# Random Number Generator Protocols

## EFI\_RNG\_PROTOCOL Test

Reference Document:

*UEFI Specification*, EFI\_RNG\_PROTOCOL Section.

### GetInfo ()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.33.4.1.1 | 0xb0aeea8a, 0xcd05, 0x4254, 0xb2, 0xcb, 0x30, 0xbb, 0x90, 0x87, 0x73, 0xc6 | **EFI\_RNG\_PROTOCOL.GetInfo() - GetInfo()** returns EFI\_SUCCESS with valid parameters. | Call GetInfo() to get the  RNGAlgorithmListSize.  Allocate a list buffer with the  RNGAlgorithmListSize gotten from step1.  3. Call GetInfo() with the new allocated buffer, the return status should be EFI\_SUCCESS. |
| 5.33.4.1.2 | 0x50df54e5, 0x1449, 0x4a34, 0x95, 0x6a, 0xb6, 0x61, 0x66, 0xc2, 0xd5, 0x8a | EFI\_RNG\_PROTOCOL.GetInfo() - GetInfo() returns valid algorithm with valid parameters. | Call GetInfo() to get the  RNGAlgorithmListSize.  Allocate a list buffer with the  RNGAlgorithmListSize gotten from step1.  Call GetInfo() with the new allocated  buffer, the return status should be EFI\_SUCCESS.  Compare the Algorithm gotten from  Step3 with the given algorithms, the result should be success. |
| 5.33.4.1.3 | 0x0db3b0d2, 0x859f, 0x4682, 0x87, 0x67, 0x62, 0x35, 0x67, 0x91, 0xb7, 0x9d | EFI\_RNG\_PROTOCOL.GetInfo() - GetInfo() returns EFI\_BUFFER\_TOO\_SMALL with small RNGAlgorithmListSize and returns valid size | Call GetInfo() with small RNGAlgorithmListSize, the return  status should be EFI\_BUFFER\_TOO\_SMALL and returns valid size |

### GetRNG()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
| 5.33.4.2.1 | 0x4a54a35e, 0x66ac, 0x4c2e, 0x92, 0xd8, 0x7b, 0x26, 0x3d, 0x8a, 0x77, 0xa8 | EFI\_RNG\_PROTOCOL.GetRNG() - GetRNG() returns EFI\_SUCCESS with valid parameters. | Call GetInfo() to get the  RNGAlgorithmListSize.  Allocate a list buffer with the  RNGAlgorithmListSize gotten from step1.  Call GetInfo() with the new allocated  Buffer.  Call GetRNG() with valid parameters,  the return status should be EFI\_SUCCESS. |
| 5.33.4.2.2 | 0xe3d11e22, 0xeddb, 0x40c4, 0x8f, 0x6d, 0x25, 0x79, 0x33, 0xea, 0x62, 0xf8 | EFI\_RNG\_PROTOCOL.GetRNG() - GetRNG() returns EFI\_SUCCESSwith default algorithm. | Call GetRNG() with default algorithm,  the return status should be EFI\_SUCCESS. |
| 5.33.4.2.3 | 0xe79e5379, 0xd4dc, 0x4624, 0x88, 0x05, 0x09, 0x46, 0x1c, 0x09, 0x78, 0x28 | EFI\_RNG\_PROTOCOL.GetRNG() - GetRNG() returns EFI\_INVALID\_PARAMETER when RNGValueLength is 0. | Call GetRNG()when RNGValueLength is 0, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.33.4.2.4 | 0x76ac3e4c, 0x5f59, 0x4c21, 0x82, 0x0a, 0xe4, 0x24, 0xc2, 0xef, 0x36, 0x14 | EFI\_RNG\_PROTOCOL.GetRNG() - GetRNG() returns EFI\_INVALID\_PARAMETERwhen RNGValue is NULL. | Call GetRNG()when RNGValue is NULL, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.33.4.2.5 | 0x27451869, 0x357d, 0x4e92, 0xb8, 0xb0, 0xb8, 0xc5, 0xba, 0xb9, 0xa4, 0xe9 | EFI\_RNG\_PROTOCOL.GetRNG() - GetRNG() returns EFI\_INVALID\_PARAMETERwhen RNGAlgorithm is NULL and RNGValueLength is 0. | Call GetRNG()when RNGAlgorithm is NULL and RNGValueLength is 0, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.33.4.2.6 | 0x31ce0e8, 0x3604, 0x4489, 0x93, 0x6c, 0x60, 0x8c, 0x9b, 0x2c, 0xf8, 0xf4 | EFI\_RNG\_PROTOCOL.GetRNG() - GetRNG() returns EFI\_INVALID\_PARAMETERwhen RNGValueLength is 0 after the RNGAlgorithm is freed. | Call GetRNG()when RNGValue is NULL after the RNGAlgorithm is freed, the return status should be EFI\_INVALID\_PARAMETER. |
| 5.33.4.2.7 | 0x7a4ea182, 0xa4cd, 0x441d, 0x98, 0xd7, 0x73, 0x65, 0x87, 0x6f, 0xfa, 0x77 | EFI\_RNG\_PROTOCOL.GetRNG() - GetRNG() returns EFI\_UNSUPPORTEDwhen RNGAlgorithm is unsupported. | Call GetRNG()when RNGAlgorithm is unsupported, the return status should be EFI\_UNSUPPORTED. |