# Protocols EFI Loaded Image Test

## EFI\_LOADED\_IMAGE Protocol Test

Reference Document:

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

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.3.1.1.1 | 0xb324a56f, 0x5714, 0x44b4, 0xa2, 0x0f, 0x6e, 0x9b, 0x13, 0x7b, 0x8d, 0xf9 | EFI\_LOADED\_IMAGE\_PROTOCOL – BS.HandleProtocol() to handle Loaded Image Protocol returns EFI\_SUCCESS. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle. The return code should be EFI\_SUCCESS. |
| 5.3.1.1.2 | 0xbce0c845, 0x4ce1, 0x4c3b, 0x9f, 0x94, 0x84, 0x6c, 0x27, 0x9c, 0x93, 0xd0 | EFI\_LOADED\_IMAGE\_PROTOCOL – Revision is equal to EFI\_IMAGE\_INFORMATION\_REVISION | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. Revision on each image handle should equal EFI\_IMAGE\_INFORMATION\_REVISION. |
| 5.3.1.1.3 | 0x12b28b7b, 0x8255, 0x4fad, 0xb3, 0x05, 0x81, 0x31, 0x16, 0x71, 0xb2, 0xe1 | EFI\_LOADED\_IMAGE\_PROTOCOL – ParentHandle is equal to the test driver’s image handle | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. ParentHandle should be equal to the test driver’s image handle. |
| 5.3.1.1.4 | 0xb8e8ce9f, 0x3324, 0x4134, 0xab, 0x08, 0x3f, 0x3c, 0x9e, 0xe2, 0x5c, 0x27 | EFI\_LOADED\_IMAGE\_PROTOCOL – SystemTable is not NULL. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. SystemTable should not be NULL. |
| 5.3.1.1.5 | 0x3bf1e23d, 0x86e1, 0x4f8a, 0x8c, 0x1a, 0x7f, 0xdc, 0x5c, 0x49, 0x11, 0xb9 | EFI\_LOADED\_IMAGE\_PROTOCOL – **DeviceHandle** is not NULL. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. **DeviceHandle** should not be NULL. |
| 5.3.1.1.6 | 0x7df05248, 0x72ff, 0x40a5, 0x94, 0x8c, 0xc6, 0x47, 0xd1, 0xfd, 0xc1, 0xae | EFI\_LOADED\_IMAGE\_PROTOCOL - ImageBase is not NULL and ImageSize is not 0. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. ImageBase is not NULL and ImageSize is not 0. |
| 5.3.1.1.7 | 0xfede5dd0, 0x92f6, 0x42de, 0x81, 0x4f, 0xf2, 0xe3, 0x33, 0x9b, 0x5d, 0xe1 | EFI\_LOADED\_IMAGE\_PROTOCOL - Application image’s ImageCodeType equals **EfiLoaderCode** and ImageDataType equals **EfiLoaderData**. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. Application image’s ImageCodeType should be **EfiLoaderCode** and ImageDataType should be **EfiLoaderData**. |
| 5.3.1.1.8 | 0x9ead501b, 0x4a09, 0x4c24, 0xba, 0x47, 0xcf, 0x27, 0xbf, 0xf0, 0x66, 0xdb | EFI\_LOADED\_IMAGE\_PROTOCOL - **BootService** image’s ImageCodeType equals **EfiBootServiceCode** and ImageDataType equals **EfiBootServiceData**. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. **BootService** image’s ImageCodeType equals **EfiBootServiceCode** and ImageDataType equals **EfiBootServiceData**. |
| 5.3.1.1.9 | 0x064e5c37, 0xcfaf, 0x4b5a, 0xa2, 0xa0, 0xf6, 0x17, 0xdd, 0x41, 0xa4, 0x12 | EFI\_LOADED\_IMAGE\_PROTOCOL - **RuntimeService** image’s ImageCodeType equals **EfiRuntimeServiceCode** and ImageDataType equals **EfiRuntimeServiceData**. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. **RuntimeService** image’s ImageCodeType equals **EfiRuntimeServiceCode** and ImageDataType equals **EfiRuntimeServiceData**. |
| 5.3.1.1.10 | 0xc7606256, 0x8a89, 0x48ce, 0xb5, 0x7b, 0xa1, 0xb0, 0x6b, 0x3c, 0x62, 0x3b | EFI\_LOADED\_IMAGE\_PROTOCOL – Unload() is NULL if the image has no Unload function. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. Check on Application Images which have no unload function. Unload field should be NULL. |
| 5.3.1.1.11 | 0xfc2330ce, 0xaa7a, 0x4c64, 0xac, 0x5e, 0xfe, 0xb1, 0xf0, 0xf7, 0xda, 0xc7 | EFI\_LOADED\_IMAGE\_PROTOCOL – Unload() is not NULL and its address is valid if the image has Unload function. | 1. Call BS.LoadImage() to get image handle by filename.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. Check on Application Images which have Unload function. Unload field should be valid and its entry address should be within the range of [ImageBase, ImageBase+ImageSize] |
| 5.3.1.1.12 | 0x69cb9798, 0x5b57, 0x4381, 0xb9, 0xb2, 0x54, 0xb9, 0xa2, 0x4b, 0x8d, 0x16 | EFI\_LOADED\_IMAGE\_PROTOCOL – LoadOptions is used in notify function. | 1. Call BS.LoadImage() to get image handle by filename with specified LoadOptions.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. Call BS.StartImage(). LoadOptions should be used. |
| 5.3.1.1.13 | 0x6da9aef4, 0xdadd, 0x4bda, 0xa7, 0x0d, 0x29, 0x47, 0x0e, 0x05, 0xf3, 0x17 | EFI\_LOADED\_IMAGE\_PROTOCOL – LoadOptions is used in notify function. | 1. Call BS.LoadImage() to get image handle by filename with specified LoadOptions.  2. Call BS.HandleProtocol() to handle Loaded Image Protocol on each image handle.  3. Call BS.StartImage(). LoadOptions should be used.  4. Unload Image.  5. Change LoadOptions and call BS.**LoadImage()** again.  6. Call BS.HandleProtocol() and BS.StartImage(). Updated LoadOptions value should be used. |
| 5.3.1.1.14 | 0x0caae7f5, 0x0742, 0x458f, 0xbf, 0x02, 0x65, 0x2d, 0x33, 0xa4, 0xf1, 0xab | EFI\_LOADED\_IMAGE\_PROTOCOL – SystemTable is not NULL | 1. Check on all images in system. SystemTable should not be NULL. |
| 5.3.1.1.15 | 0xa7bc2e01, 0x3162, 0x482c, 0xa6, 0x8b, 0x93, 0x9d, 0x0c, 0xf7, 0x9a, 0x45 | EFI\_LOADED\_IMAGE\_PROTOCOL – ImageBase is not NULL and ImageSize is not 0 | 1. Check on all images in system. ImageBase is not NULL and ImageSize is not 0. |
| 5.3.1.1.16 | 0xa3ada89a, 0xef4e, 0x475b, 0xbc, 0x53, 0x00, 0x98, 0xd5, 0xc6, 0x5b, 0xee | EFI\_LOADED\_IMAGE\_PROTOCOL – ImageCodeType matches with the ImageDataType. | 1. Check on all images in system.  If ImageCodeType is **EfiLoaderCode**, ImageDataType should be **EfiLoaderData**;  If ImageCodeType is **EfiBootServicesCode**, ImageDataType should be **EfiBootServicesData**;  If ImageCodeType is **EfiRuntimeServicesCode**, ImageDataType should be **EfiRuntimeServicesData**; |
| 5.3.1.1.17 | 0xda215e1d, 0x5ac8, 0x480a, 0xa7, 0x9e, 0xa0, 0x66, 0xb9, 0x74, 0x58, 0x65 | EFI\_LOADED\_IMAGE\_PROTOCOL – If Unload() function is not NULL, its address is valid. | 1. Check on all images in system. If Unload() function is not NULL, its address should be within the range of [ImageBase, ImageBase+ImageSize] |
| 5.3.1.1.18 | 0xe2f6c4a6, 0xe2a8, 0x4bab, 0x94, 0xbb, 0x70, 0x44, 0x54, 0xd6, 0x2a, 0xea | EFI\_LOADED\_IMAGE\_PROTOCOL – Revision equals EFI\_IMAGE\_INFORMATION\_REVISION. | 1. Check Revision on all file images. Revision should be equal to EFI\_IMAGE\_INFORMATION\_REVISION |