# Protocols HII Test

## EFI\_HII\_FONT\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_HII\_FONT\_PROTOCOL Section.

### StringToImage()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.1.1.1 | 0x6fca8706, 0x7d83, 0x4914,  0x8a, 0x16, 0x92, 0x0b, 0x07, 0xb1, 0x68, 0xb9 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_INVALID\_PARAMETER with String been NULL. | Call StringToImage() with valid parameters except String being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.1.2 | 0x80ee2790, 0x9ff7, 0x4abe,  0x90, 0xaf, 0x05, 0x4a, 0x86, 0x69, 0xba, 0x51 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_INVALID\_PARAMETER with Blt been NULL. | Call StringToImage() with valid parameters except Blt being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.1.3 | 0xe2f66ec3, 0x585a, 0x45ba,  0x8f, 0x7a, 0xd5, 0x18, 0x5f, 0xeb, 0x4e, 0x9a | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_INVALID\_PARAMETER with wrong flag combination. | Call StringToImage() with Flag being EFI\_HII\_OUT\_FLAG\_CLEAN\_X with EFI\_HII\_OUT\_FLAG\_WRAP. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.1.4 | 0xabf68512, 0x0bb8, 0x4ef8,  0x97, 0xc1, 0xda, 0x93, 0x55,0xda, 0x1b, 0x07 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_INVALID\_PARAMETER with wrong flag combination. | Call StringToImage() with Flag being EFI\_HII\_OUT\_FLAG\_CLEAN\_X without EFI\_HII\_OUT\_FLAG\_CLIP. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.1.5 | 0x6ff9c8b4, 0xeb8f, 0x4e0b, 0x9a, 0x97, 0x82, 0x94, 0x37, 0x0c, 0xdd, 0x3c | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with valid parameters. | Call StringToImage() with valid paramenters and use EFI\_GRAPHICS\_OUTPUT\_BLT\_PIXEL structure in EFI\_IMAGE\_OUTPUT structure. |
| 5.18.1.1.6 | 0x182cc281, 0xb462, 0x458f, 0xaa, 0xb6, 0xca, 0x98, 0xb5, 0x27, 0x37, 0x31 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with valid parameters. | Call StringToImage() with valid paramenters and use EFI\_GRAPHICS\_OUTPUT\_PROTOCOL in EFI\_IMAGE\_OUTPUT structure. |
| 5.18.1.1.7 | 0xcdf439d0, 0xe471, 0x4fe7, 0x86, 0x98, 0xf5, 0xb0, 0x5c, 0xcd, 0xa6, 0xae | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with valid parameters for all ASCII visible characters. Each images must equal to sys default glyph. | Call StringToImage() with valid paramenters and StringInfo = NULL..  Compare image output with system default font glyph image |
| 5.18.1.1.8 | 0xa8f40eac, 0x8633, 0x40ca, 0x95, 0x6d, 0x75, 0xb2, 0x81, 0x50, 0x75, 0x39 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with valid parameters for all ASCII visible characters. Each image must equal to the specific font glyph. | Register a specific font package  Call StringToImage() with valid paramenters and StringInfo = specific font. Compare image output with specific font glyph image registered |
| 5.18.1.1.9 | 0x42dc1626, 0x36ce, 0x421b, 0x8d, 0x66, 0x21, 0xb8, 0xaa, 0x43, 0x6c, 0x7b | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameters EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Call StringToImage() with EFI\_HII\_DIRECT\_TO\_SCREEN. For the final row, the RowInfoArray.LineHeight and RowInfoArray.BaseLine may describe pixels which are outside the limit specified by Blt. Height (unless EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y is specified) even though those pixels were not drawn. 2.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.10 | 0xf8b5b9b6, 0xc3c6, 0x4993, 0x9b, 0x3c, 0xbc, 0x8d, 0x91, 0xee, 0x8c, 0x20 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Call StringToImage with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_DIRECT\_TO\_SCREEN. For the final row, the RowInfoArray.LineHeight and RowInfoArray.BaseLine 'May' describe pixels which are outside the limit specified by Blt. Height (unless EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y is specified) even though those pixels were not drawn. 2.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.11 | 0x4c70adb5, 0xcc05, 0x435a,  0x8c, 0xc4, 0xce, 0xd1, 0x54, 0x6e, 0xd7, 0xf6 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X | EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Call StringToImage() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X | EFI\_HII\_DIRECT\_TO\_SCREEN. If a character's right-most on pixel cannot fit, then it will not be drawn at all. 2.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.12 | 0xa000d36f, 0x2918, 0x448c,  0xad, 0x6d, 0x15, 0x77, 0xb5, 0x2f, 0xdc, 0x66 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y | EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Call StringToImage() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y | EFI\_HII\_DIRECT\_TO\_SCREEN. If a row's bottom-most pixel exceed screen Height, then it will not be drawn at all.  2.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.13 | 0x266f881, 0x409b, 0x47e5,  0x8f, 0x22, 0x21, 0x7d, 0x14, 0xa4, 0x8a, 0xab | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_IGNORE\_IF\_NO\_GLYPH | EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String with line break opportunity | 1.Call StringToImage() with EFI\_HII\_IGNORE\_IF\_NO\_GLYPH | EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String with line break opportunity (SPACE is a line break opportunity). Check display will wrapper at right place.  2.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.14 | 0x2fa4edd2, 0xa193, 0x4882,  0xae, 0x1e, 0xeb, 0xfe, 0xf5, 0x57, 0x42, 0xcc | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String without line break opportunity | 1.Call StringToImage() with EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String without line break opportunity. String is designed to display as if EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X is set.  2.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.15 | 0x57300788, 0xba79, 0x4727, 0xb5, 0xe6, 0xe9, 0x20, 0xcd, 0x7e, 0xd6, 0x93 | HII\_FONT\_PROTOCOL.StringToImage -StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_IGNORE\_LINE\_BREAK | EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Call StringToImage() with EFI\_HII\_IGNORE\_LINE\_BREAK | EFI\_HII\_DIRECT\_TO\_SCREEN. If a row's bottom-most pixel cannot fit, then it will not be drawn at all. This flag requires that EFI\_HII\_OUT\_FLAG\_CLIP be set. 2.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.16 | 0xf3b0daef, 0xab51, 0x4ebc, 0x93, 0x51, 0x74, 0xf6, 0x18, 0xaa, 0x9f, 0x9f | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Register a new font package  2.Call StringToImage() with EFI\_HII\_DIRECT\_TO\_SCREEN. 3.Check EFI\_HII\_DIRECT\_TO\_SCREEN only case If Blt is not NULL, then EFI\_HII\_OUT\_FLAG\_CLIP is implied String is designed to displayed with one full line 4.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.17 | 0x23ab3935, 0x483c, 0x4d75,  0xab, 0x3, 0xef, 0x50, 0x32, 0xea, 0x30, 0xbf | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_CLIP | 1.Register a new font package  2.Call StringToImage() with EFI\_HII\_OUT\_FLAG\_CLIP.  3. For the final row, the RowInfoArray.LineHeight and RowInfoArray.BaseLine may describe pixels which are outside the limit specified by Blt. Height (unless EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y is specified) even though those pixels were not drawn. 4.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.18 | 0x9e992f5a, 0x4a3b, 0x44d8, 0x89, 0x47, 0xca, 0x30, 0x92, 0x2b, 0x69, 0xa5 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X | EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Register a new font package  2.Call StringToImage() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X | EFI\_HII\_DIRECT\_TO\_SCREEN. 3. If a character's right-most on pixel cannot fit, then it will not be drawn at all.  4.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.19 | 0xc8999c53, 0xd56, 0x4545, 0xbc, 0x55, 0x91, 0xf0, 0xd1, 0x1, 0x60, 0x4a | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y | EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Register a new font package  2.Call StringToImage() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y | EFI\_HII\_DIRECT\_TO\_SCREEN. 3.If a row's bottom-most pixel exceed screen Height, then it will not be drawn at all.  4.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.20 | 0x9b71db4d, 0x5a06, 0x4246, 0x83, 0xd2, 0x9d, 0x31, 0x70, 0x73, 0x63, 0xd0 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_IGNORE\_IF\_NO\_GLYPH | EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String with line break opportunity | 1.Register a new font package  2.Call StringToImage() with EFI\_HII\_IGNORE\_IF\_NO\_GLYPH | EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String with line break opportunity (SPACE is a line-break). 3.Check if the display is right. 4.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.21 | 0xb0e526b1, 0xc399, 0x4e31, 0xb2, 0x97, 0xc1, 0x29, 0x18, 0x37, 0x95, 0x79 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String without line break opportunity | 1.Register a new font package  2.Call StringToImage() with EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String without line break opportunity.  3. String is designed to display as if EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X is set.  4.The return code should be EFI\_SUCCESS . |
| 5.18.1.1.22 | 0xcbdae1b4, 0xc99b, 0x4a08, 0x9b, 0xf9, 0x76, 0x69, 0x77, 0x71, 0x66, 0x30 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_IGNORE\_LINE\_BREAK | EFI\_HII\_DIRECT\_TO\_SCREEN | 1.Register a new font package  2.Call StringToImage() with EFI\_HII\_IGNORE\_LINE\_BREAK | EFI\_HII\_DIRECT\_TO\_SCREEN. 3. If a row's bottom-most pixel cannot fit, then it will not be drawn at all. This flag requires that EFI\_HII\_OUT\_FLAG\_CLIP is set. 4. The return code should be EFI\_SUCCESS . |
| 5.18.1.1.23 | 0x36a9a186, 0x363f, 0x4b4b, 0xa3, 0xaf, 0xa9, 0x9b, 0x29, 0x7a, 0x6d, 0x41 | HII\_FONT\_PROTOCOL.StringToImage - StringToImage() returns EFI\_SUCCESS with parameter EFI\_HII\_OUT\_FLAG\_TRANSPARENT | 1.Register a new font package  2.Call StringToImage() with EFI\_HII\_OUT\_FLAG\_TRANSPARENT.  3. Check output buffer StringInfo background should be ignored according to EFI spec.  4. The return code should be EFI\_SUCCESS . |

### StringIdToImage()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.1.2.1 | 0xf4e2c51e, 0x92a3, 0x4752, 0x92, 0x64, 0x27, 0xb1, 0x54, 0x21, 0x70, 0x3a | HII\_FONT\_PROTOCOL.StringIdToImage - StringIdToImage() returns EFI\_INVALID\_PARAMETER with Blt been NULL. | Call StringIdToImage() with valid parameters except Blt being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.2.2 | 0x9aecc9b3, 0x3bff, 0x4c7c, 0x96, 0x6b, 0xa9, 0x64, 0x84, 0xfe, 0xd9, 0x89 | HII\_FONT\_PROTOCOL.StringIdToImage - StringIdToImage() returns EFI\_INVALID\_PARAMETER with PackageList been NULL. | Call StringIdToImage() with valid parameters except PackageList being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.2.3 | 0x479e2e87, 0xf833, 0x4d2b, 0xbb, 0x47, 0x16, 0x77, 0x7b, 0x52, 0xb6, 0x6a | HII\_FONT\_PROTOCOL.StringIdToImage - StringIdToImage() returns EFI\_NOT\_FOUND with an invalid PackageList. | Call StringIdToImage() with valid parameters except an invalid PackageList, The return status should be EFI\_NOT\_FOUND. |
| 5.18.1.2.4 | 0xe1d5168a, 0x26da, 0x4000, 0xa9, 0xc8, 0x15, 0x85, 0xee, 0xea, 0x38, 0x33 | HII\_FONT\_PROTOCOL.StringIdToImage - StringIdToImage() returns EFI\_NOT\_FOUND with StringId not in PackageList. | Call StringIdToIamge() with a StringId which isn’t in PackageList. The return status should be EFI\_NOT\_FOUND. |
| 5.18.1.2.5 | 0xfba0a646, 0x9942, 0x4790, 0x86, 0xef, 0xe8, 0x52, 0x32, 0xf1, 0xb5, 0xeb | HII\_FONT\_PROTOCOL.StringIdToImage - StringToImage() returns EFI\_INVALID\_PARAMETER with invalid Flags combination. | Call StringIdToImage() with Flag being EFI\_HII\_OUT\_FLAG\_CLEAN\_X with EFI\_HII\_OUT\_FLAG\_WRAP. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.2.6 | 0xd9b59551, 0xa799, 0x4c87, 0x89, 0xb3, 0x89, 0xc5, 0x6a, 0xb8, 0x43, 0x9f | HII\_FONT\_PROTOCOL.StringIdToImage - StringToImage() returns EFI\_INVALID\_PARAMETER with invalid Flags combination. | Call StringIdToImage() with Flag being EFI\_HII\_OUT\_FLAG\_CLEAN\_X without EFI\_HII\_OUT\_FLAG\_CLIP. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.2.7 | 0x3df4b27f, 0x7b07, 0x4a3d, 0xaa, 0x09, 0x60, 0xfa, 0xbe, 0x82, 0x99, 0x9f | HII\_FONT\_PROTOCOL.StringIdToImage - StringIdToImage() returns EFI\_SUCCESS with valid parameters. | Call StringIdToImage() with valid paramenters and use EFI\_GRAPHICS\_OUTPUT\_BLT\_PIXEL structure in EFI\_IMAGE\_OUTPUT structure. The return status should EFI\_SUCCESS. |
| 5.18.1.2.8 | 0xedcca70f, 0xcb25, 0x4d22, 0x98, 0x5e, 0x18, 0x86, 0x66, 0x8c, 0xc1, 0x9c | HII\_FONT\_PROTOCOL.StringIdToImage - StringIdToImage() returns EFI\_SUCCESS with valid parameters. | Call StringIdToImage() with valid paramenters and use EFI\_GRAPHICS\_OUTPUT\_PROTOCOL in EFI\_IMAGE\_OUTPUT structure. The return status should EFI\_SUCCESS. |

### GetGlyph()

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 5.18.1.3.1 | 0xb94b394f, 0x8e3e, 0x4adc, 0x8f, 0x5c, 0x64, 0x12, 0x69, 0xa2, 0xed, 0xfe | HII\_FONT\_PROTOCOL.GetGlyph - GetGlyph() returns EFI\_INVALID\_PARAMETER with Blt being NULL. | Call GetGlyph() with Blt being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.3.2 | 0xee445b90, 0xf370, 0x43fd, 0x83, 0xff, 0x00, 0x2d, 0x29, 0x1e, 0xcd, 0x42 | HII\_FONT\_PROTOCOL.GetGlyph - GetGlyph() returns EFI\_INVALID\_PARAMETER with non NULL \*Blt. | Call GetGlyph() with non NULL Blt. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.3.3 | 0x0687a598, 0xa2a6, 0x4073, 0xa7, 0x4f, 0x05, 0xae, 0x9c, 0xe2, 0x1e, 0x33 | HII\_FONT\_PROTOCOL.GetGlyph - GetGlyph() returns EFI\_SUCCESS with valid parameters. | Call GetGlyph() with valid parameters. The return status should be EFI\_SUCCESS. |

### GetFontInfo()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.1.4.1 | 0xf43589d3, 0xfccd, 0x413f, 0xb7, 0x50, 0xf8, 0xb4, 0x00, 0xd2, 0x92, 0x7b | HII\_FONT\_PROTOCOL.GetFontInfo - GetFontInfo() returns EFI\_INVALID\_PARAMETER with invalid EFI\_FONT\_INFO\_MASK Combination. | Call GetFontInfo() with StringInfoIn->FontInfoMask being invalid combination. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.1.4.2 | 0x6e5210d4, 0xead5, 0x4042, 0xac, 0x30, 0xa4, 0xfb, 0x8f, 0x9f, 0xf1, 0x9a | HII\_FONT\_PROTOCOL.GetFont - GetFont() returns EFI\_SUCCESS with valid parameters | Call GetFontInfo() with valid parameters. The return status should be EFI\_SUCCESS. |
| 5.18.1.4.3 | 0x88294411, 0x3dd7, 0x4030, 0xb6, 0x40, 0x65, 0xa3, 0x85, 0x7b, 0x2f, 0x46 | HII\_FONT\_PROTOCOL.GetFont - GetFont() returns EFI\_SUCCESS with valid parameters(StringInfoIn is NULL) | Call GetFontInfo() with valid parameters(StringInfoIn is NULL). The return status should be EFI\_SUCCESS. |

## EFI\_HII\_STRING\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_HII\_STRING\_PROTOCOL Section.

### NewString()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.2.1.1 | 0xb0eb04d6, 0x3328, 0x4157, 0xa8, 0x8e, 0xe9, 0x9a, 0x15, 0x62, 0x6b, 0x88 | HII\_STRING\_PROTOCOL.NewString - NewString() returns EFI\_INVALID\_PARAMETER with StringId being NULL. | Call NewString() with StringId being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.1.2 | 0x9223196c, 0xadf1, 0x4181, 0xbc, 0xc3, 0x1d, 0x9e, 0xa4, 0xcf, 0x7a, 0x8e | HII\_STRING\_PROTOCOL.NewString - NewString() returns EFI\_INVALID\_PARAMETER with Language being NULL. | Call NewString() with Language being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.1.3 | 0x8d9e83aa, 0x9bf1, 0x4466, 0xba, 0xba, 0xec, 0x14, 0xfd, 0xb3, 0x82, 0x14 | HII\_STRING\_PROTOCOL.NewString - NewString() returns EFI\_INVALID\_PARAMETER with String being NULL. | Call NewString() with String being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.1.4 | 0x23b3df9d, 0x2330, 0x4db7, 0xa1, 0x71, 0x0c, 0x2a, 0x61, 0xb7, 0xd2, 0x24 | HII\_STRING\_PROTOCOL.NewString - NewString() returns EFI\_INVALID\_PARAMETER with PackageList beinf NULL. | Call NewString() with PackageList being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.1.5 | 0x2077cb3b, 0xb8b4, 0x4ba9, 0xab, 0x49, 0x36, 0xc4, 0xe3, 0xb7, 0x1e, 0xb5 | HII\_STRING\_PROTOCOL.NewString - NewString() returns EFI\_SUCCESS with valid parameters and result checked. | Part 1: Call NewString() with valid parameters. The return Status should be EFI\_SUCCESS. |
| 5.18.2.1.6 | 0x8cd4cc42, 0xe5f0, 0x4f6f, 0x9f, 0x7d, 0x60, 0x47, 0x95, 0xd5, 0x05, 0x36 | HII\_STRING\_PROTOCOL.NewString - output the string and compare with the original string. | Part2: Call GetString() to check the output string with the original string. They should be same. |

### GetString()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.2.2.1 | 0x640acc2d, 0x1174, 0x4735,  0x94, 0xb3, 0xbc, 0xe2, 0xca, 0xbb, 0x92, 0xc1 | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_NOT\_FOUND with StringId being invalid. | Call GetString() with an invalid StringId. The return status should be EFI\_NOT\_FOUND. |
| 5.18.2.2.2 | 0x3c0c9dfe, 0xe56e, 0x43ee, 0x80, 0x26, 0x55, 0xb1, 0x14, 0x29, 0x2c, 0x38 | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_NOT\_FOUND with an invalid PackageList. | Call GetString() with an invalid PackageList. The return status should be EFI\_NOT\_FOUND. |
| 5.18.2.2.3 | 0x0460a672, 0xcba9, 0x4ee8, 0x9e, 0x43, 0x9d, 0xba, 0x85, 0x52, 0x3f, 0xab | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_BUFFER\_TOO\_SMALL with StringSize indicates the String is too small. | Call GetString() with StringSize which indicates the String buffer is small. The return status should EFI\_BUFFER\_TOO\_SMALL. The StringSize is updated with the required size. |
| 5.18.2.2.4 | 0xeed5460f, 0x826e, 0x4e1b, 0xad, 0x79, 0xb7, 0x3b, 0x58, 0xc9, 0x57, 0x01 | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_INVALID\_LANGUAGE with string is not in the specified Language. | Call GetString() with string specified by StringId is available but not in the specified Language. The return status should be EFI\_INVALID\_LANGUAGE. |
| 5.18.2.2.5 | 0xafd0b70c, 0xe1b4, 0x43c1, 0x94, 0x60, 0x96, 0xf5, 0x3e, 0xe9, 0xaa, 0xe9 | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_INVALID\_PARAMETER with Language being NULL. | Call GetString() with Language being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.2.6 | 0xcf15f5f5, 0x7eaf, 0x4e63, 0x80, 0xd2, 0x5c, 0x9b, 0x89, 0x02, 0x1b, 0xf8 | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_INVALID\_PARAMETER with String being NULL. | Call GetString() with String being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.2.7 | 0xc37a209f, 0xaeab, 0x4152, 0xbf, 0x74, 0x27, 0x27, 0xea, 0x48, 0x4f, 0x38 | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_INVALID\_PARAMETER with StringSize being NULL. | Call GetString() with StringSize being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.2.8 | 0x62a545c3, 0x3da2, 0x4f46, 0xb9, 0x07, 0xd4, 0xfe, 0x3e, 0xdf, 0x59, 0xc0 | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_INVALID\_PARAMETER with PackageList been NULL. | Call GetString() with PackageList being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.2.9 | 0x276f380d, 0x96d6, 0x46d5, 0x8a, 0xbb, 0x2a, 0xf3, 0xb7, 0x3c, 0x2d, 0x43 | HII\_STRING\_PROTOCOL.GetString - GetString() returns EFI\_SUCCESS with valid parameters and the result checked. | Step1: Call NewString() with valid parameters.  Step2: Call GetString() with valid parameters. The return status should be EFI\_SUCCESS. The output string should be same with the original one. |

### SetString()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.2.3.1 | 0xb7d699ce, 0xb3e9, 0x4327,  0x8b, 0x52, 0xdd, 0xd5, 0xa2, 0xff, 0xb9, 0x0c | HII\_STRING\_PROTOCOL.SetString - SetString() returns EFI\_NOT\_FOUND with StringId been invalid. | Call SetString() with an invalid StringId which is not in the database. The return status should be EFI\_NOT\_FOUND. |
| 5.18.2.3.2 | 0xfda7ec68, 0xbf34, 0x4086, 0xad, 0x72, 0x26, 0xe1, 0xd6, 0xdd, 0x45, 0x48 | HII\_STRING\_PROTOCOL.SetString - SetString() returns EFI\_INVALID\_PARAMETER with Language been NULL. | Call SetString() with Language being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.3.3 | 0xb66221c2, 0xc6e7, 0x4129, 0xb3, 0x83, 0xa6, 0x51, 0x26, 0x2b, 0xcf, 0x57 | HII\_STRING\_PROTOCOL.SetString - SetString() returns EFI\_INVALID\_PARAMETER with String been NULL. | Call SetString() with String being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.3.4 | 0x7439d8aa, 0xe2f6, 0x4c3b, 0x98, 0x0c, 0x13, 0xbd, 0xab, 0x97, 0xff, 0x95 | HII\_STRING\_PROTOCOL.SetString - SetString() returns EFI\_NOT\_FOUND with an invalid PackageList. | Call SetString() with an invalid PackageList. The return status should be EFI\_NOT\_FOUND. |
| 5.18.2.3.5 | 0x66495376, 0x042b, 0x460a, 0xbb, 0x45, 0x19, 0xfd, 0x13, 0xf2, 0xe0, 0x2c | HII\_STRING\_PROTOCOL.SetString - SetString() returns EFI\_INVALID\_PARAMETER with PackageList been NULL. | Call SetString() with PackageList being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.3.6 | 0xf346d13b, 0xcbd0, 0x451f, 0xa6, 0x93, 0x75, 0xf1, 0xe9, 0xdd, 0x1f, 0x74 | HII\_STRING\_PROTOCOL.SetString - SetString() returns EFI\_SUCCESS with valid parameters and result checked | Part 1: Call SetString() with valid parameters. The return status should be EFI\_SUCCESS. |
| 5.18.2.3.7 | 0xbf8f4ae6, 0xf506, 0x43d2, 0xa6, 0x43, 0xa7, 0xb4, 0xb2, 0x33, 0xe8, 0xe0 | HII\_STRING\_PROTOCOL.SetString - output the string and compare with the reset string. | Part2: Call GetString() to check the output string with the original string. They should be same. |

### GetLanguages()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.2.4.1 | 0x7a983202, 0x322e, 0x4d12, 0x90, 0xb3, 0xcf, 0x8b, 0x6e, 0xc4, 0x97, 0x5b | HII\_STRING\_PROTOCOL.GetLanguages - GetLanguages() returns EFI\_INVALID\_PARAMETER with Languages been NULL. | Call GetLanguages() with Languages being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.4.2 | 0xa9299182, 0xcd9a, 0x4014, 0xb4, 0x03, 0xe2, 0x67, 0xc7, 0xf4, 0x80, 0x7f | HII\_STRING\_PROTOCOL.GetLanguages - GetLanguages() returns EFI\_INVALID\_PARAMETER with LanguagesSize been NULL. | Call GetLanguages() with LanguagesSize being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.4.3 | 0x83a0f73c, 0xdd2c, 0x4652, 0x8e, 0xbe, 0x32, 0xd5, 0xf9, 0x8e, 0x24, 0xef | HII\_STRING\_PROTOCOL.GetLanguages - GetLanguages() returns EFI\_NOT\_FOUND with an invalid PackageList. | Call GetLanguages() with an invalid PackageList. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.4.4 | 0x696870ed, 0xfff5, 0x4b76, 0x9f, 0x82, 0xbe, 0x78, 0xf6, 0x58, 0x9b, 0x8b | HII\_STRING\_PROTOCOL.GetLanguages - GetLanguages() returns EFI\_INVALID\_PARAMETER with PackageList been NULL. | Call GetLanguages() with PackageList being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.2.4.5 | 0x65dca7c5, 0x85a0, 0x48a0, 0x9a, 0x49, 0xa9, 0xbb, 0xae, 0xa2, 0x55, 0xf3 | HII\_STRING\_PROTOCOL.GetLanguages - GetLanguages() returns EFI\_BUFFER\_TOO\_SMALL with LanguagesSize indicates the Languages is too small. | Call GetLanguages() with LanguagesSize which indicates the Languages buffer is small. The return status should EFI\_BUFFER\_TOO\_SMALL. The LanguagesSize is updated with the required size. |
| 5.18.2.4.6 | 0xba61367b, 0x33b6, 0x41cc, 0x94, 0x60, 0x54, 0x75, 0xf1, 0xe5, 0x81, 0x89 | HII\_STRING\_PROTOCOL.GetLanguages - GetLanguages() returns EFI\_SUCCESS with valid parameters. | Call GetLanguages() with valid parameters. The return status should be EFI\_SUCCESS. |

### GetSecondaryLanguages()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.2.5.1 | 0xff558856, 0xcf19, 0x47b2, 0x89, 0xc0, 0xdb, 0xdf, 0x0e, 0xf5, 0x31, 0xe2 | HII\_STRING\_PROTOCOL.GetSecondaryLanguages - GetSecondaryLanguages() returns EFI\_INVALID\_PARAMETER with FirstLanguage been NULL. | Call GetSecondaryLanguages() with FirstLanguage being NULL. The return status should EFI\_INVALID\_PARAMETER. |
| 5.18.2.5.2 | 0x05c043da, 0xd0dd, 0x4833, 0xa1, 0x27, 0x92, 0x3b, 0x6a, 0x58, 0x05, 0xdc | HII\_STRING\_PROTOCOL.GetSecondaryLanguages - GetSecondaryLanguages() returns EFI\_INVALID\_PARAMETER with SecondLanguages been NULL. | Call GetSecondaryLanguages() with SecondLanguages being NULL. The return status should EFI\_INVALID\_PARAMETER. |
| 5.18.2.5.3 | 0xa891d992, 0x6296, 0x4670, 0xa5, 0xbe, 0x5c, 0x53, 0xaa, 0xc0, 0x34, 0x48 | HII\_STRING\_PROTOCOL.GetSecondaryLanguages - GetSecondaryLanguages() returns EFI\_INVALID\_PARAMETER with SecondLanguagesSize been NULL. | Call GetSecondaryLanguages() with SecondLanguagesSize being NULL. The return status should EFI\_INVALID\_PARAMETER. |
| 5.18.2.5.4 | 0x050d991f, 0xd6f0, 0x4a07, 0x91, 0x6d, 0x58, 0xde, 0xc2, 0xec, 0xf3, 0x2f | HII\_STRING\_PROTOCOL.GetSecondaryLanguages - GetSecondaryLanguages() returns EFI\_NOT\_FOUND with an invalid PackageList. | Call GetSecondaryLanguages() with an invalid PackageList. The return status should EFI\_NOT\_FOUND. |
| 5.18.2.5.5 | 0x68d1489e, 0x587b, 0x44e5, 0xb8, 0x72, 0x17, 0xc1, 0x1e, 0xc9, 0xd3, 0xf7 | HII\_STRING\_PROTOCOL.GetSecondaryLanguages - GetSecondaryLanguages() returns EFI\_INVALID\_PARAMETER with PackageList been NULL. | Call GetSecondaryLanguages() with PackageList being NULL. The return status should EFI\_INVALID\_PARAMETER. |
| 5.18.2.5.6 | 0xa25ea8dd, 0x5681, 0x4912, 0xb5, 0xda, 0xe3, 0x04, 0x36, 0x7c, 0x23, 0x89 | HII\_STRING\_PROTOCOL.GetSecondaryLanguages - GetSecondaryLanguages() returns EFI\_NOT\_FOUND with FirstLanguage is not present in the PackageList. | Call GetSecondaryLanguages() with FirstLanguagewhich is not in the specified PackageList. The return status should EFI\_NOT\_FOUND. |
| 5.18.2.5.7 | 0x6750c8c6, 0x54b5, 0x4a95, 0xa4, 0x15, 0x44, 0xbc, 0x64, 0xb1, 0x9f, 0x81 | HII\_STRING\_PROTOCOL.GetLanguages - GetSecondaryLanguages() returns EFI\_BUFFER\_TOO\_SMALL with SecondLanguagesSize indicates the SecondLanguages is too small. | Call GetSecondaryLanguages() with SecondLanguagesSize which indicates the SecondLanguages buffer is small. The return status should EFI\_BUFFER\_TOO\_SMALL. The SecondLanguagesSize is updated with the required size. |
| 5.18.2.5.8 | 0x302b21ca, 0xbc47, 0x4c26, 0xa0, 0x21, 0x24, 0x2d, 0xba, 0x57, 0x42, 0x65 | HII\_STRING\_PROTOCOL.GetSecondaryLanguages - GetSecondaryLanguages() returns EFI\_SUCCESS with SecondLanguagesSize is large enough. | Call GetSecondaryLanguages() with valid parameters. The return status should be EFI\_SUCCESS. |

## EFI\_HII\_IMAGE\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_HII\_IMAGE\_PROTOCOL Section.

### NewImage()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.3.1.1 | 0x20eafa16, 0xc9cd, 0x41b3, 0x96, 0x81, 0x46, 0x7b, 0x7f, 0x17, 0x3d, 0x71 | HII\_IMAGE\_PROTOCOL.NewImage - NewImage() returns EFI\_INVALID\_PARAMETER with ImageId been NULL. | Call NewImage() with ImageId being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.1.2 | 0x0227338d, 0xb459, 0x4209, 0xb1, 0xa0, 0x10, 0x3c, 0xe8, 0x3e, 0x71, 0xf5 | HII\_IMAGE\_PROTOCOL.NewImage - NewImage() returns EFI\_INVALID\_PARAMETER with Image been NULL. | Call NewImage() with Image being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.1.3 | 0x4930f94e, 0x6bdb, 0x42aa, 0xaf, 0xde, 0x87, 0x55, 0x55, 0x2c, 0x77, 0x1d | HII\_IMAGE\_PROTOCOL.NewImage - NewImage() returns EFI\_NOT\_FOUND with PackageList been NULL. | Call NewImage() with PackageList being NULL, The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.1.4 | 0x170bc177, 0xa2f7, 0x46ba, 0xa8, 0xd6, 0x09, 0xe5, 0xa4, 0xb1, 0x81, 0x8f | HII\_IMAGE\_PROTOCOL.NewImage - NewImage() returns EFI\_SUCCESS with valid parameters and result checked. | Call NewImage() with valid parameters, The return status should be EFI\_SUCCESS. |

### GetImage()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.3.2.1 | 0x55488ca5, 0x2a0c, 0x4dcb, 0xbc, 0x7d, 0xca, 0xaf, 0x05, 0x2f, 0xac, 0x13 | HII\_IMAGE\_PROTOCOL.GetImage - GetImage() returns EFI\_NOT\_FOUND with ImageId been invalid. | Call GetImage() with an invalid ImageId which is not in the database. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.2.2 | 0xdde7e63e, 0xa889, 0x47ce, 0xad, 0xe1, 0x15, 0x0b, 0xb8, 0xa3, 0x8e, 0x10 | HII\_IMAGE\_PROTOCOL.GetImage - GetImage() returns EFI\_BUFFER\_TOO\_SMALL with ImageSize is small. | Call GetImage() with ImageSize which indicates the Image buffer is small. The return status should EFI\_BUFFER\_TOO\_SMALL. The ImageSize is updated with the required size. |
| 5.18.3.2.3 | 0xa1f286a0, 0x26da, 0x4919, 0xa3, 0xc4, 0x90, 0x5b, 0x18, 0x03, 0x6c, 0x36 | HII\_IMAGE\_PROTOCOL.GetImage - GetImage() returns EFI\_INVALID\_PARAMETER with Image been NULL. | Call GetImage() with Image being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.2.4 | 0x17a11dcc, 0x8d3d, 0x40dc, 0xb0, 0x9c, 0x37, 0xfc, 0x8e, 0x72, 0x46, 0xab | HII\_IMAGE\_PROTOCOL.GetImage - GetImage() returns EFI\_INVALID\_PARAMETER with ImageSize been NULL. | Call GetImage() with ImageSize being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.2.5 | 0x51363bef, 0x2eb6, 0x4eef, 0x86, 0xdf, 0x48, 0xf1, 0x87, 0x75, 0x6f, 0x9e | HII\_IMAGE\_PROTOCOL.GetImage - GetImage() returns EFI\_NOT\_FOUND with an invalid PackageList. | Call GetImage() with an invalid PackageList. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.2.6 | 0x05fc7f10, 0xe1ef, 0x4fd0, 0x91, 0x3d, 0x86, 0x46, 0x53, 0x7e, 0x4c, 0xbd | HII\_IMAGE\_PROTOCOL.GetImage - GetImage() returns EFI\_NOT\_FOUND with PackageList been NULL. | Call GetImage() with PackageList being NULL. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.2.7 | 0x14cd0647, 0x3fd7, 0x4831, 0x9e, 0xa5, 0x9b, 0x3d, 0xd7, 0xc8, 0xeb, 0xb7 | HII\_IMAGE\_PROTOCOL.GetImage - GetImage() returns EFI\_SUCCESS with valid parameters and the result checked. | Call GetImage() with valid parameters, The return status should be EFI\_SUCCESS. |

### SetImage()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.3.3.1 | 0x9af36ab7, 0x8bd2, 0x417b, 0xa5, 0x10, 0x1f, 0x22, 0x99, 0x13, 0x72, 0x64 | HII\_IMAGE\_PROTOCOL.SetImage - SetImage() returns EFI\_NOT\_FOUND with ImageId been invalid | Call SetImage() with an invalid ImageId which is not in the database. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.3.2 | 0x5d9b72d9, 0x01f4, 0x47cd, 0x96, 0xbb, 0xb1, 0xf2, 0xf2, 0x1f, 0xf7, 0x2a | HII\_IMAGE\_PROTOCOL.SetImage - SetImage() returns EFI\_INVALID\_PARAMETER with Image been NULL. | Call SetImage() with Image being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.3.3 | 0xa411c5ef, 0x0eeb, 0x4a9a, 0x85, 0x9a, 0x4a, 0x64, 0x0d, 0xa6, 0x16, 0xf7 | HII\_IMAGE\_PROTOCOL.SetImage - SetImage() returns EFI\_NOT\_FOUND with an invalid PackageList. | Call SetImage() with an invalid PackageList. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.3.4 | 0x870c9c4c, 0xe099, 0x4024, 0xac, 0x3a, 0x7b, 0x8c, 0x30, 0x98, 0x8c, 0x2e | HII\_IMAGE\_PROTOCOL.SetImage - SetImage() returns EFI\_NOT\_FOUND with PackageList been NULL. | Call SetImage() with PackageList being NULL. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.3.5 | 0xc99ad1a4, 0x3f5b, 0x46dc, 0xb4, 0x85, 0xb2, 0x23, 0x9d, 0xef, 0xbc, 0x2c | HII\_IMAGE\_PROTOCOL.SetImage - SetImage() returns EFI\_SUCCESS with valid parameters and result checked. | Call SetImage() with valid parameters, The return status should be EFI\_SUCCESS. |

### DrawImage()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.3.4.1 | 0x4bb8ee94, 0x8a57, 0x470f, 0x9d, 0xd5, 0xef, 0x81, 0xea, 0xd9, 0xd6, 0xad | HII\_IMAGE\_PROTOCOL.DrawImage - DrawImage() returns EFI\_INVALID\_PARAMETER with Image been NULL. | Call DrawImage() with Image being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.4.2 | 0xd9034d5d, 0xde07, 0x4458,  0x92, 0xb7, 0x4c, 0xd1, 0x50, 0x1c, 0xe8, 0x90 | HII\_IMAGE\_PROTOCOL.DrawImage - DrawImage() returns EFI\_INVALID\_PARAMETER with EFI\_HII\_DRAW\_FLAG\_FORCE\_TRANS and Blt been NULL. | Call DrawImage() with Flags being EFI\_HII\_DRAW\_FLAG\_FORCE\_TRANS and Blt being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.4.3 | 0x5c232904, 0x23f8, 0x4b0f, 0x9c, 0x85, 0xb7, 0xe8, 0xa5, 0xc9, 0x80, 0x05 | HII\_IMAGE\_PROTOCOL.DrawImage - DrawImage() returns EFI\_INVALID\_PARAMETER with EFI\_HII\_DIRECT\_TO\_SCREEN and no screen. | Call DrawIamge() with Flags being EFI\_HII\_DIRECT\_TO\_SCREEN and use EFI\_GRAPHICS\_OUTPUT\_BLT\_PIXEL structure in EFI\_IMAGE\_OUTPUT structure. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.4.4 | 0xf9e86ff1, 0x611c, 0x41b8,  0xb0, 0x8d, 0x2a, 0xe2, 0x5e, 0x34, 0x2a, 0x1d | HII\_IMAGE\_PROTOCOL.DrawImage - DrawImage() returns EFI\_INVALID\_PARAMETER with EFI\_HII\_DRAW\_FLAG\_CLIP and Blt points to NULL. | Call DrawImage() with Flags being EFI\_HII\_DRAW\_FLAG\_CLIP and Blt being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.4.5 | 0x11ae81e8, 0xfe20, 0x472d, 0x8c, 0xdb, 0x40, 0xb7, 0x56, 0x09, 0xd9, 0xdc | HII\_IMAGE\_PROTOCOL.DrawImage - DrawImage() returns EFI\_INVALID\_PARAMETER with EFI\_HII\_DRAW\_FLAG\_DEFAULT, Blt points to NULL and Image->Flags is EFI\_IMAGE\_TRANSPARENT. | Call DrawImage() with Blt being NULL, Flags being EFI\_HII\_DRAW\_FLAG\_DEFAULT and Image->Flags being EFI\_IMAGE\_TRANSPARENT. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.4.6 | 0x6e409e86, 0x16d3, 0x4b31, 0x96, 0x71, 0xf9, 0x2c, 0xe6, 0x26, 0x1b, 0xcf | HII\_IMAGE\_PROTOCOL.DrawImage - DrawImage() returns EFI\_SUCCESS with valid parameter. | Call DrawImage() with Flags being EFI\_HII\_DRAW\_FLAG\_FORCE\_OPAQUE, Blt being NULL and other valid parameters. The return status should be EFI\_SUCCESS. |
| 5.18.3.4.7 | 0xedbef6eb, 0xf68f, 0x4154, 0xb0, 0x12, 0xb9, 0xd7, 0x55, 0x3b, 0xa6, 0x0a | HII\_IMAGE\_PROTOCOL.DrawImage - DrawImage() returns EFI\_SUCCESS with valid parameter. | Call DrawImage() with Flags being valid combination, Blt being not NULL and other valid parameters. The return status should be EFI\_SUCCESS. |

### DrawImageId()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.3.5.1 | 0xcb1936c7, 0x53c7, 0x4a65, 0xa5, 0x3d, 0x85, 0xc2, 0x35, 0x72, 0xff, 0x29 | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImageId() returns EFI\_NOT\_FOUND with an invalid PackageList. | Call DrawImageId() with an invalid PackageList. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.5.2 | 0xb1372c26, 0x3de4, 0x4a5c, 0x8a, 0x1f, 0x71, 0x4a, 0x7b, 0x07, 0x0e, 0x67 | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImageId() returns EFI\_NOT\_FOUND with PackageList been NULL. | Call DrawImageId() with PackageList being NULL. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.5.3 | 0x23a7fcfd, 0x4d0f, 0x4460, 0xb8, 0xcc, 0x7a, 0xfa, 0xf7, 0x4d, 0xe5, 0xaa | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImageId() returns EFI\_NOT\_FOUND with invalid ImageId. | Call DrawImageId() with an invalid ImageId which is not in the specified PackageList. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.5.4 | 0x5433fcf6, 0x06f4, 0x45f3,  0x91, 0x23, 0x79, 0x5f, 0x49, 0x69, 0x77, 0x4d | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImageId() returns EFI\_NOT\_FOUND with invalid PackageList. | Call DrawImageId() with an invalid PackageList which is not in the database. The return status should be EFI\_NOT\_FOUND. |
| 5.18.3.5.5 | 0x2df19349, 0xec8c, 0x42f7,  0x9f, 0x8e, 0x1d, 0x56, 0x13, 0x6c, 0x95, 0xbc | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImage() returns EFI\_INVALID\_PARAMETER with EFI\_HII\_DRAW\_FLAG\_FORCE\_TRANS and Blt been NULL. | Call DrawImageId() with Flags being EFI\_HII\_DRAW\_FLAG\_FORCE\_TRANS and Blt being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.5.6 | 0x059732f0, 0x431e, 0x4ad3, 0x92, 0xa0, 0x4b, 0xda, 0xaa, 0x8d, 0x98, 0x92 | HII\_IMAGE\_PROTOCOL.DrawImage - DrawImage() returns EFI\_INVALID\_PARAMETER with EFI\_HII\_DIRECT\_TO\_SCREEN and no screen. | Call DrawIamgeId() with Flags being EFI\_HII\_DIRECT\_TO\_SCREEN and use EFI\_GRAPHICS\_OUTPUT\_BLT\_PIXEL structure in EFI\_IMAGE\_OUTPUT structure. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.5.7 | 0xd12320fa, 0x063e, 0x48e3, 0x85, 0xd5, 0x1c, 0x9b, 0x7c, 0x48, 0x71, 0x13 | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImageId() returns EFI\_INVALID\_PARAMETER with EFI\_HII\_DRAW\_FLAG\_CLIP and Blt points to NULL. | Call DrawImageId() with Flags being EFI\_HII\_DRAW\_FLAG\_CLIP and Blt being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.5.8 | 0xb3e326cb, 0x67bc, 0x49a7, 0x8c, 0xb6, 0xc3, 0xec, 0x3b, 0x83, 0x20, 0x1e | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImageId() returns EFI\_INVALID\_PARAMETER with EFI\_HII\_DRAW\_FLAG\_DEFAULT, Blt points to NULL and Image->Flags is EFI\_IMAGE\_TRANSPARENT. | Call DrawImageId() with Blt being NULL, Flags being EFI\_HII\_DRAW\_FLAG\_DEFAULT and Image->Flags being EFI\_IMAGE\_TRANSPARENT. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.3.5.9 | 0xaeeb761e, 0x1b38, 0x4b06, 0x8d, 0x26, 0xf3, 0x6f, 0xde, 0xa4, 0x3d, 0x88 | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImageId() returns EFI\_SUCCESS with valid parameter. | Call DrawImageId() with Flags being EFI\_HII\_DRAW\_FLAG\_FORCE\_OPAQUE, Blt being NULL and other valid parameters. The return status should be EFI\_SUCCESS. |
| 5.18.3.5.10 | 0x2b844dec, 0xc8cf, 0x442c, 0x89, 0xc0, 0x9f, 0x44, 0xe0, 0x96, 0x4b, 0xcb | HII\_IMAGE\_PROTOCOL.DrawImageId - DrawImageId() returns EFI\_SUCCESS with valid parameter. | Call DrawImage() with Flags being valid combination, Blt being not NULL and other valid parameters. The return status should be EFI\_SUCCESS. |

## EFI\_HII\_DATABASE\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_HII\_DATABASE\_PROTOCOL Section.

### NewPackageList()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.1.1 | 0x17364518, 0x35c4, 0x481a, 0x82, 0x45, 0xdd, 0x8b, 0x85, 0xbf, 0x01, 0x7c | HII\_DATABASE\_PROTOCOL.NewPackageList - NewPackageList() returns EFI\_INVALID\_PARAMETER with PackageList being NULL. | Call NewPackageList() with PackageList being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.1.2 | 0xd12127b3, 0x3a61, 0x498d, 0xbb, 0x8f, 0x9f, 0x9e, 0xb3, 0x9a, 0xfd, 0x95 | HII\_DATABASE\_PROTOCOL.NewPackageList - NewPackageList() returns EFI\_INVALID\_PARAMETER with Handle being NULL. | Call NewPackageList() with Handle being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.1.3 | 0x3ee6503d, 0x5fab, 0x4f51, 0x9a, 0xee, 0xc9, 0x0f, 0x9d, 0x73, 0xe5, 0xd7 | HII\_DATABASE\_PROTOCOL.NewPackageList - NewPackageList() returns EFI\_SUCCESS with valid inputs | Call NewPackageList() with valid parameters. The return status should be EFI\_SUCCESS. |

### RemovePackageList ()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.2.1 | 0x244e5792, 0x471b, 0x456b, 0x8b, 0xfe, 0x1f, 0x68, 0xeb, 0x8f, 0xcd, 0xd0 | HII\_DATABASE\_PROTOCOL.RemovePackageList - RemovePackageList() returns EFI\_NOT\_FOUND with Handle being NULL. | Call RemovePackageList() with Handle being NULL. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.2.2 | 0x4f2588b4, 0xadb6, 0x48ba, 0xac, 0x53, 0x97, 0x3e, 0x05, 0x64, 0x5d, 0x4f | HII\_DATABASE\_PROTOCOL.RemovePackageList - RemovePackageList() returns EFI\_NOT\_FOUND with Handle has already been removed once. | Call RemovePackageList() with Handle which has been removed. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.2.3 | 0x28c05503, 0x33ce, 0x41ae, 0x90, 0x2e, 0xbc, 0x34, 0xe0, 0xb8, 0x0e, 0x9d | HII\_DATABASE\_PROTOCOL.RemovePackageList - RemovePackageList() returns EFI\_NOT\_FOUND with an invalid Handle. | Call RemovePackageList() with an invalid Handle. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.2.4 | 0xb4a3a9ac, 0x0dfa, 0x4025, 0xa6, 0x36, 0xac, 0x53, 0x19, 0x7a, 0x5e, 0xd2 | HII\_DATABASE\_PROTOCOL.RemovePackageList - RemovePackageList() returns EFI\_SUCCESS with valid inputs. | Part1: Call RemovePackageList() with valid parameters. The return status should be EFI\_SUCCESS. |
| 5.18.4.2.5 | 0xad310e29, 0x2112, 0x485b, 0xa4, 0xdc, 0xc8, 0xec, 0xf8, 0x49, 0x7b, 0xc9 | HII\_DATABASE\_PROTOCOL.RemovePackageList - ExportPackageLists() returns EFI\_NOT\_FOUND when RemovePackageList work. | Part2: Call ExportPackageList() with Handle which has been removed. The return status should be EFI\_NOT\_FOUND. |

### UpdatePackageList ()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.3.1 | 0xb4bf4c19, 0x64cc, 0x4efe, 0xa7, 0x21, 0x3f, 0xc2, 0x07, 0x88, 0x51, 0xb4 | HII\_DATABASE\_PROTOCOL.UpdatePackageList - UpdatePackageList() returns EFI\_NOT\_FOUND with Handle being NULL. | Call UpdatePackageList() with Handle being NULL. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.3.2 | 0xcd591535, 0x7df7, 0x4f99, 0x9d, 0x13, 0x3b, 0x8e, 0x39, 0x85, 0x39, 0x6f | HII\_DATABASE\_PROTOCOL.UpdatePackageList - UpdatePackageList() returns EFI\_NOT\_FOUND with Handle has already been removed before. | Call UpdatePackageList() with Handle which has been removed. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.3.3 | 0x2a2f8bf0, 0x0c27, 0x41f3, 0xae, 0x19, 0xb0, 0x66, 0x16, 0x92, 0x5c, 0x0b | HII\_DATABASE\_PROTOCOL.UpdatePackageList - UpdatePackageList() returns EFI\_NOT\_FOUND with an invalid handle. | Call UpdatePackageList() with an invalid Handle. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.3.4 | 0xfcb45969, 0x37f8, 0x430e, 0x86, 0x99, 0x7f, 0x89, 0xde, 0x52, 0x6f, 0x94 | HII\_DATABASE\_PROTOCOL.UpdatePackageList - UpdatePackageList() returns EFI\_INVALID\_PARAMETER with PackageList been NULL. | Call UpdatePackageList() with PackageList being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.3.5 | 0xe1f18c0b, 0xfa2f, 0x488a, 0x80, 0x25, 0x77, 0x35, 0x49, 0x55, 0x36, 0xe0 | HII\_DATABASE\_PROTOCOL.UpdatePackageList - UpdatePackageList() returns EFI\_SUCCESS with valid inputs | Call UpdatePackageList() with valid parameters. The return status should be EFI\_SUCCESS. |

### ListPackageLists()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.4.1 | 0x7b5c4246, 0xe6b3, 0x4eb0, 0xaf, 0xc4, 0x23, 0xb1, 0xbf, 0xfd, 0x46, 0x39 | HII\_DATABASE\_PROTOCOL.ListPackageLists - ListPackageLists() returns EFI\_INVALID\_PARAMETER with Handle being NULL. | Call ListPackageList() with Handle being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.4.2 | 0x9268a2d0, 0xc922, 0x42bc, 0xb0, 0x5d, 0x3d, 0x18, 0xab, 0xf2, 0xe9, 0x37 | HII\_DATABASE\_PROTOCOL.ListPackageLists - ListPackageLists() returns EFI\_INVALID\_PARAMETER with HandleBufferLength being NULL. | Call ListPackageList() with HandleBufferLength being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.4.3 | 0x7c577327, 0x562c, 0x4333, 0x9b, 0x81, 0x9b, 0xf6, 0xf2, 0x80, 0x83, 0xec | HII\_DATABASE\_PROTOCOL.ListPackageLists - ListPackageLists() returns EFI\_NOT\_FOUND when no matching handles were found. | Call ListPackageList() with no match Handle being found. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.4.4 | 0xf5771b8e, 0x6db5, 0x473d, 0xba, 0x32, 0x21, 0xfe, 0xf2, 0x7f, 0x05, 0xf2 | HII\_DATABASE\_PROTOCOL.ListPackageLists - ListPackageLists() returns EFI\_BUFFER\_TOO\_SMALL when the HandleBufferLength indicates the buffer is too small. | Part1: Call ListPackageList() with HandleBufferLength which indicates the Handle buffer is small. The return status should EFI\_BUFFER\_TOO\_SMALL. |
| 5.18.4.4.5 | 0x08c276ef, 0x185c, 0x4eac, 0xbe, 0x84, 0x7d, 0xb0, 0x8c, 0x38, 0x5f, 0xe7 | HII\_DATABASE\_PROTOCOL.ListPackageLists - ListPackageLists() returns EFI\_BUFFER\_TOO\_SMALL when the HandleBufferLength indicates the buffer is too small and return the needed buffer length. | Part 2: The HandleBufferLength is updated with the required size. |
| 5.18.4.4.6 | 0x212bb7e2, 0xa998, 0x4ede, 0xba, 0x08, 0x8d, 0x8c, 0xda, 0x9d, 0xb7, 0xd4 | HII\_DATABASE\_PROTOCOL.ListPackageLists - ListPackageLists() returns EFI\_INVALID\_PARAMETER with PackageType is not Guid and PackageGuid is not NULL. | Call ListPackageList() with no Guid PackageType and no NULL PackageGuid. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.4.7 | 0x9b711922, 0x06d3, 0x4ba4, 0x98, 0x5b, 0x50, 0x72, 0x46, 0x94, 0x8b, 0xb2 | HII\_DATABASE\_PROTOCOL.ListPackageLists - ListPackageLists() returns EFI\_INVALID\_PARAMETER with PackageType is EFI\_HII\_DATABASE\_TYPE\_GUID and PackageGuid is NULL. | Call ListPackageList() with Guid PackageType and PackageGuid being NULL. The return status should EFI\_INVALID\_PARAMETER. |
| 5.18.4.4.8 | 0x1dd024a0, 0xc53b, 0x439e, 0x86, 0x43, 0xc3, 0xe2, 0x82, 0x1f, 0x34, 0x75 | HII\_DATABASE\_PROTOCOL.ListPackageLists - ListPackageLists() returns EFI\_SUCCESS with valid inputs and return length checked. | Call ListPackageList() with valid parameters. The return status should be EFI\_SUCCESS. |

### ExportPackageLists ()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.5.1 | 0xdc1afed1, 0x5be4, 0x4488, 0xaf, 0xeb, 0x75, 0x70, 0xb6, 0x3d, 0xea, 0xc4 | HII\_DATABASE\_PROTOCOL.ExportPackageLists - ExportPackageLists() returns EFI\_INVALID\_PARAMETER with BufferSize being NULL. | Call ExportPackageList() with BufferSize being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.5.2 | 0xd25ed0fa, 0xe829, 0x4e68, 0xbb, 0xa3, 0xef, 0x82, 0x5a, 0xa0, 0xba, 0x85 | HII\_DATABASE\_PROTOCOL.ExportPackageLists - ExportPackageLists() returns EFI\_INVALID\_PARAMETER with Buffer being NULL | Call ExportPackageList() with Buffer being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.5.3 | 0x0462bf1f, 0xce31, 0x4314, 0xbd, 0x34, 0x40, 0x4a, 0x05, 0x04, 0xd3, 0x0c | HII\_DATABASE\_PROTOCOL.ExportPackageLists - ExportPackageLists() returns EFI\_BUFFER\_TOO\_SMALL with BufferSize indicates the buffer is too small. | Part1: Call ExportPackageList() with BufferSize which indicates the Buffer is small. The return status should EFI\_BUFFER\_TOO\_SMALL. |
| 5.18.4.5.4 | 0xf03af69e, 0x3bba, 0x4092, 0xb0, 0x40, 0x75, 0x4b, 0x42, 0x6b, 0x2f, 0xd0 | HII\_DATABASE\_PROTOCOL.ExportPackageLists - ExportPackageLists() returns EFI\_BUFFER\_TOO\_SMALL with BufferSize indicates the buffer is too small and return the needed BufferSize. | Part2: The BufferSize is updated with the required size. |
| 5.18.4.5.5 | 0x55ce12c1, 0x35eb, 0x4d8c, 0xbf, 0xd9, 0x9b, 0x0c, 0x52, 0x4d, 0xc0, 0x76 | HII\_DATABASE\_PROTOCOL.ExportPackageLists - ExportPackageLists() returns EFI\_NOT\_FOUND with handle has been already removed once. | Call ExportPackageList() with Handle which has been removed once. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.5.6 | 0x22a02d74, 0xc2a8, 0x439f, 0xbd, 0x4c, 0xf6, 0xb0, 0x1a, 0xbe, 0x03, 0xe4 | HII\_DATABASE\_PROTOCOL.ExportPackageLists - ExportPackageLists() returns EFI\_NOT\_FOUND with the invalid handle. | Call ExportPackageList() with an invalid Handle. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.5.7 | 0xc9741024, 0x3073, 0x4827,  0x92, 0x23, 0x06, 0x33, 0x96, 0x0b, 0x8d, 0x6d | HII\_DATABASE\_PROTOCOL.ExportPackageLists - ExportPackageLists() returns EFI\_SUCCESS with valid inputs and result checked. | Call ExportPackageList() with valid parameters. The return status should be EFI\_SUCCESS. |

### RegisterPackageNotify()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.6.1 | 0x1665f366, 0x70af, 0x4348, 0xbb, 0xc8, 0xb1, 0xaf, 0x38, 0xe1, 0x2d, 0xfd | HII\_DATABASE\_PROTOCOL.RegisterPackageNotify - RegisterPackageNotify() returns EFI\_INVALID\_PARAMETER with NotifyHandle been NULL. | Call RegisterPackageNotify() with NotifyHandle being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.6.2 | 0x7541d67b, 0xe837, 0x46bf, 0x85, 0x7e, 0xbc, 0x22, 0xf2, 0xe1, 0x0d, 0x60 | HII\_DATABASE\_PROTOCOL.RegisterPackageNotify - RegisterPackageNotify() returns EFI\_INVALID\_PARAMETER with PackageType is not Guid and PackageGuid not been NULL. | Call RegisterPackageNotify() with no Guid PackageType and no NULL PackageGuid. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.6.3 | 0x625abf38, 0x2d02, 0x46b2, 0xae, 0xa9, 0xcc, 0x5b, 0x0c, 0x83, 0xf1, 0x69 | HII\_DATABASE\_PROTOCOL.RegisterPackageNotify - RegisterPackageNotify() returns EFI\_INVALID\_PARAMETER with PackageType is EFI\_HII\_PACKAGE\_TYPE\_GUID and PackageGuid been NULL. | Call RegisterPackageNotify() with Guid PackageType and NULL PackageGuid. The return status should be EFI\_INVALID\_PARAMETER. |

### UnregisterPackageNotify()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.7.1 | 0xef67f1ff, 0x9b53, 0x40ac, 0x8e, 0xec, 0xca, 0x5c, 0x59, 0xfd, 0xbd, 0x0d | HII\_DATABASE\_PROTOCOL.UnregisterPackageNotify - UnregisterPackageNotify() returns EFI\_NOT\_FOUND with the NotifyHandle has been removed already. | Call UnRegisterPackageNotify() with NotifyHandle which has been removed once. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.7.2 | 0xc5266e09, 0xe5e8, 0x4c85, 0xb3, 0x0a, 0xc9, 0x83, 0x04, 0x4f, 0x23, 0xfc | HII\_DATABASE\_PROTOCOL.UnregisterPackageNotify - UnregisterPackageNotify() returns EFI\_NOT\_FOUND with an invalid NotifyHandle. | Call UnRegisterPackageNotify() with NotifyHandle which can’t be found in the database. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.7.3 | 0x51c64bb1, 0x3266, 0x4ccd, 0x82, 0xde, 0xed, 0x6b, 0xa7, 0x68, 0x35, 0xe5 | HII\_DATABASE\_PROTOCOL.UnregisterPackageNotify - UnregisterPackageNotify() returns EFI\_NOT\_FOUND with NotifyHandle been NULL. | Call UnRegisterPackageNotify() with NotifyHandle being NULL. The return status should be EFI\_NOT\_FOUND. |

### FindKeyboardLayouts()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.8.1 | 0xa61bf4b2, 0xb1e0, 0x4e62, 0x95, 0x2d, 0xa0, 0x68, 0x98, 0x48, 0x06, 0xb2 | HII\_DATABASE\_PROTOCOL.FindKeyboardLayouts - FindKeyboardLayouts() returns EFI\_INVALID\_PARAMETER with KeyGuidBufferLength been NULL. | Call FindKeyboardLayouts() with KeyGuidBufferLength being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.8.2 | 0x1ea6e881, 0x6f47, 0x4fdc, 0x8b, 0x8c, 0xba, 0x33, 0x9a, 0x13, 0xbe, 0xc0 | HII\_DATABASE\_PROTOCOL.FindKeyboardLayouts - FindKeyboardLayouts() returns EFI\_INVALID\_PARAMETER with KeyGuidBuffer been NULL. | Call FindKeyboardLayouts() with KeyGuidBuffer being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.8.3 | 0xc3bacca3, 0x901a, 0x49ad, 0xa9, 0x86, 0x41, 0x62, 0xff, 0xb3, 0xa1, 0x8f | HII\_DATABASE\_PROTOCOL.FindKeyboardLayouts - FindKeyboardLayouts() returns EFI\_BUFFER\_TOO\_SMALL with KeyGuidBufferLength indicates the buffer is too small. | Call FindKeyboardLayouts() with KeyGuidBufferLength which indicates KeyGuidBuffer is small. The return status should be EFI\_BUFFER\_TOO\_SMALL. The KeyGuidBufferLength should be updated with required length. |
| 5.18.4.8.4 | 0x1dc41f45, 0x9e3a, 0x41e2, 0x8f, 0x99, 0x8d, 0x4d, 0x39, 0x32, 0x12, 0x85 | HII\_DATABASE\_PROTOCOL.FindKeyboardLayouts - FindKeyboardLayouts() returns EFI\_SUCCESS with valid inputs. | Call FindKeyboardLayouts() with valid parameters. The return status should be EFI\_SUCCESS. The KeyGuidBufferLength should be updated with actual length. |

### GetKeyboardLayout()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.9.1 | 0xbc4b4ea1, 0x069c, 0x459c, 0x8c, 0x22, 0x68, 0x19, 0x01, 0x71, 0x78, 0x48 | HII\_DATABASE\_PROTOCOL.GetKeyboardLayout - GetKeyboardLayout() returns EFI\_INVALID\_PARAMETER with KeyboardLayoutLength been NULL. | Call GetKeyboardLayout() with KeyboardLayoutLength being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.9.2 | 0xe2aeca1e, 0x5c50, 0x4ee7, 0x8f, 0x69, 0x46, 0xa7, 0xb9, 0x01, 0x3e, 0x0d | HII\_DATABASE\_PROTOCOL.GetKeyboardLayout - GetKeyboardLayout() returns EFI\_INVALID\_PARAMETER with KeyboardLayout been NULL. | Call GetKeyboardLayout() with KeyboardLayout being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.9.3 | 0x0d41d349, 0xe5f4, 0x43d5, 0x85, 0x0e, 0xfe, 0x4f, 0x08, 0x5a, 0xbf, 0xb2 | HII\_DATABASE\_PROTOCOL.GetKeyboardLayout - GetKeyboardLayout() returns EFI\_BUFFER\_TOO\_SMALL with KeyboardLayoutLength not enough. | Call GetKeyboardLayout() with KeyboardLayoutLength which indicates KeyboardLayout is small. The return status should be EFI\_BUFFER\_TOO\_SMALL. The KeyboardLayoutLength should be updated with required length. |
| 5.18.4.9.4 | 0xc2732202, 0x48ca, 0x49f8, 0xbb, 0x18, 0xd3, 0x6c, 0xe1, 0xb4, 0x83, 0xfa | HII\_DATABASE\_PROTOCOL.GetKeyboardLayout - GetKeyboardLayout() returns EFI\_NOT\_FOUND with the requested keyboard layout not found. | Call GetKeyboardLayout() with a Guid which can’t be found in the database. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.9.5 | 0x4ffc59ee, 0xefb8, 0x4533, 0x81, 0x4f, 0x85, 0xed, 0x90, 0x93, 0x44, 0xc7 | HII\_DATABASE\_PROTOCOL.GetKeyboardLayout - GetKeyboardLayout() returns EFI\_SUCCESS with valid inputs. | Call GetKeyboardLayout() with valid parameters. The return status should be EFI\_SUCCESS. |

### SetKeyboardLayout()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.10.1 | 0xad8c6cdc, 0xc749, 0x42e6, 0x88, 0xf7, 0x73, 0x44, 0x7c, 0x38, 0x9e, 0x4d | HII\_DATABASE\_PROTOCOL.SetKeyboardLayout - SetKeyboardLayout() returns EFI\_INVALID\_PARAMETER with KeyGuid set to be NULL. | Call SetKeyboardLayout() with KeyGuid being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.10.2 | 0x07018fe1, 0xdde0, 0x449b, 0xa5, 0xe2, 0xb1, 0x7a, 0xb5, 0x68, 0x7c, 0x97 | HII\_DATABASE\_PROTOCOL.SetKeyboardLayout - SetKeyboardLayout() returns EFI\_NOT\_FOUND with the referenced keyboard layout not found. | Call SetKeyboardLayout() with KeyGuid which can’t be found in database. The return status should be EFI\_NOT\_FOUND. |
| 5.18.4.10.3 | 0xe7a3dffa, 0x4cca, 0x4402,  0x8f, 0xf1, 0xe3, 0xf3, 0x16, 0xf5, 0x45, 0x1f | HII\_DATABASE\_PROTOCOL.SetKeyboardLayout - SetKeyboardLayout() returns EFI\_SUCCESS with valid inputs. | Call SetKeyboardLayout() with valid parameters. The return status should be EFI\_SUCCESS. |

### GetPackageListHandle()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.4.11.1 | 0x373b128d, 0x2216, 0x415b, 0xbb, 0xb1, 0x99, 0x0e, 0xe3, 0x79, 0xf2, 0x85 | HII\_DATABASE\_PROTOCOL.GetPackageListHandle - GetPackageListHandle() returns EFI\_INVALID\_PARAMETER with DriverHandle been NULL. | Call GetPackageListHandle () with DriverHandle being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.11.2 | 0xb50cffb8, 0x7b74, 0x4b93, 0xb4, 0x87, 0xb3, 0x39, 0xf4, 0x7e, 0xa6, 0x25 | HII\_DATABASE\_PROTOCOL.GetPackageListHandle - GetPackageListHandle () returns EFI\_INVALID\_PARAMETER with a PackageListHandle which has been removed. | Call GetPackageListHandle () with a PackageListHandle which has been removed. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.11.3 | 0x27a38687, 0x398a, 0x4d65, 0xab, 0x7b, 0x4d, 0xf2, 0xd1, 0x1f, 0x21, 0xa0 | HII\_DATABASE\_PROTOCOL.GetPackageListHandle - GetPackageListHandle() returns EFI\_INVALID\_PARAMETER with an invalid PackageListHandle. | Call GetPackageListHandle ()with an invalid PackageListHandle. The return status should be EFI\_INVALID\_PARAMETER . |
| 5.18.4.11.4 | 0x2bc2dae8, 0x2692, 0x487a, 0x94, 0x9d, 0xa7, 0x45, 0x08, 0x82, 0x65, 0x11 | HII\_DATABASE\_PROTOCOL. GetPackageListHandle - GetPackageListHandle() returns EFI\_INVALID\_PARAMETER with PackageListHandle being NULL . | Call GetPackageListHandle ()with PackageListHandle being NULL. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.4.11.5 | 0xa81329db, 0xcc91, 0x491c, 0xb1, 0x2a, 0x44, 0x0d, 0xf7, 0xed, 0x77, 0xc6 | HII\_DATABASE\_PROTOCOL. GetPackageListHandle - GetPackageListHandle() returns EFI\_SUCCESS with valid inputs. | Call GetPackageListHandle () with valid parameters. The return status should be EFI\_SUCCESS. |

## EFI\_HII\_CONFIG\_ROUTING\_PROTOCOL Test

Reference Document:

*UEFI Specification,* EFI\_HII\_CONFIG\_ROUTING\_PROTOCOL Section.

### ExtractConfig()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.5.1.1 | 0x04697ed6, 0xcb4e, 0x4e02, 0xbb, 0x8e, 0x9b, 0x76, 0x0b, 0x90, 0xe2, 0xcd | HII\_CONFIG\_ROUTING\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_INVALID\_PARAMETER with Request been NULL. | Call ExtractConfig() with valid parameters except Request being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.1.2 | 0x4a1e3525, 0x5247, 0x40dc, 0x93, 0xf7, 0x81, 0x30, 0x6a, 0xce, 0x20, 0xb5 | HII\_CONFIG\_ROUTING\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_INVALID\_PARAMETER with Progress been NULL. | Call ExtractConfig() with valid parameters except Progress being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.1.3 | 0x05b967d0, 0xe19d, 0x46d8, 0x87, 0xd8, 0x7d, 0x29, 0x65, 0x53, 0x61, 0xc7 | HII\_CONFIG\_ROUTING\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_INVALID\_PARAMETER with Results been NULL. | Call ExtractConfig() with valid parameters except Results being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.1.4 | 0xee200b58, 0x3714, 0x4cb6, 0x91, 0xc6, 0x31, 0xbe, 0xbd, 0xf4, 0x64, 0x96 | HII\_CONFIG\_ROUTING\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_NOT\_FOUND if Routing data doesn’t match any known driver. | Call ExtractConfig() with an invalid Request. The ConfigHdr of Request can’t be found in current system. The return status should be EFI\_NOT\_FOUND. Progress should be set to the “G” in the “GUID” of the routing header that doesn’t match. |
| 5.18.5.1.5 | 0xa18aebb6, 0x140f, 0x454f, 0x8f, 0xe5, 0x34, 0xdd, 0x38, 0xd8, 0xb0, 0xf0 | HII\_CONFIG\_ROUTING\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_INVALID\_PARAMETER if name in Request can’t match any known driver. | Call ExtractConfig() with an invalid Request. The name in Request can’t be found in current system. The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.1.6 | 0x67adfcdd, 0xda46, 0x4eb8, 0x82, 0x9d, 0xa4, 0x92, 0x8c, 0x10, 0xba, 0x68 | HII\_CONFIG\_ROUTING\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_SUCCESS with valid parameter and Progress points to the Request’s NULL terminator. | Call ExtractConfig() with valid parameters. The return status should be EFI\_SUCCESS and Progress points to the Request’s NULL terminator. |
| 5.18.5.1.7 | 0xf91ef5f3, 0xe0c6, 0x4aca, 0xa0, 0xd0, 0x5, 0xf9, 0xb1, 0x6a, 0x13, 0xbd | HII\_CONFIG\_ROUTING\_PROT OCOL.ExtractConfig - ExtractConfig() returns EFI\_SUCCESS & Check if Results is in <MultiConfigAltResp> format | 1.Call ExtractConfig() with valid parameters.  2.Check if Results is in <MultiConfigAltResp> format. The return status should be EFI\_SUCCESS |

### ExportConfig()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.5.2.1 | 0x81f9658b, 0xbae2, 0x4e08, 0x87, 0xe3, 0x75, 0xe4, 0xe1, 0x47, 0x13, 0xba | HII\_CONFIG\_ROUTING\_PROTOCOL.ExportConfig - ExportConfig() returns EFI\_INVALID\_PARAMETER with Request been NULL. | Call ExportConfig() with Request being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.2.2 | 0xe23425ee, 0xaa38, 0x4074, 0xa1, 0xaa, 0xad, 0x5d, 0x98, 0x5a, 0x34, 0xe4 | HII\_CONFIG\_ROUTING\_PROTOCOL.ExportConfig - ExportConfig () returns EFI\_SUCCESS with valid parameter. | Call ExportConfig() with valid parameter, The return status should be EFI\_SUCCESS. |

### RouteConfig()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.5.3.1 | 0x3a5c09d6, 0x0532, 0x4b4d, 0x87, 0xc8, 0x5e, 0x20, 0x33, 0x78, 0xbc, 0x3f | HII\_CONFIG\_ROUTING\_PROTOCOL.RouteConfig - RouteConfig() returns EFI\_INVALID\_PARAMETER with Configuration been NULL. | Call RouteConfig() with Configuration being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.3.2 | 0x18cdf3f8, 0xf9e6, 0x4128, 0xa4, 0xa6, 0x88, 0xea, 0x88, 0x5d, 0x59, 0x7c | HII\_CONFIG\_ROUTING\_PROTOCOL.RouteConfig - RouteConfig() returns EFI\_NOT\_FOUND if Routing data was not found. | Call RouteConfig() with an invalid Configuration. The ConfigHdr of Configuration can’t be found in current system. The return status should be EFI\_NOT\_FOUND. |
| 5.18.5.3.3 | 0x20833aeb, 0x9ff1, 0x4315, 0xb1, 0x0f, 0x31, 0x7c, 0x7b, 0x92, 0x45, 0x21 | HII\_CONFIG\_ROUTING\_PROTOCOL.RouteConfig - RouteConfig () returns EFI\_SUCCESS with valid parameter and Progresspoints to theConfiguration’s NULL terminator. | Call RouteConfig() with valid parameters. The return status should be EFI\_SUCCESS and Progresspoints to theConfiguration’s NULL terminator. |

### BlockToConfig()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.5.4.1 | 0xb1dfee09, 0x73e5, 0x4659, 0x9a, 0xc6, 0x59, 0x46, 0xc1, 0xa1, 0x53, 0xcb | HII\_CONFIG\_ROUTING\_PROTOCOL.BlockToConfig - BlockToConfig() returns EFI\_INVALID\_PARAMETER with ConfigRequest been NULL. | Call BlockToConfig() with valid parameters except ConfigRequest being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.4.2 | 0x544bf56b, 0x3bdc, 0x46d5, 0x88, 0x4f, 0x19, 0xde, 0x76, 0x19, 0xef, 0xd3 | HII\_CONFIG\_ROUTING\_PROTOCOL.BlockToConfig - BlockToConfig() returns EFI\_INVALID\_PARAMETER with Block been NULL. | Call BlockToConfig() with valid parameters except Block being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.4.3 | 0xee6f8fd8, 0x951d, 0x4976, 0x86, 0xf0, 0xae, 0x7f, 0x5c, 0x69, 0x5b, 0x40 | HII\_CONFIG\_ROUTING\_PROTOCOL.BlockToConfig - BlockToConfig() returns EFI\_INVALID\_PARAMETER with <ConfigElement> in ConfigRequest being a <NvConfig>. | Call BlockToConfig() with valid parameters except <ConfigElement> in ConfigRequest being a <NvConfig>, The return status should be EFI\_INVALID\_PARAMETER and Progresspoints to ‘&’ of the first non-<BlockName>. |
| 5.18.5.4.4 | 0xd38890ec, 0xd43e, 0x4e28, 0xab, 0x47, 0xef, 0x67, 0xeb, 0x2d, 0x3d, 0x92 | HII\_CONFIG\_ROUTING\_PROTOCOL.BlockToConfig - BlockToConfig() returns EFI\_DEVICE\_ERROR if Block is not large enough. | Call BlockToConfig() with with valid parameters except Block is not large enough. The return status should be EFI\_DEVICE\_ERROR. |
| 5.18.5.4.5 | 0x8b1b960c, 0xda67, 0x423c, 0x85, 0x31, 0x76, 0x28, 0x0d, 0xb8, 0x2a, 0xc1 | HII\_CONFIG\_ROUTING\_PROTOCOL.BlockToConfig - BlockToConfig() returns EFI\_SUCCESS with valid parameter and Progresspoints to theConfigRequest’s NULL terminator**.** | Call BlockToConfig() with valid parameters. The return status should be EFI\_SUCCESS and Progresspoints to theConfigRequest’s NULL terminator. |

### ConfigToBlock ()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.5.5.1 | 0x76ab8420, 0x7c61, 0x4ebc, 0x8b, 0x5b, 0x62, 0xa3, 0x35, 0x64, 0x6f, 0x8f | HII\_CONFIG\_ROUTING\_PROTOCOL. ConfigToBlock - ConfigToBlock() returns EFI\_INVALID\_PARAMETER with ConfigResp been NULL. | Call ConfigToBlock() with valid parameters except ConfigResp being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.5.2 | 0xcc284047, 0x45d6, 0x4fec, 0x88, 0x50, 0x70, 0x3f, 0x45, 0x22, 0x01, 0xdc | HII\_CONFIG\_ROUTING\_PROTOCOL. ConfigToBlock - ConfigToBlock() returns EFI\_INVALID\_PARAMETER with Block been NULL. Progressshouldpoint to the first character of ConfigResp. | Call ConfigToBlock() with valid parameters except Block being NULL, The return status should be EFI\_INVALID\_PARAMETER and Progressshouldpoint to the first character of ConfigResp. |
| 5.18.5.5.3 | 0x2d30da76, 0x9ec7, 0x480e, 0xb9, 0xe9, 0x6d, 0x50, 0x0d, 0x89, 0x21, 0xad | HII\_CONFIG\_ROUTING\_PROTOCOL. ConfigToBlock - ConfigToBlock() returns EFI\_INVALID\_PARAMETER with <RequestElement> in ConfigResp being a <Lable>. | Call ConfigToBlock() with valid parameters except < RequestElement > in ConfigResp being a <Lable>. The return status should be EFI\_INVALID\_PARAMETER and Progresspoints to ‘&’ of the first non-<BlockName>. |
| 5.18.5.5.4 | 0xa5b33ea4, 0x767b, 0x489a, 0xb3, 0x7b, 0xf9, 0xef, 0xfd, 0x62, 0xbc, 0x7b | HII\_CONFIG\_ROUTING\_PROTOCOL. ConfigToBlock - ConfigToBlock() returns EFI\_DEVICE\_ERROR if Block is not large enough. | Call ConfigToBlock() with valid parameters except Block is not large enough. The return status should be EFI\_DEVICE\_ERROR. |
| 5.18.5.5.5 | 0x59b759ff, 0x6c84, 0x407a, 0x9e, 0x24, 0x71, 0xe0, 0x65, 0x2d, 0xe3, 0x30 | HII\_CONFIG\_ROUTING\_PROTOCOL. ConfigToBlock - ConfigToBlock() returns EFI\_SUCCESS with valid parameter and Progresspoints to theConfigResp’s NULL terminator**.** | Call ConfigToBlock() with valid parameters. The return status should be EFI\_SUCCESS and Progresspoints to theConfigResp’s NULL terminator. |

### GetAltCfg ()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.5.6.1 | 0x1ff2326a, 0x8e88, 0x45db, 0x94, 0x81, 0x02, 0x83, 0x80, 0x20, 0xad, 0x02 | HII\_CONFIG\_ROUTING\_PROTOCOL. GetAltCfg - GetAltCfg() returns EFI\_INVALID\_PARAMETER with ConfigResp been NULL. | Call GetAltCfg() with valid parameters except ConfigResp being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.6.2 | 0xb9b88d34, 0x7479, 0x4807, 0xa4, 0xbf, 0x90, 0x35, 0x87, 0x0a, 0x3c, 0x1a | HII\_CONFIG\_ROUTING\_PROTOCOL. GetAltCfg - GetAltCfg() returns EFI\_INVALID\_PARAMETER with AltCfgResp been NULL. | Call GetAltCfg() with valid parameters except AltCfgResp being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.5.6.3 | 0xfe1e4232, 0x8819,  0x4f52, 0xac, 0xaa, 0xb2, 0x02, 0x72, 0x86, 0xc8, 0xe4 | HII\_CONFIG\_ROUTING\_PROTOCOL. GetAltCfg - GetAltCfg() returns EFI\_SUCCESS with NULL Guid, Name, DevicePath, except a valid AltCfgId. | Call GetAltCfg() with NULL Guid, Name, DevicePath, except a valid AltCfgId. The return status should be EFI\_SUCCESS and AltCfgRespshouldpoints to retrieved data. |
| 5.18.5.6.4 | 0xdf88e78e, 0x8f4d, 0x4027, 0xbb, 0xcd, 0xae, 0x10, 0x68, 0x58, 0xb6, 0x03 | HII\_CONFIG\_ROUTING\_PROTOCOL. GetAltCfg - GetAltCfg()returns EFI\_SUCCESS with NULL Name, DevicePath, except a valid Guid, AltCfgId. | Call GetAltCfg()with NULL Name, DevicePath, except a valid Guid, AltCfgId. The return status should be EFI\_SUCCESS and AltCfgRespshouldpoints to retrieved data. |
| 5.18.5.6.5 | 0x2b56a57a, 0xd906, 0x416c, 0x89, 0x76, 0x43, 0x5f, 0xc7, 0x1c, 0xb7, 0x73 | HII\_CONFIG\_ROUTING\_PROTOCOL. GetAltCfg - GetAltCfg() returns EFI\_SUCCESS with NULL Guid, DevicePath, except a valid Name, AltCfgId. | Call GetAltCfg() with NULL Guid, DevicePath, except a valid Name, AltCfgId. The return status should be EFI\_SUCCESS and AltCfgRespshouldpoints to retrieved data. |
| 5.18.5.6.6 | 0x17c575b3, 0x051f, 0x41eb, 0x89, 0xd1, 0x79, 0xb5, 0x8b, 0x0c, 0x92, 0x3c | HII\_CONFIG\_ROUTING\_PROTOCOL. GetAltCfg - GetAltCfg() returns EFI\_SUCCESS with NULL DevicePath, except a valid Guid, Name, AltCfgId. | Call GetAltCfg() with NULL DevicePath, except a valid Guid,Name, AltCfgId. The return status should be EFI\_SUCCESS and AltCfgRespshouldpoints to retrieved data. |
| 5.18.5.6.7 | 0xb948d2f8, 0x5c45, 0x4b10, 0x97, 0xb4, 0x95, 0x96, 0x97, 0x98, 0xe5, 0x8b | HII\_CONFIG\_ROUTING\_PROTOCOL. GetAltCfg - GetAltCfg() returns EFI\_SUCCESS returns EFI\_SUCCESS with NULL DevicePath, AltCfgId, except a valid Guid, Name. | Call GetAltCfg()with NULL DevicePath, AltCfgId, except a valid Guid, Name. The return status should be EFI\_SUCCESS and AltCfgRespshouldpoints to retrieved data. |
| 5.18.5.6.8 | 0xf732d246, 0x9fa5, 0x4ed3, 0x88, 0x95, 0x28, 0x63, 0xba, 0xf4, 0x68, 0x5d | HII\_CONFIG\_ROUTING\_PROT OCOL.GetAltCfg - GetAltCfg() returns EFI\_SUCCESS with valid Name | 1.Call GetAltCfg() with NULL GUID DevicePath, AltCfgId, except a valid Name.  2. The return status should be EFI\_SUCCESS and AltCfgResp should points to right data. |

## �EFI\_HII\_CONFIG\_ACCESS\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_HII\_CONFIG\_ACCESS\_PROTOCOL Section.

### ExtractConfig()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.6.1.1 | 0xa7173eb5, 0xf76a, 0x4ea1, 0x95, 0x0d, 0x14, 0x91, 0x1e, 0x49, 0x86, 0xc1 | HII\_CONFIG\_ACCESS\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_INVALID\_PARAMETER with Request been <MultiConfigRequest> format. | Call ExtractConfig() with valid parameters except with Request being <MultiConfigRequest> format., The return status should be EFI\_INVALID\_PARAMETER. And Progress should point to the most recent ‘&’ before the error or beginning of the string. |
| 5.18.6.1.2 | 0xfa5973e2, 0x0d05, 0x44c2, 0xaf, 0x2d, 0x1b, 0x68, 0x33, 0x42, 0x6d, 0x76 | HII\_CONFIG\_ACCESS\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_INVALID\_PARAMETER with Progress been NULL. | Call ExtractConfig() with valid parameters except Progress being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.6.1.3 | 0x6f6d1dd, 0x49b8, 0x488a, 0xa7, 0x75, 0xde, 0xbc, 0xc7, 0x60, 0xfd, 0x28 | HII\_CONFIG\_ACCESS\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_INVALID\_PARAMETER with Results been NULL. | Call ExtractConfig() with valid parameters except Results being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.6.1.4 | 0x28652613, 0x6bf4, 0x4f42, 0xab, 0xe2, 0x84, 0x4f, 0x2f, 0x77, 0xec, 0x2f | HII\_CONFIG\_ACCESS\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_NOT\_FOUND if Routing data doesn’t match any known driver or EFI\_INVALID\_PARAMETER if there is an unknown name in Request. | Call ExtractConfig() with an invalid Request. The ConfigHdr of Request can’t be found in current system. The return status should be EFI\_NOT\_FOUND. Progress should point to the error reason. If an unknown name in the Request, the return status should be EFI\_INVALID\_PARAMETER and Progress should point to the ‘&’ before the name in question. |
| 5.18.6.1.5 | 0x24dcf8bf, 0xbfbf, 0x4588, 0xba, 0x0f, 0x77, 0x1e, 0x24, 0x4e, 0x3e, 0x08 | HII\_CONFIG\_ACCESS\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_SUCCESS with valid parameters and and Progresspoints to theRequest’s NULL terminator. | Call ExtractConfig() with valid parameters. The return status should be EFI\_SUCCESS and the Progress should point to Request’s NULL terminator. |
| 5.18.6.1.6 | 0x961a5268, 0x1998, 0x4a7e, 0x9d, 0x9d, 0xce, 0xdc, 0x67, 0xfb, 0xcc, 0x77 | HII\_CONFIG\_ACCESS\_PROTOCOL.ExtractConfig - ExtractConfig() returns EFI\_SUCCESS with valid parameter exceptRequest been NULL**.** | Call ExtractConfig() with valid parameters exceptRequest been NULL. The return status should be EFI\_SUCCESS. |
| 5.18.6.1.7 | 0xab163674, 0x6c27, 0x4169, 0xa6, 0xa9, 0xe1, 0x9c, 0x88, 0x14, 0x94, 0x96 | HII\_CONFIG\_ACCESS\_PROT OCOL.ExtractConfig - ExtractConfig() returns EFI\_SUCCESS. Check if Results is in **<**MultiConfigAltResp> format | Call ExtractConfig() with valid parameters. The return status should be EFI\_SUCCESS and Check if Results is in <MultiConfigAltResp> format. |

### RouteConfig()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.6.2.1 | 0xebba197a, 0x467f, 0x4736, 0x92, 0xf2, 0x11, 0xb1, 0x91, 0x2e, 0xe9, 0x90 | HII\_CONFIG\_ACCESS\_PROTOCOL.RouteConfig - RouteConfig() returns EFI\_INVALID\_PARAMETER with Configuration been NULL. | Call RouteConfig() with valid parameters except with Configuration being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.6.2.2 | 0x341fe3e0, 0xf688, 0x45f2, 0x91, 0x56, 0xc7, 0xae, 0x9f, 0x2c, 0xcb, 0xb0 | HII\_CONFIG\_ACCESS\_PROTOCOL. RouteConfig - RouteConfig() returns EFI\_INVALID\_PARAMETER with Progress been NULL. | Call RouteConfig() with valid parameters except Progress being NULL, The return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.6.2.3 | 0x1f99ebc8, 0x0253, 0x455f, 0x88, 0xac, 0x9e, 0x2b, 0xa6, 0xdc, 0xd7, 0x29 | HII\_CONFIG\_ACCESS\_PROTOCOL. RouteConfig - RouteConfig() returns EFI\_NOT\_FOUND if no target was found with the routing data. | Call RouteConfig() with no found target for the routing data. The return status should be EFI\_NOT\_FOUND. |
| 5.18.6.2.4 | 0x603e52f0, 0x2ce3, 0x4e7a, 0xa7, 0x2e, 0xdf, 0x8c, 0xa3, 0xfd, 0xb2, 0x0d | HII\_CONFIG\_ACCESS\_PROTOCOL. RouteConfig - RouteConfig()returns EFI\_SUCCESS with valid parameters and and Progresspoints to theConfiguration’s NULL terminator. | Call RouteConfig()with valid parameters. The return status should be EFI\_SUCCESS and the Progress should point to Configuration’s NULL terminator. |

## EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL Section.

### SetData()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.7.1.1 | 0xf046a19c, 0xffc1, 0x4fd9, 0x9d, 0x73, 0x92, 0x4f, 0x8c, 0x43, 0xcf, 0xfb | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. SetData() - SetData()  returns EFI\_NOT\_FOUNDwhen an element of the KeywordString was not found. Progress points to the most recent '&' before the first failing string element and ProgressErr should be KEYWORD\_HANDLER\_KEYWORD\_NOT\_FOUND. | 1. Call SetData() when an element of the KeywordString was not found, the return status should be EFI\_NOT\_FOUND. Progress points to the most recent '&' before the first failing string element and  ProgressErr should be KEYWORD\_HANDLER\_KEYWORD\_NOT\_FOUND. |
| 5.18.7.1.2 | 0x553c956c, 0x78c1, 0x44d4, 0x81, 0x8e, 0x98, 0xdf, 0xd2, 0x25, 0x8, 0xe5 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. SetData() - GetData()returns EFI\_SUCCESS, ProgressErr should be KEYWORD\_HANDLER\_ NO\_ERROR. Progress points to the string’s NULL terminator. | 2.  Check the system with GetData(), the storage associated with the earlier keywords is not modified when an EFI\_NOT\_FOUND error is generated during processing the second or later keyword element. |
| 5.18.7.1.3 | 0xe334ff21, 0x4005, 0x449a, 0x83, 0x1, 0x97, 0x44, 0xc1, 0xb0, 0xaf, 0xd5 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. SetData() -  SetData()returns EFI\_SUCCESSwhen an element of the KeywordString was found. Progress points to the string’s NULL terminator and ProgressErr should be KEYWORD\_HANDLER\_ NO\_ERROR. | 1. Call SetData() when an element of the KeywordString was found, the return status should be EFI\_SUCCESS. Progress points to the string’s NULL terminator and ProgressErr should be KEYWORD\_HANDLER\_ NO\_ERROR. |
| 5.18.7.1.4 | 0x8a4618b3, 0xa012, 0x40c4, 0xba, 0x6, 0xa, 0x93, 0x79, 0xb4, 0x64, 0x58 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. SetData() - GetData()returns EFI\_SUCCESS, ProgressErr should be KEYWORD\_HANDLER\_ NO\_ERROR. Progress points to the string’s NULL terminator. | 2.  Check the system with GetData(), the storage associated with the earlier keywords should be saved correctly. |
| 5.18.7.1.5 | 0xfe4f680c, 0xcbe, 0x4f85, 0xb3, 0x20, 0x5e, 0xcc, 0x9d, 0xce, 0xc5, 0x88 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. SetData() -  SetData()returns EFI\_INVALID\_PARAMETERwhen KeywordString was NULL. | 1. Call SetData() when KeywordString was found, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.7.1.6 | 0xe7966ef2, 0x941e, 0x4a59, 0x8e, 0x15, 0x2f, 0xde, 0x41, 0x9d, 0xfc, 0x91 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. SetData() -  SetData()returns EFI\_INVALID\_PARAMETERwhen parsing of the KeywordString resulted in an error and Progress points to the most recent '&' before the first failing string element. | 1. Call SetData() when parsing of the KeywordString resulted in an error, the return status should be EFI\_INVALID\_PARAMETER. Progress should point to the most recent '&' before the first failing string element. |
| 5.18.7.1.7 | 0x1eff122d, 0xa263, 0x43bd, 0x94, 0xfc, 0x82, 0xb, 0x8b, 0xc9, 0xfa, 0x7c | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. SetData() -  SetData()returns EFI\_NOT\_FOUNDwhen an element of the KeywordString was not found and Progress points to the most recent '&' before the first failing string element. | 1. Call SetData() when an element of the KeywordString was not found, the return status should be EFI\_NOT\_FOUND. Progress should point to the most recent '&' before the first failing string element. |
| 5.18.7.1.8 | 0x4bd58084, 0xb158, 0x43fe, 0xbb, 0x87, 0x31, 0x8f, 0xb2, 0x3f, 0x7a, 0xe9 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. SetData() -  SetData()returns EFI\_ACCESS\_DENIEDwhen the ReadOnly element is written and Progress points to the most recent '&' before the first failing string element. | 1. Call SetData() when the ReadOnly element is written, the return status should be EFI\_ACCESS\_DENIED. Progress should point to the most recent '&' before the first failing string element. |

### GetData()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.7.2.1 | 0x852b267e, 0xcbe, 0x4bd6, 0x85, 0x4d, 0x3b, 0xbd, 0xf0, 0xa0, 0xc, 0x49 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()returns EFI\_SUCCESS when KeywordString and NameSpaceId are NULL. | 1. Call GetData()when KeywordString and NameSpaceId are NULL, the return status should be EFI\_SUCCESS. ProgressErr should be KEYWORD\_HANDLER\_NO\_ERROR. |
| 5.18.7.2.2 | 0x247b91db, 0xf60b, 0x457f, 0xb9, 0x10, 0xb3, 0xc3, 0x30, 0xa8, 0xaf, 0x88 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()should output the correct result as expected format. | 2. The preinstalled Str should be included in the Results outputted from the GetData(). |
| 5.18.7.2.3 | 0xf57e9ce0, 0x827a, 0x4d35, 0x89, 0xb8, 0xde, 0x24, 0x57, 0xe7, 0x94, 0xfb | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()returns EFI\_SUCCESS when KeywordString is NULL and NameSpaceId is one valid expression. | 1. Call GetData()when KeywordString is NULL and NameSpaceId is one valid expression, the return status should be EFI\_SUCCESS. ProgressErr should be KEYWORD\_HANDLER\_NO\_ERROR. |
| 5.18.7.2.4 | 0x170ab626, 0x648c, 0x4088, 0x8b, 0x5d, 0xf8, 0xf2, 0x9d, 0x65, 0xaf, 0xba | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()should output the correct result as expected format. | 2. The preinstalled Str should be included in the Results outputted from the GetData(). |
| 5.18.7.2.5 | 0x60bcfe65, 0xe73a, 0x46dd, 0xa9, 0x42, 0x22, 0xb4, 0xeb, 0x30, 0xb8, 0x7c | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()returns EFI\_SUCCESS when KeywordString is the valid expression (with PathHdr) and NameSpaceId is one valid expression. | 1. Call GetData() when KeywordString is the valid expression (with PathHdr) and NameSpaceId is one valid expression, the return status should be EFI\_SUCCESS. ProgressErr should be KEYWORD\_HANDLER\_NO\_ERROR and Progress points to the string’s NULL terminator. |
| 5.18.7.2.6 | 0x7cc0b84, 0x4128, 0x4c66, 0x91, 0x90, 0x76, 0x15, 0x81, 0xb, 0x95, 0x9d | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()returns EFI\_SUCCESS when KeywordString is the valid expression (without PathHdr) and NameSpaceId is one valid expression. | 2. Call GetData() when KeywordString is the valid expression (without PathHdr) and NameSpaceId is one valid expression, the return status should be EFI\_SUCCESS. ProgressErr should be KEYWORD\_HANDLER\_NO\_ERROR and Progress points to the string’s NULL terminator. |
| 5.18.7.2.7 | 0x6114b15, 0xab62, 0x40f5, 0x86, 0xf6, 0x21, 0xd1, 0x81, 0x2b, 0x7f, 0x6c | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()should output the correct result as expected format. | 3. The Results outputted with PathHdr should be included in the Results outputted without PathHdr |
| 5.18.7.2.8 | 0x378ef819, 0x29ee, 0x4875, 0x8c, 0xb2, 0x94, 0x6a, 0x77, 0xb1, 0x48, 0x73 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()returns EFI\_INVALID\_PARAMETER when Progress, ProgressErr, or Resuts is NULL. | 1. Call GetData() when Progress, ProgressErr, or Resuts is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.7.2.9 | 0xb90fe257, 0xf693, 0x4c3e, 0x89, 0x59, 0x14, 0xb, 0xcf, 0x44, 0x7b, 0x5d | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() - GetData()returns EFI\_INVALID\_PARAMETER when Parsing of the KeywordString resulted in an error. | 1. Call GetData() when Parsing of the KeywordString resulted in an error, the return status should be EFI\_INVALID\_PARAMETER. Progress should point to the most recent '&' before the first failing string element and ProgressErr should be KEYWORD\_HANDLER\_MALFORMED\_STRING. |
| 5.18.7.2.10 | 0x138298f2, 0x7b86, 0x49b7, 0x9c, 0xa7, 0x6d, 0x69, 0xbe, 0x8b, 0x52, 0xfd | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()returns EFI\_NOT\_FOUND when an element of the KeywordString was not found. | 1. Call GetData() when an element of the KeywordString was not found, the return status should be EFI\_NOT\_FOUND. Progress should point to the most recent '&' before the first failing string element and ProgressErr should be KEYWORD\_HANDLER\_KEYWORD\_NOT\_FOUND. |
| 5.18.7.2.11 | 0x48dab3bf, 0xb3dc, 0x4960, 0xa6, 0xf8, 0xb5, 0x1c, 0xd3, 0xfa, 0xfa, 0xe0 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()returns EFI\_NOT\_FOUND when the NamespaceId specified was not found. | 1. Call GetData() when the NamespaceId specified was not found, the return status should be EFI\_NOT\_FOUND. ProgressErr should be KEYWORD\_HANDLER\_KEYWORD\_NOT\_FOUND. |
| 5.18.7.2.12 | 0xab69961e, 0xd77d, 0x4781, 0x8e, 0xe5, 0xf9, 0x13, 0x55, 0xc7, 0xce, 0x91 | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()returns EFI\_NOT\_FOUND when an element of the KeywordString was not found. | 1. Call GetData() when an element of the KeywordString was not found, the return status should be EFI\_NOT\_FOUND. Progress should point to the most recent '&' before the first failing string element and ProgressErr should be KEYWORD\_HANDLER\_KEYWORD\_NOT\_FOUND. |
| 5.18.7.2.13 | 0xc6b310c5, 0xdddf, 0x4e1d, 0x9d, 0x8c, 0x20, 0x16, 0xe7, 0x66, 0xa6, 0xae | EFI\_CONFIG\_KEYWORD\_HANDLER\_PROTOCOL. GetData() -  GetData()should output Results string contains values returned for all keywords processed prior to the keyword generating the error. | 2. The returned Results string should contain values for all keywords processed prior to the keyword generating the error. |

## EFI\_HII\_FONT\_EX\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_HII\_FONT\_EX\_PROTOCOL Section.

### StringToImageEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.8.1.1 | 0x81b18c28, 0x7d09, 0x4794, 0xab, 0x4e, 0x92, 0x9b, 0xb7, 0x2f, 0x19, 0x67 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_INVALID\_PARAMETER when String is NULL. | 1. Call StringToImageEx() when String is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.1.2 | 0xeba34749, 0x9763, 0x4203, 0x9f, 0xd, 0x26, 0x3a, 0xa4, 0xe9, 0xd6, 0x9a | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_INVALID\_PARAMETER when Blt is NULL. | 1. Call StringToImageEx() when Blt is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.1.3 | 0xd6514302, 0x4b34, 0x4bae, 0xa0, 0xcd, 0x37, 0x77, 0xb8, 0x43, 0xc, 0x26 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_INVALID\_PARAMETER with invalid Flags combination. | 1. Call StringToImageEx() when Flags is the combination of EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X and EFI\_HII\_OUT\_FLAG\_WRAP, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.1.4 | 0xf711f218, 0x8987, 0x4fa9, 0xb4, 0xb6, 0x64, 0x1e, 0xc1, 0x76, 0xe1, 0xc8 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_INVALID\_PARAMETER with invalid Flags combination. | 1. Call StringToImageEx() when Flags is EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X without EFI\_HII\_OUT\_FLAG\_CLIP, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.1.5 | 0x4dd0210d, 0x87b1, 0x4352, 0xa6, 0x16, 0x57, 0x91, 0x78, 0x73, 0xe0, 0xa0 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with valid Flags combination. | 1. Call StringToImageEx() with the valid Flags combination and use EFI\_GRAPUICS\_OUTPUT\_BLT\_PIXEL structure in EFI\_IMAGE\_OUTPUT structure, the return status should be EFI\_SUCCESS. |
| 5.18.8.1.6 | 0x2af74a94, 0xed7, 0x4b68, 0x9c, 0xdd, 0xfa, 0xdf, 0xfe, 0x6, 0x68, 0x1f | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with valid Flags combination. | 1. Call StringToImageEx() with the valid Flags combination and use EFI\_GRAPUICS\_OUTPUT\_PROTOCOL in EFI\_IMAGE\_OUTPUT structure, the return status should be EFI\_SUCCESS. |
| 5.18.8.1.7 | 0x7047fe55, 0x6c8c, 0x4062, 0x8a, 0x24, 0x26, 0xb5, 0x33, 0x88, 0x62, 0x81 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with valid parameters for all ASCII visible characters. Each image must equal to sys default glyph. | 1. Call StringToImageEx() with the valid parameters and StringInfo is NULL. Compare image output with system default font glyph image. |
| 5.18.8.1.8 | 0xf09da704, 0x352, 0x4afa, 0x90, 0x8f, 0x83, 0x73, 0xf2, 0xe9, 0xe6, 0x2c | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with valid parameters for all ASCII visible characters. Each image must equal to the specific font glyph. | 1. Register a specific font package. Call StringToImageEx() with the valid parameters and StringInfo is the specific font. Compare image output with specific font glyph image registered. |
| 5.18.8.1.9 | 0xbee39111, 0x1e5b, 0x4574, 0xae, 0xeb, 0x2, 0xdd, 0xaa, 0x17, 0x42, 0xbf | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Call StringToImageEx() with  EFI\_HII\_DIRECT\_TO\_SCREEN. For the final row, the RowInfoArray.LineHeight and RowInfoArray.BaseLine may describe pixels which are outside the limit specified by Blt.Height (unless EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y is specified) even though those pixels were not drawn.  2. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.10 | 0x2c36e6b5, 0x983f, 0x4e05, 0x90, 0xdd, 0xfa, 0x79, 0xfd, 0xdb, 0x15, 0xcd | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_DIRECT\_TO\_SCREEN. For the final row, the RowInfoArray.LineHeight and RowInfoArray.BaseLine may describe pixels which are outside the limit specified by Blt.Height (unless EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y is specified) even though those pixels were not drawn.  2. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.11 | 0x7dd51e66, 0xf38f, 0x4412, 0xa6, 0xd8, 0x32, 0x37, 0x85, 0xb9, 0x8, 0x31 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X | EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X | EFI\_HII\_DIRECT\_TO\_SCREEN. If a character’s right-most pixel can’t fit, then it will not be drawn at all.  2. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.12 | 0x76805500, 0x3e74, 0x44cb, 0x95, 0x9b, 0x63, 0xf7, 0xb7, 0x78, 0x92, 0x17 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y | EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y | EFI\_HII\_DIRECT\_TO\_SCREEN. If a row’s bottom-most pixel exceeds screen Height, then it will not be drawn at all.  2. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.13 | 0xe18566cf, 0x619d, 0x454c, 0x85, 0x6b, 0xe, 0x4e, 0xd3, 0x1c, 0x4a, 0xf1 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_IGNORE\_IF\_NO\_GLYPH | EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String with line break opportunity. | 1. Call StringToImageEx() with EFI\_HII\_IGNORE\_IF\_NO\_GLYPH | EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String with line break opportunity (SPACE is a line break opportunity). Check display with wrapper at right place.  2. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.14 | 0xacba2f9a, 0x1052, 0x478d, 0x96, 0x99, 0x78, 0xa1, 0x1e, 0x65, 0x5, 0x5d | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String without line break opportunity. | 1. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String without line break opportunity. String is designed to display as if EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X is set.  2. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.15 | 0x82482a71, 0x2a32, 0x4104, 0xb7, 0x32, 0x91, 0xa0, 0x95, 0x81, 0x50, 0x49 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_IGNORE\_LINE\_BREAK | EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Call StringToImageEx() with EFI\_HII\_IGNORE\_LINE\_BREAK | EFI\_HII\_DIRECT\_TO\_SCREEN. If a row’s bottom-most pixel can’t fit, then it will not be drawn at all. This flag requires that EFI\_HII\_OUT\_FLAG\_CLIP be set.  2. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.16 | 0xf1c89a03, 0x5b7a, 0x4d1d, 0xbe, 0x9, 0x5c, 0xf7, 0xe5, 0x67, 0xe, 0x77 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Register a specific font package.  2. Call StringToImageEx() with EFI\_HII\_DIRECT\_TO\_SCREEN.  3. Check EFI\_HII\_DIRECT\_TO\_SCREEN only case if Blt is not NULL, then EFI\_HII\_OUT\_FLAG\_CLIP is implied. String is designed to display with full line.  4. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.17 | 0x2154d7a2, 0x37e2, 0x43a3, 0xb4, 0xaf, 0xb3, 0x74, 0x8a, 0x6c, 0x54, 0xf0 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_CLIP. | 1. Register a specific font package.  2. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_CLIP.  3. For the final row, the RowInfoArray.LineHeight andRowInfoArray.BaseLine may describe pixels which are outside the limit specified by Blt.Height (unless EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y is specified) even though those pixels were not drawn.  4. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.18 | 0x6206dfcf, 0x6fb3, 0x4020, 0xba, 0xf3, 0x74, 0xe, 0xed, 0xac, 0x9c, 0xb2 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X | EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Register a specific font package.  2. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X | EFI\_HII\_DIRECT\_TO\_SCREEN.  3. If a character’s right-most pixel can’t fit, then it will not be drawn at all.  4. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.19 | 0x76bd46eb, 0x56a1, 0x4b66, 0xab, 0x63, 0x2e, 0xf1, 0x69, 0x1a, 0xfd, 0x80 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y | EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Register a specific font package.  2. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_CLIP | EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_Y | EFI\_HII\_DIRECT\_TO\_SCREEN.  3. If a row’s bottom-most pixel exceeds screen Height, then it will not be drawn at all.  4. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.20 | 0x9782016a, 0xcd4c, 0x4d39, 0x91, 0xc3, 0x7e, 0xe3, 0xce, 0xfd, 0xcc, 0x2d | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_IGNORE\_IF\_NO\_GLYPH | EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String with line break opportunity. | 1. Register a specific font package.  2. Call StringToImageEx() with EFI\_HII\_IGNORE\_IF\_NO\_GLYPH | EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String with line break opportunity (Space is a line-break).  3. Check if the display is right.  4. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.21 | 0x2833962d, 0x3800, 0x45b3, 0x90, 0xf8, 0xfb, 0xe2, 0xee, 0xc6, 0x6e, 0xd9 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String without line break opportunity. | 1. Register a specific font package.  2. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_WRAP | EFI\_HII\_DIRECT\_TO\_SCREEN and String without line break opportunity.  3. String is designed to display as if EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X is set.  4. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.22 | 0x12eb38a6, 0xfc, 0x4568, 0xa3, 0x44, 0x75, 0x40, 0xd3, 0x89, 0x88, 0xbe | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_IGNORE\_LINE\_BREAK | EFI\_HII\_DIRECT\_TO\_SCREEN. | 1. Register a specific font package.  2. Call StringToImageEx() with EFI\_HII\_IGNORE\_LINE\_BREAK | EFI\_HII\_DIRECT\_TO\_SCREEN.  3. If a row’s bottom-most pixel can’t fit, then it will not be drawn at all. This flag requires that EFI\_HII\_OUT\_FLAG\_CLIP be set.  4. The return status should be EFI\_SUCCESS. |
| 5.18.8.1.23 | 0x9c9802d4, 0x98e5, 0x46b9, 0xab, 0xc7, 0x66, 0x17, 0xb7, 0x80, 0x40, 0x29 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringToImageEx() - StringToImageEx() returns EFI\_SUCCESS with parameters EFI\_HII\_OUT\_FLAG\_TRANSPARENT. | 1. Register a specific font package.  2. Call StringToImageEx() with EFI\_HII\_OUT\_FLAG\_TRANSPARENT.  3. Check the output buffer StringInfo background should be ignored according to UEFI Spec.  4. The return status should be EFI\_SUCCESS. |

### StringIdToImageEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.8.2.1 | 0x7baa464a, 0x572c, 0x4fa9, 0x80, 0xa3, 0x99, 0xa0, 0x61, 0xc0, 0x46, 0x4f | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_INVALID\_PARAMETER when Blt is NULL. | 1. Call StringIdToImageEx() when Blt is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.2.2 | 0xa086a16b, 0x6e61, 0x4f06, 0xb5, 0xd, 0xac, 0x6e, 0x80, 0x71, 0x11, 0xe4 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_INVALID\_PARAMETER when PackageList is NULL. | 1. Call StringIdToImageEx() when PackageList is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.2.3 | 0x10931cc4, 0xfa08, 0x4df8, 0xab, 0x6a, 0xb3, 0x8f, 0xa5, 0xc6, 0x84, 0x24 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_NOT\_FOUND when PackageList is not in Database. | 1. Call StringIdToImageEx() when PackageList is not in Database, the return status should be EFI\_NOT\_FOUND. |
| 5.18.8.2.4 | 0x7623d5de, 0x71e9, 0x49f6, 0xb7, 0x9f, 0xd2, 0x6f, 0x38, 0x69, 0xae, 0xe9 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_NOT\_FOUND when StringId is not in PackageList. | 1. Call StringIdToImageEx() when StringId is not in PackageList, the return status should be EFI\_NOT\_FOUND. |
| 5.18.8.2.5 | 0x36cd9086, 0x8e5e, 0x4a95, 0xb4, 0xdd, 0x56, 0x94, 0x74, 0x5c, 0x21, 0x37 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_INVALID\_PARAMETER when Flags is the invalid combination. | 1. Call StringIdToImageEx() when Flags are EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X with EFI\_HII\_OUT\_FLAG\_WRAP, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.2.6 | 0x16b7317e, 0x1196, 0x4323, 0x9d, 0xeb, 0xe8, 0xc7, 0x44, 0x32, 0x7e, 0x20 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_INVALID\_PARAMETER when Flags is the invalid combination. | 1. Call StringIdToImageEx() when Flags is EFI\_HII\_OUT\_FLAG\_CLIP\_CLEAN\_X without EFI\_HII\_OUT\_FLAG\_CLIP, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.2.7 | 0xc3a512bc, 0x6464, 0x4e74, 0xab, 0x8d, 0x41, 0xd5, 0x42, 0xd6, 0xad, 0x66 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_SUCCESS with valid parameters. | 1. Call StringIdToImageEx() with valid parameters and use EFI\_GRAPUICS\_OUTPUT\_BLT\_PIXEL structure in EFI\_IMAGE\_OUTPUT structure, the return status should be EFI\_SUCCESS. |
| 5.18.8.2.8 | 0x9c84a237, 0x9ba5, 0x417a, 0x94, 0xcd, 0xf5, 0xed, 0x37, 0xf7, 0xbb, 0x9e | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_SUCCESS with valid parameters. | 1. Call StringIdToImageEx() with valid parameters and use EFI\_GRAPUICS\_OUTPUT\_PROTOCOL structure in EFI\_IMAGE\_OUTPUT structure, the return status should be EFI\_SUCCESS. |

### GetGlyphEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.8.3.1 | 0x4e1b65f1, 0xa0c1, 0x4f13, 0xb6, 0xfb, 0x2a, 0xdc, 0xaa, 0x21, 0x8d, 0x89 | EFI\_HII\_FONT\_EX\_PROTOCOL. GetGlyphEx() - GetGlyphEx ()returns EFI\_INVALID\_PARAMETER when Blt is NULL. | 1. Call GetGlyphEx() when Blt is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.3.2 | 0x139af9e5, 0x5d3e, 0x46b2, 0x83, 0x9c, 0x52, 0x54, 0x66, 0xf1, 0xe0, 0xe | EFI\_HII\_FONT\_EX\_PROTOCOL. GetGlyphEx() - GetGlyphEx ()returns EFI\_INVALID\_PARAMETER when \*Blt is not NULL. | 1. Call GetGlyphEx() when \*Blt is not NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.3.3 | 0xf3fc9dce, 0x7f2c, 0x45d7, 0x87, 0xcf, 0x55, 0x17, 0xea, 0xcf, 0x9d, 0x4d | EFI\_HII\_FONT\_EX\_PROTOCOL. GetGlyphEx() - GetGlyphEx ()returns EFI\_SUCCESS with valid parameters. | 1. Call GetGlyphEx() with valid parameters, the return status should be EFI\_SUCCESS. |

### GetFontInfoEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.8.4.1 | 0x9511abcb, 0x462e, 0x4b96, 0xb3, 0xf, 0xbf, 0x9b, 0xf5, 0x68, 0x73, 0xeb | EFI\_HII\_FONT\_EX\_PROTOCOL.  GetFontInfoEx() - GetFontInfoEx()returns EFI\_INVALID\_PARAMETER with invalid EFI\_FONT\_INFO\_MASK combination. | 1. Call GetFontInfoEx() when StringInfoIn->FontInfoMask is the invalid combination, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.4.2 | 0x167059e1, 0x4bf6, 0x4d8c, 0xb0, 0x96, 0x7b, 0xf4, 0x61, 0x7b, 0x75, 0x4b | EFI\_HII\_FONT\_EX\_PROTOCOL. GetFontInfoEx() - GetFontInfoEx()returns EFI\_SUCCESS with valid parameters. | 1. Call GetFontInfoEx() with valid parameters, the return status should be EFI\_SUCCESS. |
| 5.18.8.4.3 | 0x29a5204a, 0x507e, 0x4dc0, 0xa1, 0xb1, 0x90, 0x53, 0xf7, 0x2e, 0xd7, 0x77 | EFI\_HII\_FONT\_EX\_PROTOCOL. GetFontInfoEx() - GetFontInfoEx()returns EFI\_SUCCESS with valid parameters(StringInfoIn is NULL). | 1. Call GetFontInfoEx() with valid parameters(StringInfoIn is NULL), the return status should be EFI\_SUCCESS. |

### GetGlyphInfo()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.8.5.1 | 0x298cb0c7, 0x7e78, 0x4e3e, 0x8d, 0x42, 0xc2, 0x2c, 0x16, 0xa0, 0x83, 0x31 | EFI\_HII\_FONT\_EX\_PROTOCOL. GetGlyphInfo() - GetGlyphInfo()returns EFI\_INVALID\_PARAMETER when GlyphInfo is NULL. | 1. Call GetGlyphInfo() when GlyphInfo is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.5.2 | 0xb20f87ce, 0xbc6b, 0x4e27, 0xb8, 0x2a, 0x61, 0x53, 0x59, 0xab, 0x92, 0xa7 | EFI\_HII\_FONT\_EX\_PROTOCOL. GetFontInfoEx() - GetFontInfoEx()returns EFI\_INVALID\_PARAMETER when FontDisplayInfo is NULL. | 1. Call GetGlyphInfo() when FontDisplayInfo is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.5.3 | 0x347f2e9e, 0x70c4, 0x4e89, 0xb9, 0x4, 0x7e, 0x5f, 0xbd, 0x78, 0x4d, 0xb3 | EFI\_HII\_FONT\_EX\_PROTOCOL. GetFontInfoEx() - GetFontInfoEx()returns EFI\_SUCCESS with valid parameters. | 1. Call GetGlyphInfo() with valid parameters, the return status should be EFI\_SUCCESS. |

## EFI\_HII\_IMAGE\_EX\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_HII\_IMAGE\_EX\_PROTOCOL Section.

### NewImageEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.9.3.1 | 0xe88ca946, 0xed6d, 0x415d, 0x85, 0x55, 0x0, 0x27, 0x9f, 0x14, 0xc3, 0xf9 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_NOT\_FOUND when ImageId is invalid. | 0x7baa464a, 0x572c, 0x4fa9, 0x80, 0xa3, 0x99, 0xa0, 0x61, 0xc0, 0x46, 0x4f |
| 5.18.9.3.2 | 5.18.9.6.2 | 0xf61dfb48, 0x1c77, 0x4907, 0x9f, 0xab, 0x43, 0x93, 0x17, 0x8c, 0x99, 0xee | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_INVALID\_PARAMETER when Image is NULL. |
| 5.18.8.2.2 | 0xa086a16b, 0x6e61, 0x4f06, 0xb5, 0xd, 0xac, 0x6e, 0x80, 0x71, 0x11, 0xe4 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_INVALID\_PARAMETER when PackageList is NULL. | 1. Call StringIdToImageEx() when PackageList is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.2.3 | 0x10931cc4, 0xfa08, 0x4df8, 0xab, 0x6a, 0xb3, 0x8f, 0xa5, 0xc6, 0x84, 0x24 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_NOT\_FOUND when PackageList is not in Database. | 1. Call StringIdToImageEx() when PackageList is not in Database, the return status should be EFI\_NOT\_FOUND. |

### GetImageEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.9.3.1 | 0xe88ca946, 0xed6d, 0x415d, 0x85, 0x55, 0x0, 0x27, 0x9f, 0x14, 0xc3, 0xf9 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_NOT\_FOUND when ImageId is invalid. | 0x7baa464a, 0x572c, 0x4fa9, 0x80, 0xa3, 0x99, 0xa0, 0x61, 0xc0, 0x46, 0x4f |
| 5.18.9.3.2 | 5.18.9.6.2 | 0xf61dfb48, 0x1c77, 0x4907, 0x9f, 0xab, 0x43, 0x93, 0x17, 0x8c, 0x99, 0xee | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_INVALID\_PARAMETER when Image is NULL. |
| 5.18.8.2.2 | 0xa086a16b, 0x6e61, 0x4f06, 0xb5, 0xd, 0xac, 0x6e, 0x80, 0x71, 0x11, 0xe4 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_INVALID\_PARAMETER when PackageList is NULL. | 1. Call StringIdToImageEx() when PackageList is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.2.3 | 0x10931cc4, 0xfa08, 0x4df8, 0xab, 0x6a, 0xb3, 0x8f, 0xa5, 0xc6, 0x84, 0x24 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_NOT\_FOUND when PackageList is not in Database. | 1. Call StringIdToImageEx() when PackageList is not in Database, the return status should be EFI\_NOT\_FOUND. |
| 5.18.8.2.4 | 0x7623d5de, 0x71e9, 0x49f6, 0xb7, 0x9f, 0xd2, 0x6f, 0x38, 0x69, 0xae, 0xe9 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_NOT\_FOUND when StringId is not in PackageList. | 1. Call StringIdToImageEx() when StringId is not in PackageList, the return status should be EFI\_NOT\_FOUND. |

### SetImageEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.9.3.1 | 0xe88ca946, 0xed6d, 0x415d, 0x85, 0x55, 0x0, 0x27, 0x9f, 0x14, 0xc3, 0xf9 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_NOT\_FOUND when ImageId is invalid. | 0x7baa464a, 0x572c, 0x4fa9, 0x80, 0xa3, 0x99, 0xa0, 0x61, 0xc0, 0x46, 0x4f |
| 5.18.9.3.2 | 5.18.9.6.2 | 0xf61dfb48, 0x1c77, 0x4907, 0x9f, 0xab, 0x43, 0x93, 0x17, 0x8c, 0x99, 0xee | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_INVALID\_PARAMETER when Image is NULL. |
| 5.18.8.2.2 | 0xa086a16b, 0x6e61, 0x4f06, 0xb5, 0xd, 0xac, 0x6e, 0x80, 0x71, 0x11, 0xe4 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_INVALID\_PARAMETER when PackageList is NULL. | 1. Call StringIdToImageEx() when PackageList is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.8.2.3 | 0x10931cc4, 0xfa08, 0x4df8, 0xab, 0x6a, 0xb3, 0x8f, 0xa5, 0xc6, 0x84, 0x24 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_NOT\_FOUND when PackageList is not in Database. | 1. Call StringIdToImageEx() when PackageList is not in Database, the return status should be EFI\_NOT\_FOUND. |
| 5.18.8.2.4 | 0x7623d5de, 0x71e9, 0x49f6, 0xb7, 0x9f, 0xd2, 0x6f, 0x38, 0x69, 0xae, 0xe9 | EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_NOT\_FOUND when StringId is not in PackageList. | 1. Call StringIdToImageEx() when StringId is not in PackageList, the return status should be EFI\_NOT\_FOUND. |

### DrawImageEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.9.4.1 | 0x42dd08a5, 0xbd85, 0x4eab, 0xb4, 0x74, 0x9f, 0xe2, 0x55, 0x71, 0x56, 0x8f | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageEx() - DrawImageEx() returns EFI\_INVALID\_PARAMETER when Image is NULL. | 1. Call DrawImageEx() when Image is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.9.4.2 | 0xcf06b84d, 0x8d1f, 0x43c1, 0xb5, 0xb2, 0xa3, 0x3a, 0x2, 0xc2, 0xd, 0x50 | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageEx() - DrawImageEx() returns EFI\_INVALID\_PARAMETER when Flag is EFI\_HII\_DRAW\_FLAG\_TRANSPARENT and Blt is NULL. | 1. Call DrawImageEx() when Flag is EFI\_HII\_DRAW\_FLAG\_TRANSPARENT and Blt is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.9.4.3 | 0xa20a8ee4, 0x9bed, 0x4538, 0x94, 0x7a, 0xbf, 0xb7, 0x42, 0xa6, 0xaf, 0xd9 | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageEx() - DrawImageEx() returns EFI\_INVALID\_PARAMETER when Flag is EFI\_HII\_DIRECT\_TO\_SCREEN and no screen. | 1. Call DrawImageEx() when Flag is EFI\_HII\_DIRECT\_TO\_SCREEN and no screen, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.9.4.4 | 0x8a4f106c, 0xdb5d, 0x4491, 0x96, 0xbd, 0x62, 0x9a, 0xa8, 0xa2, 0xc4, 0x25 | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageEx() - DrawImageEx() returns EFI\_INVALID\_PARAMETER when Flag is EFI\_HII\_DRAW\_FLAG\_CLIP and Blt points to NULL. | 1. Call DrawImageEx() when Flag is EFI\_HII\_DRAW\_FLAG\_CLIP and Blt points to NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.9.4.5 | 0x4ed61351, 0xc6de, 0x4910, 0x97, 0x15, 0xcf, 0xc5, 0x5e, 0xe, 0x75, 0x9b | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageEx() - DrawImageEx() returns EFI\_INVALID\_PARAMETER when Flag is EFI\_HII\_DRAW\_FLAG\_DEFAULT and Blt points to NULL, but Image->Flag is EFI\_IMAGE\_TRANSPARENT. | 1. Call DrawImageEx() when Flag is EFI\_HII\_DRAW\_FLAG\_DEFAULT and Blt points to NULL, but Image->Flag is EFI\_IMAGE\_TRANSPARENT, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.9.4.6 | 0x3ac875ed, 0x46d4, 0x4d1d, 0xac, 0xfe, 0xdb, 0x37, 0xe5, 0xf1, 0xb7, 0xd0 | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageEx() - DrawImageEx() return EFI\_SUCCESS with valid parameters. | 1. Call DrawImageEx() when Flag is EFI\_HII\_DRAW\_FLAG\_FORCE\_OPAQUE, Blt is NULL and other valid parameters, the return status should be EFI\_SUCCESS. |
| 5.18.9.4.7 | 0x16a8be, 0x4466, 0x4777, 0xa0, 0xbd, 0xa9, 0x10, 0x1c, 0x54, 0x19, 0xa0 | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageEx() - DrawImageEx() return EFI\_SUCCESS with valid parameters. | 1. Call DrawImageEx() when Flag is the valid combination, Blt is NULL and other valid parameters, the return status should be EFI\_SUCCESS. |

### DrawImageIdEx()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.9.5.1 | 0x24ddcd2b, 0xa9d8, 0x4ec5, 0xaf, 0xf6, 0x77, 0xf3, 0x69, 0x8c, 0xe, 0x19 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() returns EFI\_NOT\_FOUND when PackageList is not in Database. | 1. Call DrawImageIdEx() when PackageList is not in Database, the return status should be EFI\_NOT\_FOUND. |
| 5.18.9.5.2 | 0x8f114d30, 0x684d, 0x402e, 0xb5, 0x35, 0x74, 0x34, 0x1e, 0xbb, 0x88, 0x5f | EFI\_HII\_IMAGE\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() returns EFI\_NOT\_FOUND when PackageList is NULL. | 1. Call DrawImageIdEx() when PackageList is NULL, the return status should be EFI\_NOT\_FOUND. |
| 5.18.9.5.3 | 0x446d5d03, 0xf2b6, 0x4627, 0xad, 0xd1, 0x75, 0x6d, 0xfe, 0xe9, 0x18, 0x3f | EFI\_HII\_IMAGE\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() returns EFI\_NOT\_FOUND when ImageId is invalid. | 1. Call DrawImageIdEx() when ImageId is invalid, the return status should be EFI\_NOT\_FOUND. |
| 5.18.9.5.4 | 0x6dbc9f6e, 0x2694, 0x44ec, 0x99, 0xe9, 0x2d, 0x67, 0x6a, 0xfe, 0x9f, 0x37 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() returns EFI\_NOT\_FOUND when PackageList is NULL. | 1. Call DrawImageIdEx() when PackageList is invalid, the return status should be EFI\_NOT\_FOUND. |
| 5.18.9.5.5 | 0x8c43a76, 0x7f57, 0x41dd, 0x87, 0x99, 0x13, 0xcf, 0xf2, 0x5, 0x9b, 0x6 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() returns EFI\_INVALID\_PARAMETER when Flags is EFI\_HII\_DRAW\_FLAG\_FORCE\_TRANS and Blt is NULL. | 1. Call DrawImageIdEx() when Flags is EFI\_HII\_DRAW\_FLAG\_FORCE\_TRANS and Blt is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.9.5.6 | 0x39787a10, 0x1204, 0x41a5, 0xa8, 0xdb, 0xd3, 0xe9, 0x83, 0xc4, 0x47, 0x44 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() returns EFI\_INVALID\_PARAMETER when Flags is EFI\_HII\_DRAW\_FLAG\_CLIP and Blt points to NULL. | 1. Call DrawImageIdEx() when Flags is EFI\_HII\_DRAW\_FLAG\_CLIP and Blt points to NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.9.5.7 | 0x82c37f35, 0xbca3, 0x494e, 0x8a, 0xdb, 0xf6, 0xd8, 0xf0, 0x7a, 0xf6, 0xe3 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() returns EFI\_INVALID\_PARAMETER when Flags is EFI\_HII\_DRAW\_FLAG\_DEFAULT, Blt points to NULL and Image->Flags is EFI\_IMAGE\_TRANSPARENT. | 1. Call DrawImageIdEx() when Flags is EFI\_HII\_DRAW\_FLAG\_DEFAULT, Blt points to NULL and Image->Flags is EFI\_IMAGE\_TRANSPARENT, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.18.9.5.8 | 0x1c03d9b0, 0x8d9c, 0x40bf, 0x94, 0xa7, 0xa7, 0x85, 0xa3, 0x52, 0xa2, 0x68 | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() return EFI\_SUCCESS with valid parameters. | 1. Call DrawImageIdEx() when Flag is EFI\_HII\_DRAW\_FLAG\_FORCE\_OPAQUE, Blt is NULL and other valid parameters, the return status should be EFI\_SUCCESS. |
| 5.18.9.5.9 | 0x5ee23086, 0xe0ee, 0x4cc8, 0x85, 0xf2, 0x5a, 0xd3, 0x52, 0xd7, 0x4d, 0xb7 | EFI\_HII\_FONT\_EX\_PROTOCOL. DrawImageIdEx() - DrawImageIdEx() return EFI\_SUCCESS with valid parameters. | 1. Call DrawImageIdEx() when Flag is the valid combination, Blt is NULL and other valid parameters, the return status should be EFI\_SUCCESS. |

### GetImageInfo()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.18.9.3.1 | 0xe88ca946, 0xed6d, 0x415d, 0x85, 0x55, 0x0, 0x27, 0x9f, 0x14, 0xc3, 0xf9 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_NOT\_FOUND when ImageId is invalid. | 0x7baa464a, 0x572c, 0x4fa9, 0x80, 0xa3, 0x99, 0xa0, 0x61, 0xc0, 0x46, 0x4f |
| 5.18.9.6.2 | 0xf61dfb48, 0x1c77, 0x4907, 0x9f, 0xab, 0x43, 0x93, 0x17, 0x8c, 0x99, 0xee | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_INVALID\_PARAMETER when Image is NULL. | 1. Call GetImageInfo() when Image is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| EFI\_HII\_FONT\_EX\_PROTOCOL. StringIdToImageEx() - StringIdToImageEx() returns EFI\_INVALID\_PARAMETER when PackageList is NULL. | 1. Call StringIdToImageEx() when PackageList is NULL, the return status should be EFI\_INVALID\_PARAMETER. | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_INVALID\_PARAMETER when ImageId is 0. | 1. Call GetImageInfo() when ImageId is 0, the return status should be EFI\_INVALID\_PARAMETER. |
| 1. Call StringIdToImageEx() when PackageList is not in Database, the return status should be EFI\_NOT\_FOUND. | 0x9cf6b34c, 0x4d53, 0x464e, 0x99, 0x4e, 0xd0, 0x3, 0xb5, 0x7b, 0x8b, 0x67 | EFI\_HII\_IMAGE\_EX\_PROTOCOL. GetImageInfo() - GetImageInfo() returns EFI\_SUCCESS with valid parameters. | 1. Call GetImageInfo() with valid parameters, the return status should be EFI\_SUCCESS. |