Defines] Section

This is a required section.

Summary

This describes the required [Defines] section used in EDK II INF files. This file is created during installation of a UEFI distribution package or by the developer and is an input to the new build tool parsing utilities. Elements may appear in any order within this section.

~~The version for this specification is "0x00010019" and new versions of this specification must increment the minor (0019) portion of the specification code for backward compatible changes, and increment the major number for non-backward compatible specification changes. This value may also be specified as a decimal value, 1.25.~~

> The version for this specification is "0x0001001A" and new versions of this specification must increment the minor (001A) portion of the specification code for backward compatible changes, and increment the major number for non-backward compatible specification changes. This value may also be specified as a decimal value, 1.26.

The [Defines] section assigns values to the symbols that describe the component. Some items are emitted to the output makefile, others are used to create filenames during the build. Some symbols are emitted to the generated C files.

~~The FILE_GUID is required for all EDK II modules. This GUID is used to build the FW volume file list used by build tools to generate the final firmware volume, as well as processed in some SMM, PEI or DXE DEPEX statements.~~

> The FILE_GUID is required for all EDK II modules. This GUID is used to build the FW volume file list used by build tools to generate the final firmware volume, as well as processed in some SMM, PEI or DXE DEPEX statements.

~~All new EDK II INF files must include one of the following statements: INF_VERSION = 0x00010019 or INF_VERSION = 1.25 in this section, where the number varies according to the release of this specification. It is a HexVersion type, where the 0x0001 is the major number, and the 0019 is the minor number. This version of the specification provides full backward compatibility to all previous versions. This means that tools that process this version of the specification can also process earlier versions of EDK II INF files.~~

All new EDK II INF files must include one of the following statements: `INF_VERSION = 0x0001001A` or `INF_VERSION = 1.26` in this section, where the number varies according to the release of this specification. It is a `HexVersion` type, where the `0x0001` is the major number, and the `001A` is the minor number. This version of the specification provides full backward compatibility to all previous versions. This means that tools that process this version of the specification can also process earlier versions of EDK II INF files.

Parameters

Filename	
	Filenames listed in the [Defines] section must be relative to the directory the INF file is in. Use of "..", "." and "../" in the directory path is not permitted. Use of an absolute path is not permitted. The file name specified in the MODULE_UNI_FILE entry must be a Unicode file with an extension of .uni, .UNI or .Uni.
MODULE_TYPE	
	Drivers and applications are not allowed to have a `MODULE_TYPE` of "`BASE`". Only libraries are permitted to a have a `MODULE_TYPE` of "`BASE`". A INF file can be used to specify other binary files types, such as logo images or legacy16 option ROMs. The `USER_DEFINED` module type must be used in all cases where the module type is not a member of `<Edk2ModuleType>`.
INF_VERSION	
	~~For new INF files, the version value must be set to `0x00010019`. Tools that process this version of the INF file can successfully process earlier versions of the INF file (this is a backward compatible update). There is no requirement to change the value in existing INF files if no other content changes. This may also be specified as decimal value, `1.25`.~~
	> For new INF files, the version value must be set to `0x00010019`. Tools that process this version of the INF file can successfully process earlier versions of the INF file (this is a backward compatible update). There is no requirement to change the value in existing INF files if no other content changes. This may also be specified as decimal value, `1.25`.
EDK_RELEASE_VERSION	
	This optional value may be set to the major/minor number of the EDK II release required for modules to function correctly.

~~Example~~

Example (EDK II Driver)

```
[Defines]
~~  INF_VERSION              = 1.25~~
>  INF_VERSION               = 1.26
  BASE_NAME                  = PlatformAcpiTable
  FILE_GUID                  = 7E374E25-8E01-4FEE-87F2-390C23C606CD
  MODULE_TYPE                = DXE_DRIVER
  VERSION_STRING             = 1.0
  EDK_RELEASE_VERSION        = 0x00020000
  UEFI_SPECIFICATION_VERSION = 0x00020014
```

Example (UEFI Driver)

```
[Defines]
  ~~INF_VERSION = 0x00010019~~
> INF_VERSION = 0x0001001A
  BASE_NAME = Abc
  FILE_GUID = DA87D340-15C0-4824-9BF3-D52286674BEF
  MODULE_TYPE = UEFI_DRIVER
  VERSION_STRING = 1.0
```

```
  ENTRY_POINT = AbcDriverEntryPoint
  UNLOAD_IMAGE = AbcUnload
```

Example (EDK II Library)

```
[Defines]
~~  INF_VERSION   = 1.25~~
>  INF_VERSION    = 1.26
  BASE_NAME       = LzmaCustomDecompressLib
  FILE_GUID       = 22f8406f-43ee-492f-82f5-4e8a1a58e6d2
  MODULE_TYPE     = BASE
  VERSION_STRING  = 1.0
  LIBRARY_CLASS   = CustomDecompressLib
```

REGISTRY FORMAT GUIDS DOCUMENT CHANGE NOTICE

Change Document: EDK II Module Information (INF) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.25

Target Revision: 1.26

DCN Date: 08 Feb 2016

Summary

This DCN adds support for specifying VOID* PCD values as a Registry Format GUID as well as a C Format GUID.

3.8 PCD Sections

Prototype

```
<Value>         ::= if (pcddatumtype == "BOOLEAN"):
                      <Boolean>
                    elif (pcddatumtype == "UINT8"):
                      <NumValUint8>
                    elif (pcddatumtype == "UINT16"):
                      <NumValUint16>
                    elif (pcddatumtype == "UINT32"):
                      <NumValUint32>
                    elif (pcddatumtype == "UINT64"):
                      <NumValUint64>
                    else:
                      <StringVal>

~~<stringVal>   ::= {<StringType>} {<CArray>}~~
> <stringVal>   ::= {<StringType>} {<CArray>} {<GuidValue>}
>
> <GuidValue>   ::= {<CFormatGuid>} {<RegFormatGuid>}

<StringType>    ::= {<UnicodeString>} {<CString>}

<FFE>           ::= <FS> <FeatureFlagExpress>
```

PACKAGE DECLARATION FILE (DEC) DCNS FOR THE EDK II DEC SPECIFICATION

(http://github.com/tianocore-docs/Docs)

Target Revision: 1.26

Target Release Date: February 26, 2016

1. Add support for UDP SingleGuid, SingleProtocol and SinglePpi attributes
2. Allow GUID, Protocol and PPI declarations to use either Registry or C Format GUID values

Revision History

1. Revision number must change when there is a change to the specification that includes new content.
2. Changes considered bug fixes to the specification may use the same revision number, with an Errata tag. Errata tags are alpha characters.

Revision Number	Description	Date
1.25	Updates:	January 2016
	- Specification revision to 1.25	
	- Revised WORKSPACE wording for updated build system that can handle packages located outside of the WORKSPACE directory tree (refer to the TianoCore.org/EDKII website for additional instructions on setting up a development environment).	
> 1.26	*> Updates:*	*> Target Date*
	> - Specification revision to 1.26	
	> - Add doxygen tags for GUIDs, Protocols and PPIs in comment blocks	
	> - Allow Registry Format GUID values as well a C Format GUID values	
	> - Specify Private Content only used by modules within a package	

Individual DCNs

DEC VERSION UPDATE
DOCUMENT CHANGE NOTICE

Change Document: EDK II Package Declaration (DEC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.25

Target Revision: 1.26

DCN Date: 27 Jan 2016

Summary

Changes the `DEC_SPECIFICATION` value from `0x00010019` to `0x0001001A` or `1.26`

2.4 [Defines] Usage

The DEC_SPECIFICATION of existing DEC files does not need to be updated unless content in the file has been updated to match new content specified by this revision of the specification. Additionally, the package's version major number may change. Minor changes require incrementing the package's version minor number. The `PACKAGE_UNI_FILE` entry points to a Unicode file containing localization strings. The use of the `#include` statement in this file is prohibited. The file path (if present) is relative to the directory containing the DEC file.

The parsing utilities process any local symbol assignments defined in this section. Note that the sections are processed in the order listed in the DEC file, and later assignments of these local symbols override previous assignments.

This section will use only one section header:

```
[Defines]
```

The format for entries in this section is:

```
Name = Value
```

The following is an example of this section.

```
[Defines]
~~DEC_SPECIFICATION = 0x00010019~~
>  DEC_SPECIFICATION = 0x0001001A
  PACKAGE_NAME = MdePkg
  PACKAGE_GUID = 1E73767F-8F52-4603-AEB4-F29B510B6766
  PACKAGE_VERSION = 1.02
  PACKAGE_UNI_FILE = MdePkg.uni
```

3.4 [Defines] Sections

Summary This describes the `[Defines]` section, which is required in all DEC files. This file is

created during installation of a UEFI distribution package or by the developer and is an input to the EDK II build tool parsing utilities. Elements may appear in any order within this section.

~~The code for this specification is "00010019" and new versions of this specification must increment the minor (0019) portion of the specification code. This value may also be specified as a decimal value, 1.25.~~

> The code for this specification is `0x0001001A` and new versions of this specification must increment the minor (`001A`) portion of the specification code. This value may also be specified as a decimal value, `1.26`.

Existing DEC files are not required to update the `DEC_SPECIFICATION` version value. This value may be used by tools to identify any new functionality introduced by this specification version.

Parameters

UiNameType	
	The `PACKAGE_NAME` value may be used for creating directories.
DecimalVersion	
	This is a decimal number, and if not specified is assumed to be 0. Alpha characters are not permitted.
SpecVer	
	For new DEC files, the version value must be set to `0x0001001A`. Tools that process this version of the DEC file can successfully process earlier versions of the DEC file (this is a backward compatible update). There is no requirement to change the value in existing DEC files if no other content changes. This may also be specified as decimal value, `1.26`.
Filename	
	Filenames listed in the `[Defines]` section must be relative to the directory the DEC file is in. Use of "..", "." and "../" in the directory path is not permitted. Use of an absolute path is not permitted. The file name specified in the `PACKAGE_UNI_FILE` entry must be a Unicode file with an extension of .uni, .UNI or .Uni.

~~**Example**~~

Example 1

```
[DEFINES]
  ~~DEC_SPECIFICATION = 0x00010019~~
  > DEC_SPECIFICATION = 0x0001001A
  PACKAGE_NAME = MdePkg
  ~~PACKAGE_GUID = 5e0e9358-46b6-4ae2-8218-4ab8b9bbdcec~~
  ~~PACKAGE_VERSION = 0.3~~
  > PACKAGE_GUID = 1E73767F-8F52-4603-AEB4-F29B510B6766
  > PACKAGE_VERSION = 1.06
  PACKAGE_UNI_FILE = MdePkg.uni
```

Example 2

```
> [DEFINES]
  > DEC_SPECIFICATION = 1.26
  > PACKAGE_NAME = IntelFspPkg
  > PACKAGE_GUID = 444C6CDF-55BD-4744-8F74-AE98B003B955
  > PACKAGE_VERSION = 0.1
```

SINGLEGUID
DOCUMENT CHANGE NOTICE

Change Document: EDK II Package Declaration (DEC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.25

Target Revision: 1.26

DCN Date: 27 Jan 2016

Summary

1. Allow use of Registry Format GUID to declarations
2. Adding new reserved comment, `@SingleGuid <GuidType>`

2.6 [Guids] Usage

This is an optional section.

This section is used to define the GUID Value for Guid C Names.

This section uses one of the following section definitions:

```
[Guids] [Guids.IA32] [Guids.X64] [Guids.IPF] [Guids.EBC]
[Guids.common] [Guids.IA32, Guids.X64] [Guids.X64, Guids.IPF]
```

Format for the entries in this section is two fields with an equal "=" character separating the fields as shown below.

```
> #Comment Block
~~GuidCName = {C Format Guid Value} # Comment~~
> GuidCName = Guid Value # Comment
```

~~The Comment section can be used to identify the list of supported module types.~~

A comment block preceding the entry must be used to provide detailed information, including the name of the package relative header file for the GUID. An optional comment following an entry may be used to provide simple information.

3.6 [Guids] Sections

Prototype

```
<Guids>         ::= "[Guids" [<com_attribs>] "]" <EOL>
                    <GuidEntries>*

<com_attribs>    ::= {".common"} {<attribs>}

<attribs>        ::= <attrs> ["," <TS> "Guids" <attrs>]*
```

```
  <attrs>          ::= "." <arch>

  <GuidEntries>    ::= [<GuidComment>]
~~                      <TS> <CName> <Eq> <CFormatGuid>~~
>                      <TS> <CName> <Eq> <GuidValue>
                       {<CommentBlock>} {<EOL>}

  <GuidComment>    ::= [<Description>]
>                     [<SingleGuid>]*
                       <TS> "##" <TS> <GuidHeaderFile> <EOL>

>  <SingleGuid>      ::= <TS> "#" <TS> "@SingleGuid" <GuidUseType> <
>  <GuidUseType>     ::= <TS> ":" <TS> <GuidType>
>  <GuidType>        ::= {"Event"} {"File"} {"FV"} {"GUID"} {"HII"} 
>                        {"SystemTable"} {"TokenSpaceGuid"} {"Variab
>                        {"UNDEFINED"}

  ~~<GuidValue>      ::= <CFormatGUID>~~
>  <GuidValue>      ::= {<CFormatGUID>} {<RegistryFormatGUID>}

  <GuidHeaderFile> ::= <PATH> <Word> ".h"

  <Description>    ::= <TS> "##" <TS> <AsciiString> <EOL>
                       [<TS> "#" <TS> <AsciiString> <EOL>]*

  <CommentBlock>   ::= <TS> "##" <TS> <ModuleTypeList>
                       {<Comment>} {<EOL>}
```

Example

```
##############################################################################
#
# Global Guid Definition section - list of Global Guid C Name
# Data Structures that are provided by
# this package.
#
##############################################################################
~~[Guids.common]
gPcdHobGuid = { 0x582E7CA1, 0x68CD, 0x4D44, \
{ 0xB4, 0x3B, 0xF2, 0x98, 0xED, 0x58, 0x7B, 0xA6 }}
gEfiWinNtPassThroughGuid = { 0xCC664EB8, 0x3C24, 0x4086, \
{ 0xB6, 0xF6, 0x34, 0xE8, 0x56, 0xBC, 0xE3, 0x6E }}
gEfiWinNtCPUSpeedGuid = { 0xD4F29055, 0xE1FB, 0x11D4, \
{ 0xBD, 0x0D, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiWinNtCPUModelGuid = { 0xBEE9B6CE, 0x2F8A, 0x11D4, \
{ 0xBD, 0x0D, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiWinNtMemoryGuid = { 0x99042912, 0x122A, 0x11D4, \
{ 0xBD, 0x0D, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiWinNtConsoleGuid = { 0xBA73672C, 0xA5D3, 0x11D4, \
{ 0xBD, 0x00, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiWinNtUgaGuid = { 0xAB248E99, 0xABE1, 0x11D4, \
{ 0xBD, 0x0D, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiWinNtGopGuid = { 0x4e11e955, 0xccca, 0x11d4, \
```

```
{ 0xbd, 0x0d, 0x00, 0x80, 0xc7, 0x3c, 0x88, 0x81 }}
gEfiWinNtSerialPortGuid = { 0x0C95A93D, 0xA006, 0x11D4, \
{ 0xBC, 0xFA, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiWinNtFileSystemGuid = { 0x0C95A935, 0xA006, 0x11D4, \
{ 0xBC, 0xFA, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiWinNtPhysicalDisksGuid = { 0x0C95A92F, 0xA006, 0x11D4, \
{ 0xBC, 0xFA, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiWinNtVirtualDisksGuid = { 0x0C95A928, 0xA006, 0x11D4, \
{ 0xBC, 0xFA, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}
gEfiEdkNt32PkgTokenSpaceGuid = { 0x0D79A645, 0x1D91, 0x40a6, \
{ 0xA8, 0x1F, 0x61, 0xE6, 0x98, 0x2B, 0x32, 0xB4 }}~~

>[Guids]
>  ## MdeModule package token space guid
>  # Include/Guid/MdeModulePkgTokenSpace.h
>  gEfiMdeModulePkgTokenSpaceGuid     = { 0xA1AFF049, 0xFDEB, \
>    0x442a, { 0xB3, 0x20, 0x13, 0xAB, 0x4C, 0xB7, 0x2B, 0xBC }}
>
>  ## Hob guid for Pcd DataBase
>  #  Include/Guid/PcdDataBaseHobGuid.h
>  gPcdDataBaseHobGuid                = { 0xEA296D92, 0x0B69, \
>    0x423C, { 0x8C, 0x28, 0x33, 0xB4, 0xE0, 0xA9, 0x12, 0x68 }}
>
>  ## Include/Guid/AcpiS3Context.h
>  gEfiAcpiS3ContextGuid              = { 0xef98d3a, 0x3e33, \
>    0x497a, { 0xa4, 0x1, 0x77, 0xbe, 0x3e, 0xb7, 0x4f, 0x38 }}
>
>  ## Include/Guid/BootScriptExecutorVariable.h
>  # @SingleGuid : GUID
>  gEfiBootScriptExecutorVariableGuid = { 0x3079818c, 0x46d4, \
>    0x4a73, { 0xae, 0xf3, 0xe3, 0xe4, 0x6c, 0xf1, 0xee, 0xdb }}
>
>  ## Include/Guid/BootScriptExecutorVariable.h
>  # @SingleGuid : GUID
>  gEfiBootScriptExecutorContextGuid  = { 0x79cb58c4, 0xac51, \
>    0x442f, { 0xaf, 0xd7, 0x98, 0xe4, 0x7d, 0x2e, 0x99, 0x8 }}
>
>  ## Include/Guid/Ip4Config2Hii.h
>  gIp4Config2NvDataGuid              = 9b942747-154e-4d29-a436-bf71(
```

SINGLEPROTOCOL
DOCUMENT CHANGE NOTICE

Change Document: EDK II Package Declaration (DEC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.25

Target Revision: 1.26

DCN Date: 27 Jan 2016

Summary

1. Allow use of Registry Format GUID to declarations
2. Adding new reserved comment, `@SingleProtocol`

2.7 [Protocols] Usage

This is an optional section.

This section is used to define the GUID Value for Protocol C Names.

This section uses one of the following section definitions:

`[Protocols] [Protocols.IA32] [Protocols.X64] [Protocols.IPF] [Protocols.EBC] [Protocols.common]`

Format for the entries in this section is two fields with an equal "=" character separating the fields as shown below.

```
>
> \# Comment Block
>  ProtocolCName = Guid Value # Comment
~~ProtocolCName = {C Format Guid Value} # Comment~~
```

~~The Comment section can be used to identify the list of supported module types.~~

A comment block preceding the entry must be used to provide detailed information, including the name of the package relative header file for the Protocol. An optional comment following an entry may be used to provide simple information.

3.7 [Protocols] Sections

Prototype*

```
<ProtocolEntries>   ::= [<ProtocolComment>]
~~                        <TS> <CName> <Eq> <CFormatGUID>~~
~~                        {<CommentBlock>} {<EOL>}~~
>                        <TS> <CName> <Eq> <GuidValue> [<CommentBlock
```

```
> <GuidValue>         ::= {<CFormatGuid>} {<RegistryFormatGuid>}
```

```
<ProtocolComment>   ::= [<Description>]
>                       [<SingleProtocol>]
                        <ProtoHdrFile>

> <SingleProtocol>  ::= <TS> "@SingleProtocol" <EOL>
>
<ProtoHdrFile>      ::= <TS> "##" <TS> <PATH> <Word> ".h"
```

Example

```
##############################################################################
#
# Global Protocols Definition section - list of Global Protocols C Na
# Data Structures that are provided by this package.
#
##############################################################################
[Protocols]
  ~~gEfiWinNtThunkProtocolGuid = { 0x58C518B1, 0x76F3, 0x11D4, \~~
    ~~{ 0xBC, 0xEA, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}~~
  ~~gEfiWinNtIoProtocolGuid = { 0x96EB4AD6, 0xA32A, 0x11D4, \~~
    ~~{ 0xBC, 0xFD, 0x00, 0x80, 0xC7, 0x3C, 0x88, 0x81 }}~~

>  ## Print protocol defines basic print functions to print the forma
>  # ascii string.
>  # Include/Protocol/Print2.h
>  gEfiPrint2ProtocolGuid          = { 0xf05976ef, 0x83f1, 0x4f3d, \
>    { 0x86, 0x19, 0xf7, 0x59, 0x5d, 0x41, 0xe5, 0x38 } }
>
>  ## This protocol defines the generic memory test interfaces in Dx
>  # @SingleProtocol
>  # Include/Protocol/GenericMemoryTest.h
>  gEfiGenericMemTestProtocolGuid = { 0x309DE7F1, 0x7F5E, 0x4ACE, \
>    { 0xB4, 0x9C, 0x53, 0x1B, 0xE5, 0xAA, 0x95, 0xEF }}
>
>  ## Fault Tolerant Write protocol provides boot-time service to do
>  # capability for block devices.
>  # @SingleProtocol
>  #  Include/Protocol/FaultTolerantWrite.h
>  gEfiFaultTolerantWriteProtocolGuid = { 0x3EBD9E82, 0x2C78, 0x4DE6,
>  { 0x97, 0x86, 0x8D, 0x4B, 0xFC, 0xB7, 0xC8, 0x81 }}
>
>  ## This protocol provides boot-time service to do fault tolerant w
>  # block devices in SMM environment.
>  #  Include/Protocol/SmmFaultTolerantWrite.h
>  gEfiSmmFaultTolerantWriteProtocolGuid = 3868fc3b-7e45-43a7-906c-4b
>
```

SINGLEPPI
DOCUMENT CHANGE NOTICE

Change Document: EDK II Package Declaration (DEC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.25

Target Revision: 1.26

DCN Date: 27 Jan 2016

Summary

1. Allow use of Registry Format GUID to declarations
2. Adding new reserved comment, `@SinglePpi`

2.8 [Ppis] Usage

This is an optional section.

This section is used to define the GUID Value for PPI C Names.

This section uses one of the following section definitions:

`[Ppis] [Ppis.IA32] [Ppis.X64] [Ppis.IPF] [Ppis.EBC] [Ppis.common]`

Format for the entries in this section is two fields with an equal "=" character separating the fields as shown below.

```
> # Comment Block
>  PpiCName = Guid Value # Comment
~~PpiCName = {C Format Guid Value} # Comment~~
~~PpiCName = RegistryFormatGUID # Comment~~
```

~~The Comment section can be used to identify the list of supported module types.~~

A comment block preceding the entry must be used to provide detailed information, including the name of the package-relative header file for the PPI. An optional comment following an entry may be used to provide simple information.

3.8 [Ppis] Sections

Prototype

```
<PpiEntries>           ::= [<PpiComment>]
~~                         <TS> <CName> <Eq> <CFormatGUID>~~
~~                         {<CommentBlock>} {<EOL>}~~
>                          <TS> <CName> <Eq> <GuidValue> [<CommentBlo

> <GuidValue>          ::= {<CFormatGuid>} {<RegistryFormatGuid>}
```

```
<PpiComment>          ::= [<Description>]
>                         [<SinglePpi>]
                          <TS> "##" <TS> <PpiHdrFile> <EOL>

> <SinglePpi>         ::= <TS> "@SinglePpi" <EOL>
>
<PpiHdrFile>          ::=  <PATH> <Word> ".h"
```

Example

```
##############################################################################
#
# Global Ppis Definition section - list of Global Ppis C Name
# Data Structures that are provided by this package.
#
##############################################################################
~~[Ppis.common]~~
~~gPeiNtThunkPpiGuid = { 0x98C281E5, 0xF906, 0x43DD, \~~
~~{ 0xA9, 0x2B, 0xB0, 0x03, 0xBF, 0x27, 0x65, 0xDA }}~~
~~gNtPeiLoadFilePpiGuid = { 0xFD0C65EB, 0x0405, 0x4CD2, \~~
~~{ 0x8A, 0xEE, 0xF4, 0x00, 0xEF, 0x13, 0xBA, 0xC2 }}~~
~~gNtFwhPpiGuid = { 0x4E76928F, 0x50AD, 0x4334, \~~
~~{ 0xB0, 0x6B, 0xA8, 0x42, 0x13, 0x10, 0x8A, 0x57 }}~~
~~gPeiNtAutoScanPpiGuid = { 0x0DCE384D, 0x007C, 0x4BA5, \~~
~~{ 0x94, 0xBD, 0x0F, 0x6E, 0xB6, 0x4D, 0x2A, 0xA9 }}~~

> [Ppis]
>   ## Include/Ppi/PostBootScriptTable.h
>   gPeiPostScriptTablePpiGuid    =  { 0x88c9d306, 0x900, 0x4eb5, \
>     { 0x82, 0x60, 0x3e, 0x2d, 0xbe, 0xda, 0x1f, 0x89}}
>
>   ## Include/Ppi/SerialPortPei.h
>   # @SinglePpi
>   gPeiSerialPortPpiGuid         =  { 0x490e9d85, 0x8aef, 0x4193, \
>     { 0x8e, 0x56, 0xf7, 0x34, 0xa9, 0xff, 0xac, 0x8b}}
>
>   ## Include/Ppi/UfsHostController.h
>   gEdkiiPeiUfsHostControllerPpiGuid  =  dc54b283-1a77-4cd6-83bb-fd
>
```

PRIVATE SETTINGS
DOCUMENT CHANGE NOTICE

Change Document: EDK II Package Declaration (DEC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.25

Target Revision: 1.26

DCN Date: 27 Jan 2016

Summary

Add new syntax to the DEC file for specifying information that can only be used by modules within the package. When modules outside the packages attempt to use this content, the EDK II build system must break with an error regarding content not found.

The four sections, Includes, Ppis, Guids and Protocols headers will be modified with a keyword, Private following the architecture modifier. If Private is not present, then the content is usable by modules outside the package.

2.5 [Includes] Usage

Also included in this section are the directories containing headers that may be required for individual EDK II module types. Refer to Appendix, "*EDK II Module Types*", for a list of the valid types.

The section tag modifier, Private, is used to restrict the EDK II build system by preventing references to content in these sections from being used by modules outside of the package. It is not permissible to mix section tags without the Private attribute with section tags with the Private attribute.

For example, `[Includes.common, Includes.IA32.Private]` is prohibited.

Refer to the `[Includes]` definition later in this document for a complete description of this section and its contents.

The [Includes] section uses one of the following section definitions:

```
~~[Includes.common] [Includes.IA32] [Includes.X64] [Includes.IPF]~~
~~[includes.EBC] [Includes]~~
> [Includes.common] [Includes.common.Private] [Includes.IA32]
> [Includes.IA32.Private] [Includes.X64] [Includes.X64.Private]
> [Includes.IPF] [Includes.IPF.Private] [includes.EBC]
> [includes.EBC.Private] [Includes]
```

2.6 [Guids] Usage

This is an optional section.

This section is used to define the GUID Value for Guid C Names.

The section tag modifier, Private, is used to restrict the EDK II build system by preventing references to content in these sections from being used by modules outside of the package. It is not permissible to mix section tags without the Private attribute with section tags with the Private attribute.

This section use ones of the following section definitions:

```
~~[Guids] [Guids.IA32] [Guids.X64] [Guids.IPF]~~
~~[Guids.EBC] [Guids.common] [Guids.IA32, Guids.X64] [Guids.X64, Gui

> [Guids] [Guids.common] [Guids.common.Private] [Guids.IA32]
> [Guids.IA32.Private] [Guids.X64] [Guids.X64.Private]
> [Guids.IPF] [Guids.IPF.Private] [Guids.EBC]
> [Guids.EBC.Private]
```

Architectural sections may also be combined, as in:

```
[Guids.IA32, Guids.X64]
[Guids.IA32.Private, Guids.X64.Private]
```

2.7 [Protocols] Usage

This is an optional section.

This section is used to define the GUID Value for Protocol C Names.

The section tag modifier, Private, is used to restrict the EDK II build system by preventing references to content in these sections from being used by modules outside of the package. It is not permissible to mix section tags without the Private attribute with section tags with the Private attribute.

This section use ones of the following section definitions:

```
~~[Protocols] [Protocols.IA32] [Protocols.X64] [Protocols.IPF]~~
~~[Protocols.EBC] [Protocols.common]~~

> [Protocols] [Protocols.common] [Protocols.common.Private] [Protocol
> [Protocols.IA32.Private] [Protocols.X64] [Protocols.X64.Private]
> [Protocols.IPF] [Protocols.IPF.Private] [Protocols.EBC]
> [Protocols.EBC.Private]
```

Architectural sections may also be combined, as in:

```
[Protocols.IA32, Protocols.X64]
[Protocols.IA32.Private, Protocols.X64.Private]
```

2.8 [Ppis] Usage

This is an optional section.

This section is used to define the GUID Value for PPI C Names.

> The section tag modifier, Private, is used to restrict the EDK II build system by preventing references to content in these sections from being used by modules outside of the package. It is not permissible to mix section tags without the Private attribute with section tags with the Private attribute.

This section use ones of the following section definitions:

```
~~[Ppis] [Ppis.IA32] [Ppis.X64] [Ppis.IPF] [Ppis.EBC] [Ppis.common]~~
> [Ppis] [Ppis.common] [Ppis.common.Private] [Ppis.IA32] [Ppis.IA32.P
> [Ppis.X64] [Ppis.X64.Private] [Ppis.IPF] [Ppis.IPF.Private] [Ppis.P
> [Ppis.EBC.Private]
```

> Architectural sections may also be combined, as in:
>
> ```
> [Ppis.IA32, Ppis.X64]
> [Ppis.IA32.Private, Ppis.X64.Private]
> ```

3.5 [Includes] Sections

Prototype

```
<Include>       ::= "[Includes" [<com_attribs>] "]" <EOL>
                    <IncEntries>*

~~ <com_attribs>  ::= {".common"} {<attribs>}~~
> <com_attribs>  ::= {<Public>} {<Hidden>}

> <Public>       ::= {".common"} {<attribs>}
>
> <Hidden>       ::= {".common.Private"} {<hattribs>}
>
<attribs>       ::= <attrs> ["," <TS> "Includes" <attrs>]*

> <hattribs>     ::= <hattrs> ["," <TS> "includes" <hattrs>]*
>
<attrs>         ::= "." <arch>
>
> <hattrs>        ::= "." <arch> ".Private"
```

3.6 [Guids] Sections

Prototype

```
<Guids>          ::= "[Guids" [<com_attribs>] "]" <EOL>
                     <GuidEntries>*

~~<com_attribs>   ::= {".common"} {<attribs>}~~
> <com_attribs>   ::= {<Public>} {<Hidden>}
>
```

```
> <Public>         ::= {".common"} {<attribs>}
>
> <Hidden>        ::= {".common.Private"} {<hattribs>}

<attribs>        ::= <attrs> ["," <TS> "Guids" <attrs>]*

<attrs>          ::= "." <arch>
>
> <hattribs>      ::= <hattrs> ["," <TS> "Guids" <hattrs>]*
>
> <hattrs>         ::= "." <arch> ".Private"

<GuidEntries>    ::= [<GuidComment>]
                     <TS> <CName> <Eq> <CFormatGUID> {<CommentBlock>}
```

3.7 [PPIs] Sections

Prototype

```
<Ppis>           ::= "[Ppis" [<com_attribs>] "]" <EOL>
                     <PpiEntries>*

~~<com_attribs>    ::= {".common"} {<attribs>}~~
> <com_attribs>    ::= {<Public>} {<Hidden>}
>
> <Public>         ::= {".common"} {<attribs>}
>
> <Hidden>        ::= {".common.Private"} {<hattribs>}

<attribs>        ::= <attrs> ["," <TS> "Ppis" <attrs>]*

<attrs>          ::= "." <arch>
>
> <hattribs>      ::= <hattrs> ["," <TS> "Ppis" <hattrs>]*
>
> <hattrs>         ::= "." <arch> ".Private"

<PpiEntries>     ::= [<PpiComment>]
                     <TS> <CName> <Eq> <CFormatGUID> {<CommentBlock>}
```

3.7 [Protocols] Sections

Prototype

```
<Protocols>      ::= "[Protocols" [<com_attribs>] "]" <EOL>
                     <ProtocolEntries>*

~~<com_attribs>    ::= {".common"} {<attribs>}~~
> <com_attribs>    ::= {<Public>} {<Hidden>}
>
> <Public>         ::= {".common"} {<attribs>}
>
```

```
> <Hidden>       ::= {".common.Private"} {<hattribs>}

<attribs>       ::= <attrs> ["," <TS> "Protocols" <attrs>]*

<attrs>         ::= "." <arch>
>
> <hattribs>     ::= <hattrs> ["," <TS> "Protocols" <hattrs>]*
>
> <hattrs>        ::= "." <arch> ".Private"

<ProtocolEntries> ::= [<ProtocolComment>]
                      <TS> <CName> <Eq> <CFormatGUID> {<CommentBlock>
```

PLATFORM DESCRIPTION FILE (DSC) DCNS FOR THE EDK II DSC SPECIFICATION

(http://github.com/tianocore-docs/Docs)

Target Revision: 1.27

Target Release Date: February 26, 2016

1. Add support for UDP SingleGuid, SingleProtocol and SinglePpi attributes
2. Allow GUID, Protocol and PPI declarations to use either Registry or C Format GUID values

Revision History

1. Revision number must change when there is a change to the specification that includes new content.
2. Changes considered bug fixes to the specification may use the same revision number, with an Errata tag. Errata tags are alpha characters.

Revision Number	Description	Date
1.26	Updates:	January 2016
	- Specification revision to 1.26	
	- Update the DSC_SPECIFICATION version to 0x0001001A	
	- Revised WORKSPACE wording for updated build system that can handle packages located outside of the WORKSPACE directory tree (refer to the TianoCore.org/EDKII website for additional instructions on setting up a development environment).	
	- Added new system environment variables used by the build system.	
> 1.27	*> Updates:*	*> Target Date*
	> - Specification revision to 1.26	
	> - Allow Registry Format GUID values as well a C Format GUID values	
	> - Specify Private Content only used by modules within a package	

Individual DCNs

DSC VERSION UPDATE
DOCUMENT CHANGE NOTICE

Change Document: EDK II Flash Definition (FDF) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 01 Feb 2016

Summary

Changes the `DSC_SPECIFICATION` value from `0x0001001A` to `0x0001001B` or `1.27`

2 DSC OVERVIEW

~~DSC files that use any new features must use the new DSC_SPECIFICATION = 0x0001001A in the [Defines] section. Older DSC files that do not use any of these features do not need to update the DSC_SPECIFICATION value. ~~

> DSC files that use any new features must use the new DSC_SPECIFICATION = 0x0001001B in the [Defines] section. Older DSC files that do not use any of these features do not need to update the DSC_SPECIFICATION value.

2.3 [Defines] Section

Table 6. EDK II [Defines] Section Elements

Typical Tag Names	Required/Optional	Value	Notes
~~`DSC_SPECIFICATION`~~	~~Required~~	~~0x0001001A or 1.26~~	~~This entry is required for all EDK II DSC files. The value, 0x0001001A matches the 1.26 version of this specification. Build tools must continue to support DSC files that correspond to earlier versions of the document until such time as earlier versions are no longer in use. In order to maintain backward compatibility, this value must only be updated in existing DSC files if other content in the file is updated. This value may also be specified as decimal value, i.e., 1.26.~~
> `DSC_SPECIFICATION`	> Required	> 0x0001001B or 1.27	> This entry is required for all EDK II DSC files. The value, 0x0001001B matches the 1.27 version of this specification. Build tools must continue to support DSC files that correspond to earlier versions of the document until such time as earlier versions are no longer in use. In order to maintain backward compatibility, this value must only be updated in existing DSC files if other content in the file is updated. This value may also be specified as decimal value, i.e., 1.27.

3.5 [Defines] Section

Summary

Note: *Assignments of variables in other sections take precedence over global assignments.*

This section describes the defines section content in the DSC files. This file can be created by a developer and is an input to the EDK II build tool parsing utilities. Elements may appear in any order

within this section.

~~This revision of specification does not add new features. New EDK II DSC files must include the statement: DSC_SPECIFICATION = 0x0001001A in this section. Existing DSC files do not need to update the value.~~

> This revision of the specification adds support for specifying a GUID value as a Registry Format GUID in addition to a C Format GUID. Any DSC file that uses this feature must use the 0x0001001B DSC_SPECIFICATION value. Older DSC files that do not use this feature do not need to update the value.

Individual items must appear on a single line, they may not span multiple lines.

Parameters*

SpecVal	
	~~New DSC files or DSC files that get updated to use any of the new features defined in this specification must ensure that the 0x0001001A value is used. The EDK II build system must maintain backward compatibility, therefore, there is no requirement to change existing DSC files if no other content changes. This value may also be specified as a decimal value of 1.26.~~
	> New DSC files or DSC files that get updated to use any of the new features defined in this specification must ensure that the 0x0001001B value is used. The EDK II build system must maintain backward compatibility, therefore, there is no requirement to change existing DSC files if no other content changes. This value may also be specified as a decimal value of 1.27.~~

Example

```
[Defines]
  PLATFORM_NAME = NT32
  PLATFORM_GUID = EB216561-961F-47EE-9EF9-CA426EF547C2
  PLATFORM_VERSION = 0.3
  ~~DSC_SPECIFICATION = 0x0001001A~~
  > DSC_SPECIFICATION = 1.27
  OUTPUT_DIRECTORY = Build/Nt32
  SUPPORTED_ARCHITECTURES = IA32
  BUILD_TARGETS = DEBUG|RELEASE
  RFC_LANGUAGES = "en-us;zh-hans;fr-fr"
  ISO_LANGUAGES = "engchnfra"
  SKUID_IDENTIFIER = SkuTwo|DEFAULT
```

REGISTRY FORMAT GUIDS DOCUMENT CHANGE NOTICE

Change Document: EDK II Platform Description (DSC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 01 Feb 2016

Summary

This DCN adds support for specifying GUID values as a Registry Format GUID as well as a C Format GUID.

FLASH DESCRIPTION FILE (FDF)
DCN FOR THE EDK FDF SPECIFICATION

(http://github.com/tianocore-docs/Docs)

Target Revision: 1.27

Target Release Date: February 26, 2016

1. Add support multiple binary files in a FILE RAW.

Revision History

1. Revision number must change when there is a change to the specification that includes new content.
2. Changes considered bug fixes to the specification may use the same revision number, with an Errata tag. Errata tags are alpha characters.

Revision Number	Description	Date
1.26	Updates:	January 2016
	- Specification revision to 1.26	
	- Revised WORKSPACE wording for updated build system that can handle packages located outside of the WORKSPACE directory tree (refer to the TianoCore.org/EDKII website for additional instructions on setting up a development environment).	
	- Add support to generate an FMP Capsule in section 3.7 and added new section for FMP Payload data.	
	- Added new system environment variables, PACKAGES_PATH and EDK_TOOLS_BIN, used by the build system.	
	- Allow INF statements in FD regions	
	- Clarify `[UserExtensions]` content in chapter 2 (to match implementation)	
> 1.27	*> Updates:*	*> Target Date*
	> - Specification revision to 1.27	
	> - Allow support multiple binary files in FILE RAW section	

Individual DCNs

FDF VERSION UPDATE
DOCUMENT CHANGE NOTICE

Change Document: EDK II Flash Definition (FDF) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 29 Jan 2016

Summary

Changes the `FDF_SPECIFICATION` value from `0x0001001A` to `0x0001001B` or `1.27`

2 FDF DESIGN DISCUSSION

~~This revision of the specification adds support for multiple EDK II Packages directories outside of the WORKSPACE. FDF files that use this feature must use the new `FDF_SPECIFICATION = 0x00010019` in the `[Defines]` section. Older FDF files do not need to update the FDF_SPECIFICATION value. ~~

> This revision of the specification adds support for multiple binary files in an FV `FILE RAW` statement. FDF files that use this feature must use the new `FDF_SPECIFICATION = 0x0001001B` in the `[Defines]` section. Older FDF files do not need to update the `FDF_SPECIFICATION` value.

3.4 [Defines] Section

This is an optional section. This section, if present, must be the first section following comment blocks at the beginning of the file.

Summary

This section describes the defines section content in the FDF files. This file can be created by a developer and is an input to the EDK II build tool parsing utilities. Elements may appear in any order within this section.

~~The code for this version of the FDF specification is "0x0001001A" and new versions of this specification must increment the minor (001A) portion of the specification code for backward compatible changes, and increment the major number for non-backward compatible specification changes.~~

~~This revision of the specification adds FMP Capsule support. Any FDF file that uses this feature must use the 0x0001001A FDF_SPECIFICATION value. Older FDF files that do not use this feature do not need to update the value.~~

> The code for this version of the FDF specification is "0x0001001B" and new versions of this

specification must increment the minor (001B) portion of the specification code for backward compatible changes, and increment the major number for non-backward compatible specification changes.

This revision of the specification adds support for multiple binaries in an FV or Capsule `RAW FILE` statement. Any FDF file that uses this feature must use the value `0x0001001B` in the `FDF_SPECIFICATION` statement. Older FDF files that do not use this feature do not need to update the value.

Conditional statements may be used anywhere within this section.

Parameters*

Expression	
	Refer to the EDK II Expression Syntax Specification for more information.
FDF_VERSION	
	The version number for this flash definition; the value is not used by build tools, but the version element is provided for user tracking capabilities that may be used by other user interface tools.
FDF_SPECIFICATION	
	~~For this specification, the version value is 0x0001001A. Tools that process this version of the FDF file can successfully process earlier versions of the FDF files (this is a backward compatible update). If an FDF file with an earlier version of the `FDF_SPECIFICATION` is modified to use the FMP Payload section and FMP Capsule definitions, the version value should be updated to 0x0001001A. There is no requirement to change existing entries if no other content changes. This value may also be specified as decimal value, such as 1.26.~~
	For this specification, the version value is `0x0001001B`. Tools that process this version of the FDF file can successfully process earlier versions of the FDF files (this is a backward compatible update). If an FDF file with an earlier version of the `FDF_SPECIFICATION` is modified to use new features, the version value should be updated to `0x0001001B`. There is no requirement to change existing entries if no other content changes. This value may also be specified as decimal value, such as 1.27.

Example

```
[Defines]
~~FDF_SPECIFICATION = 0x0001001A~~
>  FDF_SPECIFICATION = 1.27
  DEFINE BIG_STUFF = False
  SET gEfiMyPlatformTokenSpaceGuid.MyUsbFlag = True
```

EXTEND FILE RAW FORMAT DOCUMENT CHANGE NOTICE

Change Document: EDK II Flash Definition (FDF) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 27 Jan 2016

Summary

1. Extend the FV and Capsule, FILE RAW statement format to support multiple binary files.

3.6 [FV] Sections

Prototype

```
<type3>     ::= <TS> "FILE" <MTS> "RAW" <Eq> <NamedGuidOrPcd>
                <Options2>

...

<Options2>  ::= [<Use>] [<FileOpts>] <MTS>
                "{" [<EOL>]
~~                    {<Filename>} {<SectionData>} <TS>~~
~~                  "}" [<EOL>]~~
>                     <TS> {<Filename>} {<FileList>+} {<SectionData> <I
                "}" <EOL>
>
> <FileList> ::= <TS> [<FfsAlignment>] <NormalFile> <EOL>

...

~~<Filename> ::= {<FvImage>} {<FdImage>} {<NormalFile>}~~
> <Filename> ::= <TS> {<FvImage>} {<FdImage>} {<NormalFile>} <EOL>
```

Examples

Append the following at the end of the Example in 3.6*

```
  FILE RAW = 197DB236-F856-4924-90F8-CDF12FB975F3 {
  $(OUTPUT_DIRECTORY)/$(TARGET)_$(TOOL_CHAIN_TAG)/$PLATFORM_ARCH)/Fil
  }

  FILE RAW = 197DB236-F856-4924-90F8-CDF12FB975F3 {
    Align=16 $(PLATFORM_PACKAGE)/Binaries/File1.pdb
    Align=16 $(PLATFORM_PACKAGE)/Binaries/File1.pdb
    Align=16 $(PLATFORM_PACKAGE)/Binaries/File1.pdb
  }
```

3.7 [Capsule] Sections

Prototype

```
<type3>     ::= <TS> "FILE" <MTS> "RAW" <Eq> <NamedGuidOrPcd>
                <Options2>

...

<Options2>  ::= [<Use>] [<FileOpts>] <MTS>
                "{" [<EOL>]
~~                   <TS> {<Filename>} {<SectionData>} <TS>~~
>                   {<Filename>} {<FileList>+} {<SectionData> <EOL>}
                "}" [<EOL>]
>
> <FileList> ::= <TS> [<FfsAlignment>] <NormalFile> <EOL>

...

~~<Filename>   ::= {<FvImage>} {<FdImage>} {<NormalFile>}~~
> <Filename>   ::= <TS> {<FvImage>} {<FdImage>} {<NormalFile>} <EOL>
```

BUILD SPECIFICATION
DCNS FOR THE EDK II BUILD SPECIFICATION

(http://github.com/tianocore-docs/Docs)

Target Revision: 1.27

Target Release Date: February 26, 2016

Target Revision: 1.28

Target Release Date 2: June 30, 2016

Revision History

1. Revision number must change when there is a change to the specification that includes new content.
2. Changes considered bug fixes to the specification may use the same revision number, with an Errata tag. Errata tags are alpha characters.

Revision Number	Description	Date
1.26	Updates:	January 2016
	- Specification revision to 1.26	
	- Removed data structure definitions (duplicates from PE/COFF, PI Specifications and TE headers) in Chapter 3 and included references to the industry specifications to remove potential typographical errors and inconsistencies.	
	- Removed Setup and Getting Started sections from Quick Start chapter 6 - this information is available on the TianoCore.org web-site.	
	- Revised WORKSPACE wording for updated build system that can handle packages located outside of the WORKSPACE directory tree (refer to the TianoCore.org/EDKII website for additional instructions on setting up a development environment). Added new, optional system environment variables used by the build system in this environment.	
	- Provide clarification on VPD data generation and report for VPD data content	
	- Clarify precedence of the DPX_SOURCE and [Depex] section.	
	- Specify the alignment required for VOID* PCDs based on the string, Unicode string or byte-code array values.	
	- Remove Unicode file storage requirement; refer to the Multi-String UNI File Format Specification instead.	
	- Clarify BUILDRULEORDER	
	- Add support for INF statement in an FD region.	
> 1.27	*> Updates:*	*> Target Date*
	> - Specification revision to 1.27	
	> - Allow Registry Format GUID values as well a C Format GUID values	
	> - Specify how to process Private content specified in DEC files to only be used by modules within a package	
	> - Add new sub-section to the Build Report to show PCDs that are used in conditional directive statements in the DSC and FDF files that are not used in code by modules.	
	> - Add an SHA1 HASH of a module's .efi file if the `-Y HASH` flag is present on the command-line.	
	> - Add support for calling pre-build and post-build tools if the DSC file contains `PREBUILD` or `POSTBUILD` entries in the DSC's `[Defines]` section.	
	> - Allow PCD values to be overridden by a command-line option: `--pcd TokenSpaceGuidCName.PcdCname=Value`. String values must be encapsulated by double quote marks.	
	> - Update processing rules to support merging multiple binary files (FDF `FILE RAW`) into a single RAW file.	
		>

> 1.28	*> Updates:*	*Target Date 2 **
	> - Specification revision to 1.28	
	- Add new sub-section to the Build Report to show MACROS that are used in conditional directive statemens in the DSC and FDF files.	

Individual DCNs

EXTEND FILE RAW FORMAT DOCUMENT CHANGE NOTICE

Change Document: EDK II Build Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 5 Feb 2016

Summary

FDF spec will allow specifying multiple binaries in the `FILE RAW` statements in FV and Capsule sections.

10.3.2 Creating EFI Sections

Append the following at the end of this section

`FILE` statements in the FDF file's `[FV]` or `[Capsule]` sections are provided so that a platform integrator can include complete EFI FFS files, as well as a method for constructing FFS files using curly "{}" brace scoping.

FFS file specification syntax is one of the following:

```
FILE Type $(NAMED_GUID) [Options] FileName
```

OR

```
FILE Type $(NAMED_GUID) [Options] {
SECTION SECTION_TYPE = FileName
SECTION SECTION_TYPE = FileName
}
```

When constructing a `RAW` type, (`EFI_FV_FILETYPE_RAW`) FFS file, the content is parsed and if multiple binary files are listed, they will be merged into a single binary file. Each file is appended sequentially in the order listed in the FDF file. If an Align value is specified, padding values of 0 may be inserted prior to appending the file to ensure correct address alignment. If no alignment value is specified, the files no padding is performed (byte alignment). If the alignment value is `Auto`, then each file will start on an architectural natural boundary.

COMMMAND-LINE --PCD OPTION DOCUMENT CHANGE NOTICE

Change Document: EDK II Platform Description (DSC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 5 Feb 2016

Summary

Update PCD processing to specify that a PCD's value may be set on the command-line, using the option:

```
--pcd <TokenSpaceGuidCname>.<PcdCName>=<Value>
```

A value on the command-line has the highest precedence. It will override all instances of the PCD's value specified in the DSC or FDF file.

PRE- AND POST- BUILD COMMAND ENTRIES DOCUMENT CHANGE NOTICE

Change Document: EDK II Platform Description (DSC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 26 Feb 2016

Summary

Update build process to call tools specified in the DSC `[Defines]` section, `PREBUILD` and `POSTBUILD` entries.

PROCESS PRIVATE CONTENT DOCUMENT CHANGE NOTICE

Change Document: EDK II Build Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 5 Feb 2016

Summary

Process new syntax in the DEC file that specifies information that can only be used by modules within the package. When modules outside the packages attempt to use this content, the EDK II build system must break with an error regarding content not found.

The four sections, Includes, Ppis, Guids and Protocols headers will be the keyword, Private, following the architecture modifier. If Private is not present, then the content is usable by modules outside the package.

8.2.5 Post processing

Once all files are parsed, the build tools will do following work for each EDK II module:

- Resolve the library classes to library instances, inherit and resolve library classes from them recursively, until no new library instances are found.
- Re-order the library instances according to the consuming relationship and their constructors. For each EDK II module, the tools must select one library instance per required library class (with the exception of the `NULL` library class keyword) using the following precedence (high to low):
  - The DSC file's component INF scoping `<LibraryClasses>` section
  - The DSC file's `[LibraryClasses.arch.module_type]` section tags with both architecture and module type modifiers
  - The DSC file's common arch with a module type modifier, `[LibraryClasses.common.module_type]`
  - DSC file's architecture specific modifier only [LibraryClasses.arch]
  - The DSC file's common `[LibraryClasses]` section

Note: *For modules of type `USER_DEFINED`, if a `NULL` library class is required, the library instance should be listed in the INF scoping `<LibraryClasses>` section of the component.*

- Inherit GUIDs, Protocols and PPIs from all library instances obtained above, and determine values or type of them. The value of a GUID, Protocol or PPI is defined in DEC file.

  Note: *If GUID, Protocol or PPI is listed in a DEC file, where the `Private` modifier is used in the section tag (`[Guids.common.Private]` for example), only modules within the package*

are permitted to use the GUID, Protocol or PPI. If a module or library instance outside of the package attempts to use the item, the build must fail with an appropriate error message.

- Inherit PCDs from all library instances obtained above and determine values and type. The value and type of a PCD are obtained from a DSC file, INF file or DEC file if it cannot be found in the DSC or INF file. For each EDK II module, the tools must obtain unique PCD values using the following precedence (high to low):
  - The DSC file's component INF scoping sections

8.4.1 PCD Rules

The subsections that follow cover the rules for processing PCDs defined in FDF, DSC, INF or DEC files.

8.4.1.1 General Rules:

1. A FeatureFlag PCD cannot use be listed under any other access method in the DEC file. If a PCD name is listed in an FeatureFlag section, and also in another section type, the build must break.
2. For PCDs using Dynamic or DynamicEx access methods, the PCD must be listed in the DSC file. The build parser must break with an appropriate error message if a Dynamic or DynamicEx PCD is not specified in the DSC.
3. For a given platform build, a PCD can only use one access method. Any INF files in a platform that specifically limit the PCD access method for a given PCD must all list the same access method OR for source INF files only, the list the PCD in a [Pcd] section.
4. BINARY INF files (that do not list files under a `[Sources]` section) can only contain `[PcdEx]` and `[PatchPcd]` Sections - if they contain any other type of PCD, break the build.
5. Command line cannot be used to set the PCD value.

6.If a PCD has a Token Space GUID specified in DEC file and the `[Guids]` section tag contains the Private modifier (`[Guids.common.Private]` for example), the PCD may only be used by modules in the package containing the DEC file. If a module outside of that package attempts to use the PCD, the build must break with an appropriate error message.

REGISTRY GUID SUPPORT

REPORT UPDATE - MODULE SHA1 HASH DOCUMENT CHANGE NOTICE

Change Document: EDK II Platform Description (DSC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 5 Feb 2016

Summary

Update the Report to include:

- Display a SHA1 HASH of the .efi file in the Module's Summary Section if the `-Y HASH` option is present

---

13.1 Build Report Generation

Add new item at the end of the defintions, before the Note

> Module Information	
	> Details of the module, may include the HASH of the .efi file.

13.2 Sample Launch Steps: NT32 Platform

starting with step 5b:

1. Run "build.exe -a IA32 -p Nt32Pkg\Nt32Pkg.dsc -y ReportFile.txt"

   a) **-y**: This option specifies the output file name for build report.

   ~~b -Y: This option specifies flags that control the type of build report. It must be from the set of `PCD, LIBRARY, DEPEX, BUILD_FLAGS, FLASH, FIXED_ADDRESS`and `EXECUTION_ORDER`. To specify more than one flag, repeat the option on the command line. Example of usage:~~

   b) **-Y**: This option specifies flags that control the type of build report. It must be from the set of `PCD, LIBRARY, DEPEX, HASH, BUILD_FLAGS, FLASH, FIXED_ADDRESS` and `EXECUTION_ORDER`. To specify more than one flag, repeat the option on the command line. Example of usage: On the command line, append the following arguments:

```
"-y report_filename.txt -Y PCD -Y FLASH -Y DEPEX"
```

~~The default set of flags (if **-Y** is not specified) is: `PCD, LIBRARY, FLASH, DEPEX, BUILD_FLAGS` and `FIXED_ADDRESS`.~~

The default set of flags (if **-Y** is not specified) is: `PCD, LIBRARY, FLASH, DEPEX, HASH, BUILD_FLAGS`and `FIXED_ADDRESS`.

13.3.2 Section and Sub-section Format

Example

```
Platform Name: NT32
Platform DSC Path: s:\edk2\Nt32Pkg\Nt32Pkg.dsc
Architectures: IA32
Tool Chain: VS2008x86
Target: DEBUG
Output Path: s:\edk2\Build\NT32IA32
Build Environment: Windows-7-6.1.7601-SP1
Build Duration: 00:01:53
~~Report Contents: PCD, LIBRARY, BUILD_FLAGS, DEPEX, FLASH, FIXED_ADI

  Report Contents: PCD, LIBRARY, HASH, BUILD_FLAGS, DEPEX, FLASH, FIX

>==============================================================================
Firmware Device (FD)
FD Name: NT32
Base Address: 0x0
Size: 0x2A0000(2688KB)
==============================================================================
>------------------------------------------------------------------------------
FD Region
Type: FV
Base Address: 0x0
Size: 0x280000 (2560K)
FV Name: FvRecovery (65.9% Full)
Occupied Size: 0x1A6028 (1688K)
Free Size: 0xD9FD8 (872K)
Offset Module
------------------------------------------------------------------------------
..(List of Module in FvRecovery)
<------------------------------------------------------------------------------
>------------------------------------------------------------------------------
..(List of other FD region sub-section)
>==============================================================================
```

13.7.1 Module Section Summary

The following Entries are options:

- If using defaults or the HASH flag is specified:
  - SHA1 HASH: %SHA1 HASH% and *%Module .efi file name%

- UEFI Specification Version: %The UEFI specification version:'UEFI_SPECIFICATION_VERSION' in INF [Defines] section%
- PI Specification Version: %The PI specification version:'PI_SPECIFICATION_VERSION' in the INF [Defines] section%
- PCI Device ID: %The PCI device ID for the device: 'PCI_DEVICE_ID' in INF [Defines] section%

Update Examples

Example1:

```
>=============================================================================
Module Summary
Module Name:          SmbiosDxe
Module INF Path:      MdeModule\Universal\SmbiosDxe\SmbiosDxe.inf
File GUID:            F9D88642-0737-49BC-81B5-6889CD57D9EA
Size:                 0x7000 (28.00K)

> SHA1 HASH:            d94c3f180f25d6b562f477bc4a16b286cb66a8b6 *Smb

Build Time Stamp:     1969-12-31 16:00:00
Driver Type:          0x7 (DRIVER)
=============================================================================
... (Module Section Details for SmbiosDxe)
<=============================================================================
```

Example2:

```
>=============================================================================
Module Summary
Module Name:          EbcDxe
Module INF Path:      MdeModule\Universal\EbcDxe\EbcDxe.inf
File GUID:            13AC6DD0-73D0-11D4-B06B-00AA00BD6DE7
Size:                 0x9000 (36.00K)

> SHA1 HASH:            ff4c019345614afe5c88e7fc37219c30a07f4af4 *EbcD

Time Stamp:           1969-12-31 16:00:00
Driver Type:          0x7 (DRIVER)
=============================================================================
... (Module Section Details for EbcDxe)
<=============================================================================
```

REPORT UPDATE - PCDS DOCUMENT CHANGE NOTICE

Change Document: EDK II Platform Description (DSC) File Specification

(http://github.com/tianocore-docs/Docs)

Current Document Revision: 1.26

Target Revision: 1.27

DCN Date: 23 Feb 2016

Summary

Update the Report to include:

- Listing of PCDs that are used in conditional directives in the DSC and FDF files that are not used in module code.
- Update PCD keys to show command-line option override of values

---

13.3.1 Layout

The layout of the text report file:

```
|---- Platform summary

      |----- Conditional directives section
      |----- Unused PCDs section

    |----- Global PCD section
    |----- FD section*
        |---- FD Region sub-section*
        |---- VPD PCD Data sub-section*
    |---- Module section*
        |---- Basic Information summary
        |---- PCD sub-section
        |---- Library sub-section
        |---- DEPEX sub-section
        |---- Build_flags sub-section
        |---- Notification sub-section
```

13.4 Platform Summary

Platform summary displays at the beginning of the output report, including the following items:

- Platform Name : %Platform UI name: 'PLATFORM_NAME' in DSC [Defines] section%
- Platform DSC Path: %Path of platform DSC file%
- Architectures : %List string of all architectures used in build%
- Tool Chain : %Tool chain string%
- Target : %Target String"
- Output Path : %Path to platform build directory%
- Build Environment : %Environment string reported by Python%
- Build Duration : %Build duration time string%
- Report Content : %List of flags the control the report content%

If the DSC or FDF file contains conditional directive statements (`!if, !elseif, !ifdef` or `!ifndef`) or the value of PCD is not used by a module is set in the DSC file (PCD Sections) or the FDF file (SET statements for example), the following sub-section may appear.

The sub-section title will start with the following:

```
==============================================================================
Conditional Directives used by the build system
==============================================================================
```

If the DSC or FDF file define values for PCDs that are not used by any module and are not used in conditional directive statements, the following sub-section may appear.

```
==============================================================================
PCDs not used by modules or in conditional directives
==============================================================================
```

13.4.1 PCDs in Conditional Directives

If a PCD is used in a conditional directive statement, the PCD section will be displayed.

PCD values derived from expressions or other PCDs are not differentiated in the report. Only the final value is displayed.

The first line is required:

`[*P|*F|*B] <PcdCName>: <PcdType> (<DatumType>) = <PcdValue>`

- `*P` means the Pcd's value was obtained from the DSC file
- `*F` means the PCD's value was obtained from the FDF file.
- `*B` means the PCD's value set by a build option.

Additional lines may be displayed showing default values when the value is not a default value.

Example

```
> >==============================================================================
> Conditional Directives used by the build system
> ==============================================================================
> PCD statements
> >------------------------------------------------------------------------------
> gMyTokenSpaceGuid
> *P SmmEnable                    : FEATURE (BOOLEAN) = 0x0
>                                             DEC DEFAULT = 0x1
> *B LogEnable                    : FIXED   (UNIT32) = 0x1
>                                             DEC DEFAULT = 0x0
> <------------------------------------------------------------------------------
> >==============================================================================
```

13.4.2 PCDs not used

If a PCD is not used in a conditional directive statement or by a module, the not used PCD section will be displayed.

PCD values derived from expressions or other PCDs are not differentiated in the report. Only the

final value is displayed.

The first line is required:

```
[*P|*F|*B] <PcdCName>: <PcdType> (<DatumType>) = <PcdValue>
```

- *P means the Pcd's value was obtained from the DSC file
- *F means the PCD's value was obtained from the FDF file.
- *B means the PCD's value set by a build option.

Additional lines may be displayed showing default values when the value is not a default value.

Example

```
> >============================================================
> PCDs not used by modules or in conditional directives
> ============================================================
> PCD statements
> >------------------------------------------------------------
> gMyTokenSpaceGuid
> *P SmmEnable                    : FEATURE (BOOLEAN) = 0x0
>                                           DEC DEFAULT = 0x1
> *B UsbEnable                    : FIXED   (UNIT32) = 0x1
>                                           DEC DEFAULT = 0x0
> <------------------------------------------------------------
> >============================================================
```

13.5 Global PCD Section

This section contains the information for all PCDs whose values are the same for all modules in a platform. The content of global PCD sub-section is grouped by token space:

```
gEfiNt32PkgTokenSpaceGuid
…
…
gEfiMdeModulePkgTokenSpaceGuid
…
…
…
```

PCD values derived from expressions or other PCDs are not differentiated in the report. Only the final value is displayed.

Each global PCD item contains one or more lines:

13.5.1 Required line

The first line is required:

~~`[*P|*F| ] <PcdCName>: <PcdType> (<DatumType>) = <PcdValue>`~~

```
[*P|*F|*B] <PcdCName>: <PcdType> (<DatumType>) = <PcdValue>
```

- *P means the Pcd's value was obtained from the DSC file

- *F means the PCD's value was obtained from the FDF file.
- *B means the PCD's value was obtained from a build option.
- ~~If no *P or *F is given, the PCD's value comes from DEC file. If the value obtained from either the DSC or FDF is the same as the value in the DEC, then neither *P nor *F will be shown in the report.~~

- If no *P, *F or *B is shown, the PCD's value comes from DEC file. If the value obtained from either a build option, the DSC or FDF is the same as the value in the DEC, then *B, *P or *F will not be shown in the report.

Examples:

```
*P PcdWinNtFirmwareVolume                : FIXED (VOID*) = L"..\\Fv\\I
*F PcdWinNtFlashNvStorageFtwWorkingBase : FIXED (UINT32) = 0x0028E00(
                                                  DEC DEFAULT = 0x0

> gTokenSpaceGuid
> *B LogEnable                           : FIXED (UNIT32) = 0x1
>                                                 DEC DEFAULT = 0x0
```

CHANGE FORMAT KEY
KEYS

This section documents how changes appear in Document Change Notices.

Instructions to maintainer marked like this.

Removed content marked by ~~this~~

or like this:

```
~~Delete content~~
```

It may appear as strikethrough text or with just the two tilde characters before and after a section.

New content marked like:

> this

or like this:

```
> New Content
```

For Tables,

Header1	**Header2**	**Header3**
Prev	Prev	Prev
~~**DeleteMe**~~	~~**DeleteMe**~~	~~**DeleteMe**~~
> NewCell	*> NewCell*	*> NewCell*

DCN TEMPLATE
DOCUMENT CHANGE NOTICE

Change Document: Name

(http://github.com/tianocore-docs/Docs)

Example:

```
Change Document: EDK II Package Declaration (DEC) File Specification
(http://github.com/tianocore-docs/Docs)
```

Current Document Revision:

Example:

```
Current Document Revsion: 1.25
```

Target Revision:

Example:

```
Target Revision: 1.27
```

DCN Date: ; date user creates the document for review.

Header (use the same number of # charcters as section/sub-section)

Old Content is marked up in-line.

~~Delete this line~~

> Add this line.