Release Notes

CEN XFS Manager Software Development Tool Kit

Revision 3.30.0.1

March 19, 2015

Introduction

This document describes the installation and configuration of the XFS Manager runtime environment, and specifies the functional software modules and references an operating system specific problem that may be relevant to developers.

The functionality of XFS is defined by the specification, which is published by CEN as a CEN workshop agreement and is not included here.

The XFS manager supplied here will support the following versions of the XFS standard:

2.00 and 3.00 and their later minor versions.

DISCLAIMER

The XFS Manager supplied with the XFS SDK 3.30 is a sample implementation of the XFS API specification. The XFS Manager has been enhanced over the years by several CEN XFS Workshop core members.

The CEN XFS Workshop recognizes that companies may identify reasons for developing their own XFS Manager and as such, the XFS Manager supplied with the XFS SDK is not a prerequisite for being XFS compliant.

CEN, the CEN XFS Workshop and the XFS Workshop members do NOT provide technical support or warranties of any kind, expressed or implied, for the XFS SDK downloaded from this site. Support is the sole responsibility of the XFS SDK user.

Software Requirements

This version of the XFS Manager requires one of the following software environments installed on the system:

- ❑ **Windows XP**
- ❑ **Windows XP Embedded**
- ❑ **Windows Vista**
- ❑ **Windows 7**

The Manager is also certified to work with the **Windows Terminal Server** and **Citrix MetaFrame** environments.

Installation

The CEN/XFS 3.30 SDK is a standard Microsoft Windows Installer package that can typically be installed on a supported Microsoft Windows platform through Windows Explorer by double-clicking the SDK330.msi.

Module Overview

The XFS Manager is implemented as three separate modules:

Module Name	Description	Installation Location
MSXFS.DLL	Basic XFS API and SPI functions as outlined in sections 4 and 5 of the XFS API/SPI Programming Reference publication.	Installation directory: <SystemFolder> e.g. C:\Windows\system32 or C:\Windows\SysWow64 for a 64 bit operating System.
XFS_SUPP.DLL	Support functions as outlined in section 6 of the XFS API/SPI Programming Reference publication.	Installation directory: <SystemFolder> e.g. C:\Windows\system32 or C:\Windows\SysWow64 for a 64 bit operating System.
XFS_CONF.DLL	Configuration functions as outlined in section 7 of the XFS API/SPI Programming Reference publication.	Installation directory: <SystemFolder> e.g. C:\Windows\system32 or C:\Windows\SysWow64 for a 64 bit operating System.

Registry Database

All configuration data for the XFS Manager and Service Providers are stored in the Windows Registry Database. The keys used are different between XFS version 2.0 and version 3.x, following Microsoft guidelines. The Manager can operate in either mode, searching first the version 3.x keys and then the version 2.0 keys, depending on the version negotiation parameters passed in on the WFSOpen command.

Configuration of the XFS Manager running in XFS 2.0 mode

This section describes how to set up the configuration data for the keys that are read and interpreted by the XFS Manager when running in XFS version 2.0 mode. The configuration data in this version is stored under key **HKEY_CLASSES_ROOT\WOSA/XFS_ROOT.**

It is possible to redirect the trace output generated by the internal trace functions of the manager to a specified file, by setting the **TraceFile** key in the Registry. If this value is not set in the configuration, trace data is written to the default file path\name C:\XFSTRACE.LOG.

By specifying **ShareFileName** and **ShareFileSize** the resources assigned to the memory management functions of the XFS Manager can be tailored to fit the needs of the service providers. If omitted **ShareFileName** defaults to C:\XFS_SUPP.SYS and **ShareFileSize** defaults to 0x500000, or 5MB.

ShareMapAddr specifies the virtual base address at which to allocate the storage**. If omitted, or set to 0, the Manager will allow the operating system to choose the address where the shared block resides within each process.**

NB. Care should be taken when allowing the Manager to choose the load address of the shared memory. This is not a safe practice if more than one process needs to access the manager at the same time. In these cases, the ShareMapAddr key and a suitable value should be used.

HKEY_CLASSES_ROOT\WOSA/XFS_ROOT\XFS_MANAGER

Value 0 Name:**TraceFile**
Type: REG_SZ
Data: <path-name>\<trace-file-name>

Value 1 Name:**ShareFileName**
Type: REG_SZ
Data: <path-name>\<share-file-name>

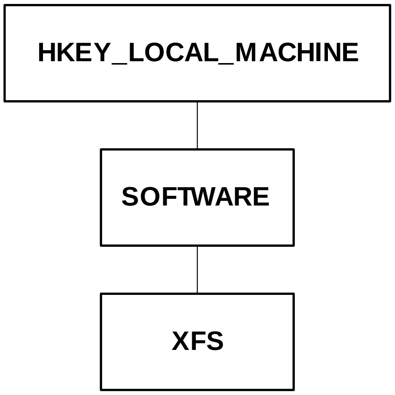
Value 2 Name:**ShareFileSize**
Type: REG_SZ
Data: <file size in bytes>

Value 3 Name:**ShareMapAddr**
Type: REG_SZ
Data: <address of shared memory>

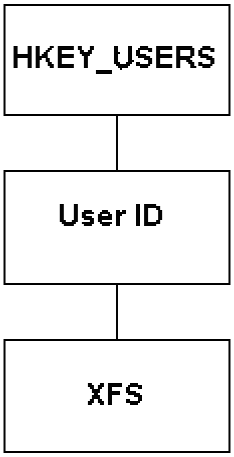
Configuration of the XFS Manager running in XFS 3.x mode

There are two logical groupings of XFS Registry information; local PC dependent configuration information and user dependent configuration information.

The local PC dependent configuration information is stored beneath the following Registry key.



User dependent configuration information is stored in the HKEY_USERS section of the Registry. NB the keys for an individual user are a mirror image of the HKEY_CURRENT_USER key when that user is logged on. Changes to either are reflected in the other.



Within the local PC dependent configuration information are stored three XFS related keys;

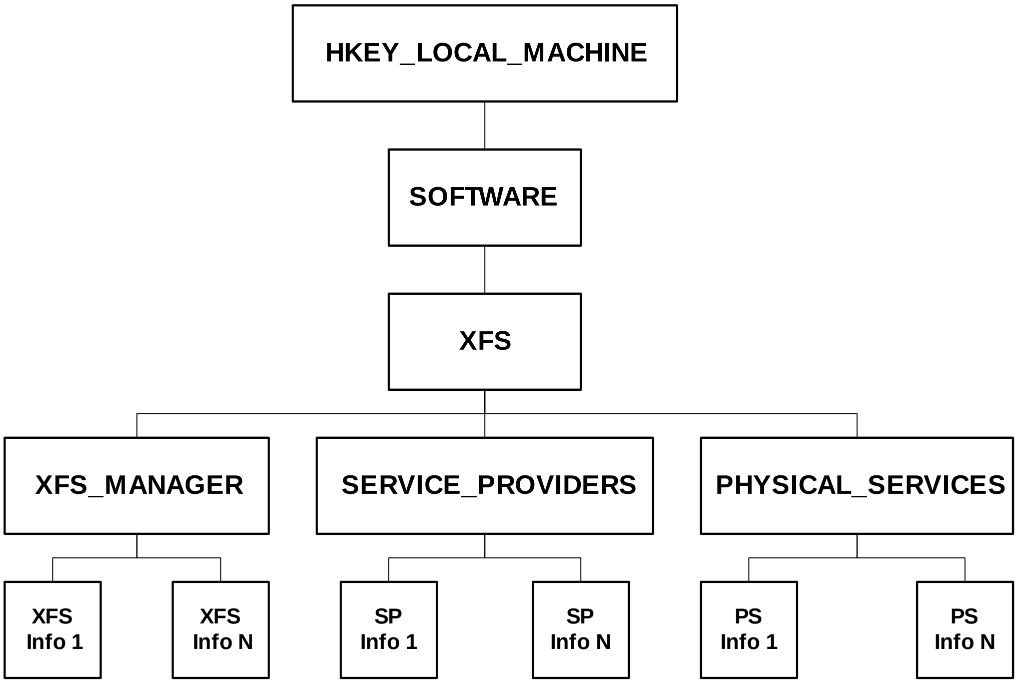
- XFS_MANAGER – Beneath this key are values and/or keys for information that the XFS Manager creates and uses.
- SERVICE_PROVIDERS – Beneath this key is a key for each XFS compliant service provider.
- PHYSICAL_SERVICES – Beneath this key are physical attachment configuration information, defined by the solution provider.

Within the User dependent configuration information is stored the following LOGICAL_SERVICES key:

- LOGICAL_SERVICES – Beneath this key is defined a key for each XFS logical service (ie: the *lpszLogicalName* parameter of the **WFSOpen**, **WFSAsyncOpen** and **WFPOpen** functions)

The configuration functions provide the capabilities to create, enumerate, open and delete keys, and to set, query and delete values within each key. Vendor-provided configuration utility programs set up the registry structure and its contents, using these functions. Configured Registry values and keys define how the XFS subsystem, services and providers are configured. These are used by the XFS Manager, applications and service providers. Note that vendor-specific information may be added to any key in this structure, using optional values.

The figure below illustrates the full structure of the local PC dependent configuration information.



The **XFS_MANAGER** key has the following optional values:

- TraceFile the name of the file containing trace data. If this value is not set in the configuration, trace data is written to the default file path\name C:\XFSTRACE.LOG.

- ShareFilename the name of the memory mapped file used by the memory management functions of the XFS Manager.

- ShareFilesize the size of the memory mapped file used by the memory management functions of the XFS Manager.

- ShareMapAddr – this key should either be omitted completely, or set to a value of 0 if you wish the Operating System to choose the load address of the shared memory block. See the description under the configuration details for version 2.0, above.

Some additional values could be also defined in the XFS SDK release notes. Please refer to the related document for more information.

A **SERVICE_PROVIDERS** key also has three mandatory values:

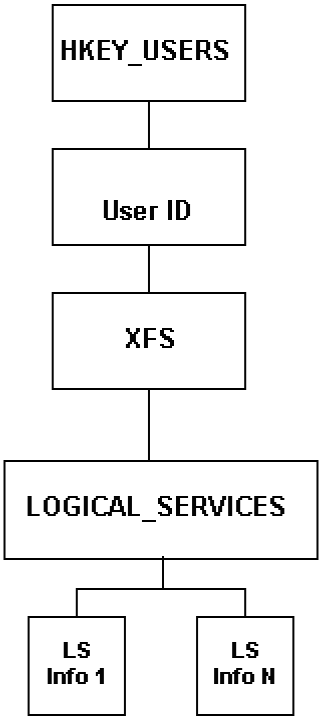
- dllname the name of the file containing the service provider DLL

- vendor_name the name of the supplier of this service provider

- version the version number of this service provider

The **PHYSICAL_SERVICES** keys are fully vendor dependent.

The figure below illustrates the full structure of the User dependent configuration information. NB the keys for an individual user are a mirror image of the HKEY_CURRENT_USER key when that user is logged on. Changes to either are reflected in the other.



Every LOGICAL_SERVICES key has two mandatory values:

- class the service class of the logical service; (see the Service Class Definition Document for the standard values)
- provider the name of the service provider that provides the logical service (the key name of the corresponding service provider key)

XFS Manager Revision Log

Revision 3.00.05	Initial 3.x release to CEN Committee
Revision 3.00.07	Enhanced tracing
Revision 3.00.08	Alter XFS_CONF handling of key query and open commands so that it tries the alternative if either MACHINE_XFS_ROOT or USER_DEFAULT_XFS_ROOT are chosen and the target cannot be found. This was because the original SDK header file had these 2 values swapped.
Revision 3.00.09	Various minor bug fixes and enhancements.
Revision 3.00.13	Various minor bug fixes mostly centered around tracing.
Revision 3.00.19	Various minor bug fixes
Revision 3.00.24	Various minor bug fixes

License Agreement

The following license terms apply to the Source Code (header) and Binary Code (.DLL and .LIB) files provided in this XFS Manager SDK. The CEN/XFS workshop members grant to you a non-exclusive royalty-free right to use and distribute the Source (header) Binary Code, provided that you:

(a) Distribute the Binary Code only in conjunction with and as a part of your software product;

(b) Agree to indemnify, hold harmless, and defend CEN XFS workshop and it's members from and against any claims or lawsuits, including attorneys' fees, that arise or result from your distribution of your software product;

(c) Agree that no liability or warranty is included with this license; and

(d) Otherwise comply with the terms of this license agreement.