File Server SUT Test Sheet

Following steps will guide you configure the file server quickly for an interoperability testing.

Note:

* Enable local Guest user on file server (SUT).
* For all the shares created previously, need to grant permissions as following:
* Grant **Full Control** Permissions to admin account
* Grant Permissions **without DELETE** and **GENERIC\_ALL** to nonadmin account
* There’re some default values for test settings, you could just follow them or use your own defined.

|  |  |
| --- | --- |
| **Test Area** | **SUT configuration** |
| **MS-SMB2**  **MS-FSRVP** | * Create SMB2 share for MS-SMB2 basic and MS-FSRVP test: (*default: SMBBasic*). * If SUT supports hosting **Continuous Availability (CA)** share, * Create a CA share: (*default: SMBClustered*) * If SUT supports **encryption**, * Create an encrypted share: (*default: SMBEncrypted*) * If file system supports **symbolic link**: * Create a symbolic link (*default: Symboliclink*)under basic share (e.g. *SMBBasic*) which links to basic share (e.g. *SMBBasic*) * Create a sub folder (*default: Sub*) under basic share (e.g. *SMBBasic*) * Create a symbolic link (*default: Symboliclink2*)under sub folder (e.g. *SMBBasic\Sub*) which also links to the basic share (e.g. *SMBBasic*) * Create a share (*default: ShareForceLevel2*)and set **SHI1005\_FLAGS\_FORCE\_LEVELII\_OPLOCK** in **SHI1005\_flags** if applicable. * If SUT supports **SMB2\_CREATE\_APP\_INSTANCE\_ID** * Create a share (*default: SameWithSMBBasic*)which points to the same local path of basic share (e.g. *SMBBasic*) * Create another share (*default: DifferentFromSMBBasic*) which points to a different local path (e.g. *C:\DifferentFromSMBBasic*). * If SUT supports **ReFS** file system * Create a share SMBReFSShare on ReFS volume. * If SUT supports **asymmetric share**   Create a share on optimum node (*default: SMBClustered*) |
| **Auth** | * If you like to test **Kerberos Authentication**:   Set the computer account password of SUT to (*default: Password04!*)   * If you like to test **Share Permission**:   Create a share named: *(default: AzShare)* with permission:  NTFS Permission: Allow Everyone, Share Permission: Allow Domain Admins   * If you like to test **Folder Permission**:   Create a share named: *(default: AzFolder)* with permission:  NTFS Permission: Allow Domain Admins, Share Permission: Allow Everyone   * If you like to test **File Permission**:   Create a share named: *(default: AzFile)* with permission:  NTFS Permission: Allow Domain Admins, Share Permission: Allow Everyone   * If you like to test **Claim-Based Access Control (CBAC)**:   Create a share named: *(default: AzCBAC)* with permission:  NTFS Permission: Allow Everyone, Share Permission: Allow Everyone |
| **MS-DFSC** | * + Create a SMB2 share for **DFSC** test: (*default*: *FileShare*)   + Create **DFS namespaces**, two Stand-alone namespaces: (*default*: *SMBDfs and Standalone*)Assume your server name is SUT\_NAME below:   Root share for **SMBDfs**: [\\SUT\_NAME\SMBDfs](file://SUT_NAME/SMBDfs) , Root share for **Standalone**: [\\SUT\_NAME\Standalone](file://SUT_NAME/Standalone)   * + Create **Domain-based namespace** if test in domain environment: (*e.g.*: DomainBased)   + Add one folder (*default*: *SMBDfsLink*)to 1st namespace (e.g. *SMBDfs*) and set **link target** to SMB2 share [\\SUT\_NAME\SMBBasic](file://SUT_NAME/SMBBasic)   + Add two folders to 2nd namespace (e.g. *Standalone*)   One is **DFSLink**, link target is [\\SUT\_NAME\FileShare](file://SUT_NAME/FileShare)  The other is **Interlink**, link target is [\\SUT\_NAME\SMBDfs\SMBDfsLink](file://SUT_NAME/SMBDfs/SMBDfsLink)   * + Add two folders to Domain-based namespace (e.g. DomainBased)   One is **DFSLink**, link target is [\\SUT\_NAME\FileShare](file://SUT_NAME/FileShare)  The other is **Interlink**, link target is [\\SUT\_NAME\SMBDfs\SMBDfsLink](file://SUT_NAME/SMBDfs/SMBDfsLink) |
| **MS-FSA** | * + Create a SMB2 share for **FSA** test: (*default*: *FileShare*)   + Create a file in the SMB2 share (*e.g.* *FileShare*), file name (*default*: *ExistingFile.txt)*   + Create a folder*\_\_\_\_\_\_\_\_\_\_(default: ExistingFoler)* in the SMB2 share *(e.g. FileShare)*   + Create a mountpoint*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(default: mountpoint)* in the share*(e.g. FileShare)* mounting to the volume   + Create a symbolic link file \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(default: link.txt) in the share*(e.g. FileShare)* linking to the file(*default*: *ExistingFile.txt)* |
| **MS-RSVD** | * + Set the ptfconfig property: ShareContainingSharedVHD as the path of a share which contains a VHD file. |
| **MS-SQOS** | * + Create a virtual hard disk file in your share, and set the ptfconfig property: SqosVHDFullPath as the full path of the vhd file (default: \\scaleoutfs\SMBClustered\sqos.vhdx)   + Create a new policy (with MinimumIoRate: 100 and MaximumIoRate: 200 and MaximumBandwidth: 1638400), and set the ptfconfig property: SqosPolicyId as the policy id.   Note: If MaximumBandwidth is not supported yet, you can ignore its configuration. |