# Protocols SCSI Bus Support Test

## EFI\_SCSI\_IO\_PROTOCOL Function Test

Reference Document:

*UEFI Specification*, EFI\_SCSI\_IO\_PROTOCOL Section..

### GetDeviceType() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.2.1.1 | 0xa9b53582, 0xcbd5, 0x4934, 0x85, 0x95, 0x2e, 0x4d, 0xc6, 0x8a, 0xb1, 0x34 | EFI\_SCSI\_IO\_PROTOCOL.GetDeviceType – GetDeviceType() should return EFI\_SUCCESS with SCSI device correctly installed | Call GetDeviceType().  The return status should be EFI\_SUCCESS. |

### GetDeviceLocation() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.2.2.1 | 0x2d1db8e2, 0xb4d3, 0x4bbf, 0x80, 0xa6, 0x4c, 0x15, 0xef, 0x54, 0x87, 0x31 | EFI\_SCSI\_IO\_PROTOCOL.GetDeviceLocation – GetDeviceLocation() should return EFI\_SUCCESS after setting Target and Lun. | Call GetDeviceLocation() with valid Target and Lun.  The return status should be EFI\_SUCCESS. |

### ResetBus() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.2.3.1 | 0xb11aec12, 0x0ffb, 0x46da, 0x82, 0x37, 0xaa, 0xa0, 0xed, 0x46, 0x29, 0x05 | EFI\_SCSI\_IO\_PROTOCOL.ResetBus - ResetBus() should return EFI\_SUCCESS or EFI\_UNSUPPORTED with SCSI device correctly installed. | Call ResetBus() after SCSI device correctly installed.  The return status should be EFI\_SUCCESS or EFI\_UNSUPPORTED. |

### ResetDevice() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.2.4.1 | 0x05720e96, 0xf8ab, 0x46f5, 0xbc, 0xf9, 0xc9, 0x24, 0x51, 0x1c, 0xd5, 0x44 | EFI\_SCSI\_IO\_PROTOCOL.ResetDevice - ResetDevice() should return EFI\_SUCCESS or EFI\_UNSUPPORTED with SCSI device correctly installed. | Call ResetDevice() after SCSI device correctly installed.  The return status should be EFI\_SUCCESS or EFI\_UNSUPPORTED. |

### ExecuteScsiCommand () Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.2.5.1 | 0xaf88a458, 0xdeab, 0x4744, 0xae, 0xf5, 0xe4, 0x1c, 0xb1, 0x0e, 0xbb, 0xb3 | EFI\_SCSI\_IO\_PROTOCOL.ExecuteScsiCommand - Invokes ExecuteScsiCommand() with NULL Event will verify interface correctness by returning EFI\_SUCCESS. | Call ExecuteScsiCommand () with NULL Event.  The return status should be EFI\_SUCCESS. |
| 5.9.2.5.2 | 0x96789d65, 0x11e6, 0x4a2d, 0xbb, 0x5b, 0xe3, 0x3d, 0x22, 0x6b, 0x28, 0xf1 | EFI\_SCSI\_IO\_PROTOCOL.ExecuteScsiCommand - Invokes ExecuteScsiCommand () with Event verifies interface correctness. | Call ExecuteScsiCommand () with Event.  The return status should be EFI\_SUCCESS and the event should be invoked. |

## EFI\_SCSI\_IO\_PROTOCOL Conformance Test

Reference Document:

*UEFI Specification*, EFI\_SCSI\_IO\_PROTOCOL Section.

### GetDeviceType() Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.3.1.1 | 0x37a8da14, 0x170a, 0x4620, 0xaa, 0xea, 0x26, 0x6f, 0x35, 0x8f, 0x0c, 0x75 | EFI\_SCSI\_IO\_PROTOCOL.GetDeviceType – GetDeviceType() should return EFI\_INVALID\_PARAMETER with DeviceType set NULL. | Call GetDeviceType() with a DeviceType value of NULL.  The return status should be EFI\_INVALID\_PARAMETER. |

### GetDeviceLocation() Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.3.2.1 | 0x6937c784, 0xb044, 0x4828, 0xb8, 0x77, 0xff, 0xc7, 0x35, 0x8f, 0xf2, 0xaa | EFI\_SCSI\_IO\_PROTOCOL.GetDeviceLocation – GetDeviceLocation() should return EFI\_INVALID\_PARAMETER with Target set NULL. | Call GetDeviceLocation () with a Target value ofNULL.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.3.2.2 | 0x6a48edf9, 0x8a3b, 0x4e9c, 0xb7, 0x6f, 0x37, 0x45, 0x83, 0xc7, 0xdc, 0x2b | EFI\_SCSI\_IO\_PROTOCOL.GetDeviceLocation – GetDeviceLocation() should return EFI\_INVALID\_PARAMETER with Lun set NULL. | Call GetDeviceLocation () with a Lun value of NULL.  The return status should be EFI\_INVALID\_PARAMETER. |

### ExecuteScsiCommand () Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.3.3.1 | 0x17503bd1, 0x4d36, 0x4183, 0x9f, 0xf1, 0x9d, 0x0f, 0xc2, 0x21, 0x33, 0x26 | EFI\_SCSI\_IO\_PROTOCOL. ExecuteScsiCommand - Calling ExecuteScsiCommand () with an too long InTransferLength value and NULL Event returns EFI\_BAD\_BUFFER\_SIZE. | Call ExecuteScsiCommand () with an InTransferLength value larger than the length which SCSI controller can handle.  The return status should be EFI\_BAD\_BUFFER\_SIZE and InTransferLength will be updated to the length that SCSI controller be able to handle. |
| 5.9.3.3.2 | 0x8c27b8c2, 0x2c40, 0x4f6a, 0xbb, 0x54, 0x26, 0x5d, 0x12, 0x9a, 0x97, 0xce | EFI\_SCSI\_IO\_PROTOCOL. ExecuteScsiCommand - Calling ExecuteScsiCommand () with invalid Packet and NULL Event returns EFI\_INVALID\_PARAMETER. | Call ExecuteScsiCommand () with invalid Packet .  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.3.3.3 | 0xbeb81209, 0x808d, 0x46d1, 0xa2, 0x36, 0x23, 0x7f, 0x17, 0x22, 0x30, 0x37 | EFI\_SCSI\_IO\_PROTOCOL. ExecuteScsiCommand - Calling ExecuteScsiCommand () with an too long InTransferLength value and no NULL Event returns EFI\_BAD\_BUFFER\_SIZE. | Call ExecuteScsiCommand () with an InTransferLength value larger than the length which SCSI controller can handle.  The return status should be EFI\_BAD\_BUFFER\_SIZE and InTransferLength will be updated to the length that SCSI controller be able to handle. |
| 5.9.3.3.4 | 0x994fd5e2, 0x2d39, 0x4fa9, 0xa7, 0x4f, 0x8d, 0x09, 0xe0, 0xb6, 0x84, 0x1c | EFI\_SCSI\_IO\_PROTOCOL. ExecuteScsiCommand - Calling ExecuteScsiCommand () with invalid Packet and no NULL Event returns EFI\_INVALID\_PARAMETER. | Call ExecuteScsiCommand () with invalid Packet .  The return status should be EFI\_INVALID\_PARAMETER. |

## EFI\_EXT\_SCSI\_PASS\_PROTOCOL Function Test

Reference Document:

UEFI Specification, EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL Section.

### GetNextTargetLun() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.4.1.1 | 0x4f658292, 0xa409, 0x4d67, 0xba, 0x13, 0x04, 0xc2, 0x51, 0x85, 0xf2, 0x80 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.GetNextTargetLun – GetNextTargetLun() retrieves the list of legal Target IDs and LUNs for SCSI devices on a SCSI channel. | Call GetNextTargetLun () with a Target value of 0xFF’s to get the first SCSI device present on a SCSI channel.  Use the values of Target andLun values that are returned to get the next SCSI device until the end.  Every call of GetNextTargetLun() should return EFI\_SUCCESS except the last one.  The last call should return EFI\_NOT\_FOUND. |

### BuildDevicePath() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.4.2.1 | 0x130d44b6, 0xce53, 0x42b6, 0x9b, 0xa6, 0x3d, 0x11, 0x5d, 0x49, 0x2b, 0x33 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.BuildDevicePath - Invoking BuildDevicePath() will verify interface correctness by returning EFI\_SUCCESS. | Call GetNextTargetLun() to get the first device’s Target and Lun.  Call BuildDevicePath() with a valid parameter. Free the DevicePath.  The return status should be EFI\_SUCCESS. |

### GetTargetLun() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.4.3.1 | 0x6ea827e4, 0x522c, 0x44b6, 0x99, 0xe4, 0x25, 0x93, 0x19, 0xba, 0xcc, 0x57 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.GetTargetLun - Invoking GetTargetLun() will verify interface correctness by returning EFI\_SUCCESS. | Call GetNextTargetLun() and BuildDevicePath() to get the valid DevicePath.  Use this DevicePath to call GetTargetLun().  The return value should be EFI\_SUCCESS. |

### ResetChannel() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.4.4.1 | 0x4e0080d2, 0x4065, 0x4b92, 0xa4, 0x61, 0x52, 0x49, 0xf3, 0x8f, 0xaf, 0x55 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.ResetChannel - Invoking ResetChannel() will verify interface correctness via return code of EFI\_SUCCESS or EFI\_UNSUPPORTED. | Call ResetChannel().  The return value should be EFI\_SUCCESS or EFI\_UNSUPPORTED. |

### ResetTargetLun() Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.4.5.1 | 0x9400bc81, 0x9e48, 0x469b, 0xa0, 0x97, 0xd0, 0x08, 0x45, 0xb6, 0x69, 0xe8 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.ResetTargetLun - Invoking ResetTargetLun() will verify interface correctness via return code of EFI\_SUCCESS or EFI\_UNSUPPORTED. | Call GetNextTargetLun() to get valid Target and Lun.  Use the Target and Lun values that are returned to call ResetTargetLun().  The return value should be EFI\_SUCCESS or EFI\_UNSUPPORTED. |

### GetNextTarget () Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.4.6.1 | 0xc89631f3, 0xbd59, 0x4959, 0xba, 0x10, 0x3f, 0xa9, 0x94, 0x62, 0x02, 0xdf | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.    �GetNextTarget – GetNextTarget() retrieves the list of legal Target IDs for SCSI devices on a SCSI channel. | Call GetNextTarget () with a Target value of 0xFF’s to get the first SCSI device present on a SCSI channel.  Use the Target value that is returned to get the next SCSI device until the end.  Every call of GetNextTarget () should return EFI\_SUCCESS except the last one.  The last call should return EFI\_NOT\_FOUND. |

### PassThru () Function

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.4.7.1 | 0xdb7841b9, 0x2a4a, 0x45b1, 0xa9, 0x9f, 0x67, 0x7a, 0xb4, 0xcd, 0x79, 0xa2 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Invoking PassThru() with NULL Event will verify interface correctness by returning EFI\_SUCCESS. | Call GetNextDevice() to get valid Target and Lun values.  Call PassThru()with the returned values of Target, Lun , and a NULL Event.  The return status should be EFI\_SUCCESS. |
| 5.9.4.7.2 | 0x4787ed6f, 0xa984, 0x4b15, 0xb2, 0xf3, 0xa0, 0xd1, 0xb8, 0xce, 0x61, 0x89 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Invoking PassThru() with Eventwill verify interface correctness by returning EFI\_SUCCESS. | Call GetNextDevice() to get valid Target and Lun values.  Call PassThru()with the returned value of Target, Lun and a **Event**.  The return status should be EFI\_SUCCESS and the event should be invoked. |

## EFI\_EXT\_SCSI\_PASS\_PROTOCOL Conformance Test

Reference Document:

*UEFI Specification*, EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL Section.

### GetNextTargetLun() Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.5.1.1 | 0xaad50e59, 0x9423, 0x427d, 0xa7, 0x5d, 0x69, 0x1c, 0x90, 0xb7, 0xf9, 0x75 | EFI\_SCSI\_PASS\_THRU\_PROTOCOL.GetNextTargetLun - Call **GetNextTargetLun()** with an invalid Target. | Call GetNextTargetLun() with Target’s all bits are 1 to get the first device.  Call GetNextTargetLun() with an invalid Target.  It should return EFI\_INVALID\_PARAMETER. |
| 5.9.5.1.2 | 0xb3e87aa1, 0x6e9c, 0x478f, 0x9b, 0xd5, 0x39, 0x50, 0x08, 0x01, 0x28, 0x96 | EFI\_SCSI\_PASS\_THRU\_PROTOCOL.GetNextTargetLun - Call **GetNextTargetLun()** with an invalid Lun. | Call GetNextTargetLun() with Target‘s all bits are 1 to get the first device.  Call GetNextTargetLun() with an invalid Lun.  It should return EFI\_INVALID\_PARAMETER. |

### BuildDevicePath() Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.5.2.1 | 0x942a0e01, 0x7b80, 0x46e4, 0xa7, 0x57, 0x86, 0xc4, 0xec, 0x53, 0xf4, 0xe4 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.BuildDevicePath - Calling BuildDevicePath() with an invalid Target returns EFI\_NOT\_FOUND. | Call BuildDevicePath() with an invalid Target.  The return status should be EFI\_NOT\_FOUND. |
| 5.9.5.2.2 | 0x222f00c1, 0xf6bf, 0x41ed, 0xae, 0xfd, 0xaa, 0xc4, 0x8f, 0x3f, 0xa9, 0xdb | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.BuildDevicePath - Calling BuildDevicePath() with invalid Lun returns EFI\_NOT\_FOUND. | Call BuildDevicePath() with invalid Lun.  The return status should be EFI\_NOT\_FOUND. |
| 5.9.5.2.3 | 0xc72e6a78, 0x5292, 0x4493, 0x90, 0x40, 0xb0, 0x44, 0x5a, 0x9c, 0x17, 0x14 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.BuildDevicePath - Calling BuildDevicePath() with NULL DevicePath returns EFI\_INVALID\_PARAMETER. | Call BuildDevicePath() with NULL DevicePath.  The return status should be EFI\_INVALID\_PARAMETER. |

### GetTargetLun() Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.5.3.1 | 0xff2f0849, 0x690b, 0x48ea, 0x8e, 0x35, 0x64, 0x36, 0x3f, 0xaa, 0x8c, 0x5c | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.GetTargetLun - Invoking GetTargetLun() with NULL DevicePath returns EFI\_INVALID\_PARAMETER. | Call GetTargetLun() with NULL DevicePath.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.3.2 | 0x6602bd0a, 0x1c05, 0x49e5, 0xa8, 0xd4, 0xc6, 0x03, 0x8c, 0x43, 0x9a, 0xf9 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.GetTargetLun - Invoking GetTargetLun() with NULL Target returns EFI\_INVALID\_PARAMETER. | Call GetTargetLun() with NULL Target.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.3.3 | 0x1b64d49a, 0x1f1b, 0x4610, 0xa2, 0x66, 0xde, 0x32, 0xa1, 0x07, 0x2b, 0x32 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.GetTargetLun - Invoking GetTargetLun() with NULL Lun returns EFI\_INVALID\_PARAMETER. | Call GetTargetLun() with NULL Lun.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.3.4 | 0xf7830eaf, 0xba30, 0x4224, 0xab, 0xc4, 0x42, 0x42, 0x8b, 0x7a, 0x04, 0x5d | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.GetTargetLun - Calling GetTargetLun() with unsupported DevicePath returns EFI\_UNSUPPORTED. | Call GetTargetLun() with unsupported DevicePath.  The return status should be EFI\_UNSUPPORTED. |

### ResetTargetLun() Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.5.4.1 | 0x106ae2fc, 0x3f34, 0x4afe, 0x82, 0x44, 0x40, 0x27, 0x57, 0x60, 0x98, 0x31 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.ResetTargetLun - Calling ResetTargetLun() with an invalid Targetreturns EFI\_INVALID\_PARAMETER. | Call GetResetTargetLun() with an invalid Target.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.4.2 | 0xc9378047, 0x9b4b, 0x4abf, 0xaa, 0x6b, 0xe3, 0xcd, 0xb6, 0xc4, 0x19, 0x39 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.ResetTargetLun - Calling ResetTargetLun() with an invalid Lun returns EFI\_INVALID\_PARAMETER. | Call GetResetTargetLun() with an invalid Lun.  The return status should be EFI\_INVALID\_PARAMETER. |

### GetNextTarget () Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.5.5.1 | 0xb564ad60, 0x32ce, 0x4f5f, 0x86, 0x7a, 0xef, 0x9f, 0xef, 0x5e, 0x94, 0xa2 | EFI\_SCSI\_PASS\_THRU\_PROTOCOL.GetNextTarget - Call **GetNextTarget()** with an invalid Target | Call GetNextTarget() with an invalid Target.  The return status should be EFI\_INVALID\_PARAMETER. |

### PassThru() Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.9.5.6.1 | 0x6d6fcacd, 0x3463, 0x41c8, 0xa5, 0x01, 0xa2, 0x99, 0x40, 0x44, 0x59, 0xb8 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Calling PassThru() with an too long InTransferLength and NULL Event returns EFI\_BAD\_BUFFER\_SIZE. | Call PassThru() with an InTransferLength larger than the SCSI controller can handle.  The return status should be EFI\_BAD\_BUFFER\_SIZE and the InTransferLength will be updated to the length that SCSI controller can handle. |
| 5.9.5.6.2 | 0x645295b5, 0xc36b, 0x4b23, 0xaf, 0xc7, 0xd4, 0xcc, 0xc0, 0x1d, 0xb6, 0x4f | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Calling PassThru() with an invalid Target and NULL Event returns EFI\_INVALID\_PARAMETER. | Call PassThru() with an invalid Target.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.6.3 | 0x9f9489a2, 0x23f3, 0x4962, 0x9d, 0x8f, 0xd2, 0xc0, 0xa7, 0xcb, 0x2f, 0xb1 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Calling PassThru() with an invalid Lun and NULL Event returns EFI\_INVALID\_PARAMETER. | Call PassThru() with an invalid Lun.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.6.4 | 0xc584b074, 0xa8cd, 0x438c, 0xb5, 0x18, 0xb1, 0xec, 0x59, 0xfa, 0xc8, 0xee | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Calling PassThru() with invalid Packet content and NULL Event returns EFI\_INVALID\_PARAMETER. | Call PassThru() with invalid Packet content.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.6.5 | 0x3cd806fd, 0x3742, 0x44e9, 0xa6, 0x19, 0xdf, 0x2d, 0x37, 0x47, 0xe7, 0x8f | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Calling PassThru() with an too long InTransferLength and no NULL Event returns EFI\_BAD\_BUFFER\_SIZE. | Call PassThru() with an InTransferLength larger than the SCSI controller can handle.  The return status should be EFI\_BAD\_BUFFER\_SIZE and the InTransferLength will be updated to the length that SCSI controller can handle. |
| 5.9.5.6.6 | 0x9648ab45, 0x898b, 0x4b44, 0xab, 0x9e, 0x24, 0x6b, 0xc6, 0x49, 0xc9, 0xfd | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Calling PassThru() with an invalid Target and no NULL Event returns EFI\_INVALID\_PARAMETER | Call PassThru() with an invalid Target.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.6.7 | 0x8662da7d, 0x6f98, 0x4051, 0xb1, 0x87, 0x85, 0xb0, 0xf4, 0xb5, 0x3a, 0xf1 | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Calling PassThru() with an invalid Lun and no NULL Event returns EFI\_INVALID\_PARAMETER. | Call PassThru() with an invalid Lun.  The return status should be EFI\_INVALID\_PARAMETER. |
| 5.9.5.6.8 | 0xf9ec9bf2, 0x743f, 0x4eed, 0x82, 0xbc, 0x35, 0xf2, 0xcc, 0x56, 0x45, 0xda | EFI\_EXT\_SCSI\_PASS\_THRU\_PROTOCOL.PassThru - Calling PassThru() with invalid Packet content and no NULL Event returns EFI\_INVALID\_PARAMETER. | Call PassThru() with invalid Packet content.  The return status should be EFI\_INVALID\_PARAMETER. |