Application Programming Interfaces

EFS provides the following API set to expose its features. These APIs are used by various tools like Explorer, Cipher, NTBackup, EDRP Policy snap-in that expose EFS capabilities to end users and administrators.

1. EncryptFile encrypts a plaintext file represented by lpFileName. The file may be local or remote.

   The syntax is

```
EncryptFile
    BOOL
    EncryptFile(
        LPCTSTR lpFileName
    );
```

2. DecryptFile decrypts an encrypted file represented by lpFileName. The file may be local or remote.

```
DecryptFile
    BOOL
    DecryptFile(
        LPCTSTR lpFileName,
        DWORD   dwReserved
    );
```

3. FileEncryptionStatus returns TRUE if the file is encryptable. A file is not encryptable if it is not on the NTFS version 5 file system, if it is marked System, and so forth.

```
FileEncryptionStatus
    BOOL
    FileEncryptionStatus(
        LPCTSTR lpFileName,
        LPDWORD  lpStatus
    );
```

4. QueryUsersOnEncryptedFile returns the information on the list of users who can decrypt the file represented by lpFileName. The information returned contains a security identifier (SID) of users (if available), user's name from the certificate that was used and a thumbprint of the certificate.

```
QueryUsersOnEncryptedFile
    DWORD
    QueryUsersOnEncryptedFile(
    IN LPCTSTR lpFileName,OUT
    PENCRYPTION_CERTIFICATE_HASH_LIST *
        pUsers
```

```
        );
QueryRecoveryAgentsOnEncryptedFile
        DWORD
QueryRecoveryAgentsOnEncryptedFile(
        IN LPCTSTR lpFileName,
        OUT PENCRYPTION_CERTIFICATE_HASH_LIST *
                pRecoveryAgents
);
```

5. QueryRecoveryAgentsOnEncryptedFile returns the information on the list of recovery agents who can recover the encrypted file represented by lpFileName. The information returned contains a SID of recovery agents (if available), their names from the certificates and the thumbprint of the certificates.

```
RemoveUsersFromEncryptedFile
        DWORD
        RemoveUsersFromEncryptedFile(
                IN LPCTSTR lpFileName,
                IN PENCRYPTION_CERTIFICATE_HASH_LIST
                        pHashes
        );
```

6. RemoveUsersFromEncryptedFile allows the caller to remove one or more users from the list of users who can decrypt the file. The caller must be able to decrypt the file in order to successfully perform this operation.

```
        AddUsersToEncryptedFile
                DWORD
                WINAPI
                AddUsersToEncryptedFile(
                        IN LPCTSTR lpFileName,
                        IN PENCRYPTION_CERTIFICATE_LIST
                        PUsers
        );
```

7. AddUsersToEncryptedFile allows the caller to add one or more users to the list of users who can decrypt the file.

```
        SetUserFileEncryptionKey
                DWORD
                SetUserFileEncryptionKey(
                IN PENCRYPTION_CERTIFICATE
                        pEncryptionCertificate
                );
```

8. SetUserFileEncryptionKey allows the user to change the certificate or private key that is used by EFS to encrypt new files or update existing files. Normally, EFS automatically handles cases

where user doesn't have a key setup or if the certificate is expired. This is done by transparently generating a key pair for the user and getting it certified. In certain cases, such as compromise of a key or if it is lost, user may want to change their key.

FreeEncryptionCertificateHashList

```
VOID
FreeEncryptionCertificateHashList(
IN PENCRYPTION_CERTIFICATE_HASH_LIST pHashes
);
```

9. FreeEncryptionCerttificateHashList allows the caller to free memory allocated during the Query APIs.

In addition to the basic APIs described above, EFS also provides four APIs for backup/restore purposes. These APIs are for the Windows 2000 Release ONLY. Applications that use them will need to handle the rewrite for subsequent releases where these APIs will be encapsulated into the planned comprehensive backup and restore APIs.

OpenRaw

```
DWORD
OpenRawW(
LPCTSTR      lpFileName,
ULONG        ulFlags,
PVOID *      pvContext
);
```

In addition to the above 9 API calls, EFS provides four transparent, or raw, file operations, called openraw, readraw, writeraw, and closeraw. These API calls provide the capability to backup and restore EFS encrypted files for back up and restore purposes. Because backup operators are not expected to possess private keys to decrypt every file, it is important that they be able to back up files in encrypted form itself.

1. OpenRaw allows backup operators to open the file in this special mode and setup a context for subsequent APIs.

OpenRaw

```
DWORD
OpenRawW(
LPCTSTR      lpFileName,
ULONG        ulFlags,
PVOID *      pvContext
);
```

2. ReadRaw allows the caller to read all the encrypted streams on the opened file (using OpenRaw) in an opaque format. The caller provides a *export function* which is used by the API to return the opaque serialized data stream to the caller. This call returns only when all data has been provided to the caller using the supplied export function.

ReadRaw

```
typedef
DWORD
(*PFE_EXPORT_FUNC)(
PBYTE pbData,
        PVOID pvCallbackContext,
        ULONG ulLength
);
        DWORD
        ReadRaw(
        PFE_EXPORT_FUNC pfExportCallback,
                PVOID       pvCallbackContext,
                PVOID       pvContext
);
```

3. WriteRaw allows the caller to write back all the opaque serialized data stream created using an earlier call to ReadRaw to recreate the original file. The caller provides an *import function* which is used by the API to obtain the opaque serialized data stream to the caller and restore the original encrypted file. This call returns only when entire file is restored or there is a failure. There may be multiple calls to the import function from within this function.

WriteRaw

```
typedef
DWORD
(*PFE_IMPORT_FUNC)(
        PBYTE pbData,
        PVOID pvCallbackContext,
        ULONG ulLength
);
        DWORD
        WriteRaw(
        PFE_IMPORT_FUNC pfImportCallback,
                PVOID       pvCallbackContext,
                PVOID       pvContext
);
```

4. CloseRaw is the cleanup API that allows EFS to cleanup the context after the file has been backed up or restored.

CloseRaw

```
        VOID
        CloseRaw(
        PVOID       pvContext
);
```