

Guide to EDIdEv's Framework EDI Runtime Component Setup

The Framework EDI (FREDI) Runtime Component Setup is a single executable file that installs and configures the FREDI components on a computer so that it allows ActiveX/COM clients, programmed to interface with these components, to execute successfully. The setup can be run as a stand-alone, or it can be called from a program capable of running an external executable program.

System File updated

The setup updates the following Windows system files. These files are copied over to the Windows System directory and will only overwrite older versions.

- MFC42.DLL (6.0.8447.0)
- MSVCRT.DLL (6.0.8397.0)
- OLEAUT32.DLL (2.40.4275.1)
- OLEPRO32.DLL (5.0.4275.1)
- REGSVR32.EXE (5.0.1641.1)
- MSVCP60.DLL (6.0.8972.0)

FREDI Components Installed

The following Framework EDI files are installed in the BIN subdirectory of the target path that the user specifies. When upgrading, these files will overwrite older versions of existing files. Also, the installer takes care of the order in which these file are installed because FREDI components are dependent on one another and must be installed and registered in the order shown below.

- Edidev_SEC.dll – Standard Error Code reference component.
- Edidev_USA.dll – Language reference component.
- Edidev_Service.dll – Basic services component.
- Edidev_Comm.dll – Communication component.
- Edidev_Core.dll – Edev2000 document reader component.
- Edidev_SecureNT.dll – Edev2000 security component.
- Edidev_UNDED.dll – UN/EDIFACT reference library (Self-Register).
- Edidev_X121A4012.dll – ASC/X12 codes description (Self-Register).
- Edidev_X12DED.dll – ASC/X12 reference library (Self-Register).
- Edidev_Dictionary.edb – Standard Reference Library database.
- Edidev_Dictionary.dll – Standard Reference Library component (Self-Register).
- Edidev_SEF.dll – Standard Exchange Format component (Self-Register).
- EDIdEV.dll – Edev2000 core component (Self-Register).
- EDIdEVctl.ocx – Edev2000 Activex Control (Self-Register).
- Fredi.dll – Framework EDI core component (Self-Register).
- Edidev_SDK_RunOnce.exe – Installer Run-Once utility.

The program to uninstall Framework EDI is installed in the Window System directory.

- Edidev_SDK_Uninstall.exe – Framework EDI uninstaller program.

Registry Updates

The FREDI Runtime Setup records the files that it installs as well as all the setup settings in the registry. The Edidev key is created in the following registry branch:

```
HKEY_LOCAL_MACHINE\Software\EDIdEv
```

When the uninstaller removes files from the computer, it reads the registry information under the key:

```
HKEY_LOCAL_MACHINE\Software\EDIdEv\UninstallInfo
```

Executing Setup in a Program

The FREDI Runtime Setup can be executed from a program that can run an external program. The syntax of the call is as follows:

```
Edidev_Sdk_Runtime.exe -s<Serial Number> -p<Target Path>
```

Where **-s** indicates the serial number of the product, and **-p** indicates the target path to install the files.

Example:

```
Edidev_Sdk_Runtime.exe -s123456ABCDEFGH -pC:\Program Files\Edidev
```

Where “123456ABCDEFGH” is the serial number and “C:\Program Files\Edidev” is the target path.

Sample Visual Basic program

The following is a sample Visual Basic program to execute the setup.

```
Option Explicit
```

```
Private Type PROCESS_INFORMATION
```

```
    hProcess As Long
```

```
    hThread As Long
```

```
    dwProcessId As Long
```

```
    dwThreadId As Long
```

```
End Type
```

```
Private Type STARTUPINFO
```

```
    cb As Long
```

```
    lpReserved As String
```

```
    lpDesktop As String
```

```

lpTitle As String
dwX As Long
dwY As Long
dwXSize As Long
dwYSize As Long
dwXCountChars As Long
dwYCountChars As Long
dwFillAttribute As Long
dwFlags As Long
wShowWindow As Integer
cbReserved2 As Integer
lpReserved2 As Long
hStdInput As Long
hStdOutput As Long
hStdError As Long
End Type

Private Declare Function CreateProcess Lib "kernel32" Alias
"CreateProcessA" _
    (ByVal lpApplicationName As String, _
    ByVal lpCommandLine As String, _
    lpProcessAttributes As Any, _
    lpThreadAttributes As Any, _
    ByVal bInheritHandles As Long, _
    ByVal dwCreationFlags As Long, _
    lpEnvironment As Any, _
    ByVal lpCurrentDirectory As String, _
    lpStartupInfo As STARTUPINFO, _
    lpProcessInformation As PROCESS_INFORMATION) As Long

Private Declare Function OpenProcess Lib "kernel32.dll" _
    (ByVal dwAccess As Long, _
    ByVal fInherit As Integer, _
    ByVal hObject As Long) As Long

Private Declare Function TerminateProcess Lib "kernel32" _
    (ByVal hProcess As Long, _
    ByVal uExitCode As Long) As Long

Private Declare Function CloseHandle Lib "kernel32" _
    (ByVal hObject As Long) As Long

Private Declare Function WaitForSingleObject Lib "kernel32" (ByVal _
    hHandle As Long, ByVal dwMilliseconds As Long) As Long

Private Const SYNCHRONIZE = 1048576
Private Const NORMAL_PRIORITY_CLASS = &H20&
Private Const INFINITE = -1&

Private Sub Command1_Click()
    Dim pInfo As PROCESS_INFORMATION
    Dim sInfo As STARTUPINFO
    Dim lSuccess As Long
    Dim lRetVal As Long

    sInfo.cb = Len(sInfo)

```

```
lSuccess = CreateProcess(vbNullString, _
    " Edidev_EDI_Runtime -s123456ABCDEF -pC:\Program
Files\Edidev", _
    ByVal 0&, _
    ByVal 0&, _
    1&, _
    NORMAL_PRIORITY_CLASS, _
    ByVal 0&, _
    vbNullString, _
    sInfo, _
    pInfo)

' Wait for the shelled application to finish
lRetVal = WaitForSingleObject(pInfo.hProcess, INFINITE)

lRetVal = CloseHandle(pInfo.hThread)
lRetVal = CloseHandle(pInfo.hProcess)

MsgBox "Setup Completed"
End Sub
```