IBM White Paper

IBM Developer API Extensions for OS/2 and Source Migration Analysis Reporting Toolset

May 22, 1995

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for technical information about IBM products should be made to your IBM authorized reseller or IBM marketing representative.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program or service is not intended to state or imply that only IBM’s product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM’s intellectual property rights or other legally protectable rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user’s responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood NY 10594, U.S.A.

The following terms are trademarks of the IBM Corporation in the United States or other countries:

DB2
IBM
OS/2
Presentation Manager (PM)
VisualAge

The following terms are trademarks of other companies:

NT	Microsoft Corporation
OpenDoc	Apple Computer, Inc.
SMART	One Up Corporation
Solaris	Sun Microsystems, Inc.
Win32	Microsoft Corporation
Windows	Microsoft Corporation
Windows NT	Microsoft Corporation

Contents

Introduction 1
IBM Developer API Extensions Overview 1
Developer API Extensions Details 2
Architecture 2
Common Code Support 3
Application Support 4
Scope 5
Source Migration Analysis Reporting Toolset (SMART) Overview 5
SMART Details 6
Analysis 6
Migration 7
Resources Translation 8
WinHelp Translation 9
Getting Started with Developer API Extensions and SMART 9

Appendix A. Developer API Extensions List 10

Introduction

IBM has a way for application developers to reduce code development costs while reaching all the major 32-bit platforms. By expanding OS/2 Warp with the IBM Developer API Extensions, IBM enables developers to create common code for OS/2 Warp, Windows* NT, and Windows 95 operating systems. In addition, these extensions to OS/2 Warp along with the Source Migration Analysis Reporting Tool Set (SMART)** will simplify migration from existing Windows applications to the OS/2 Warp environment. This paper provides technical information on the Developer API Extensions and SMART.

IBM Developer API Extensions Overview

IBM Developer API Extensions expands the OS/2 Warp APIs and messages. Developer API Extensions let application developers maintain a common code base for 32-bit Windows and OS/2 Warp applications. Developer API Extensions provide over 700 new APIs that are consistent with Windows APIs. The subset was chosen based on analysis of more than nine million lines of code from a variety of applications. Application developers use Developer API Extensions to write portions of their code to be common between OS/2 Warp and Windows, while still exploiting OS/2 Warp's existing API set and advanced features.

Developer API Extensions enhance the OS/2 Warp operating system and tools. Windows applications that conform to Developer API Extensions can be recompiled to produce a functionally equivalent OS/2 Warp application. The application then runs on OS/2 Warp and has the OS/2 Warp look and feel.

Developer API Extensions simplify the migration for procedurally-based applications. Developer API Extensions is targeted at core operating system functions. Operating system extensions such as multimedia, pen, and OLE are not within the scope of Developer API Extensions.

In addition to extensions to OS/2 Warp, IBM is working with One Up Corporation to enhance SMART with user-defined migration dictionaries (UDMD) specifically designed to exploit Developer API Extensions. SMART analyzes and migrates program code, resource files, and help files (see “Source Migration Analysis Reporting Toolset (SMART) Overview” on page 5).

Lotus Development has piloted the use of Developer API Extensions for portions of its OS/2 development and has achieved impressive results to date. Through the use of a common code base, Lotus has committed to providing OS/2 Warp applications that are functionally equivalent to their Windows 95 counterparts. These applications are targeted for delivery commencing in ’95. Certain areas within these applications, when coupled with the maturity and robustness of OS/2 Warp, may very well outperform their Windows 95 counterparts.

The combination of Developer API Extensions and SMART enhancements enable migration to OS/2 Warp applications with significantly less effort. For new development projects, Developer API Extensions reduces the development, test, and maintenance costs for applications that are targeted at multiple platforms (OS/2 Warp, Windows NT, and Windows 95). IBM believes that most application developers will realize an 80% common code solution for Windows and OS/2 Warp, with the remaining 20% customized to exploit the unique features of each operating environment. See Figure 1 on page 2. By including Developer API Extensions in OS/2 Warp, IBM makes it easier for Windows application developers to capture the market opportunities of the ever-increasing OS/2 Warp install base.

* Windows is a registered trademark of the Microsoft Corporation.

** SMART is a registered trademark of the One-Up Corporation.

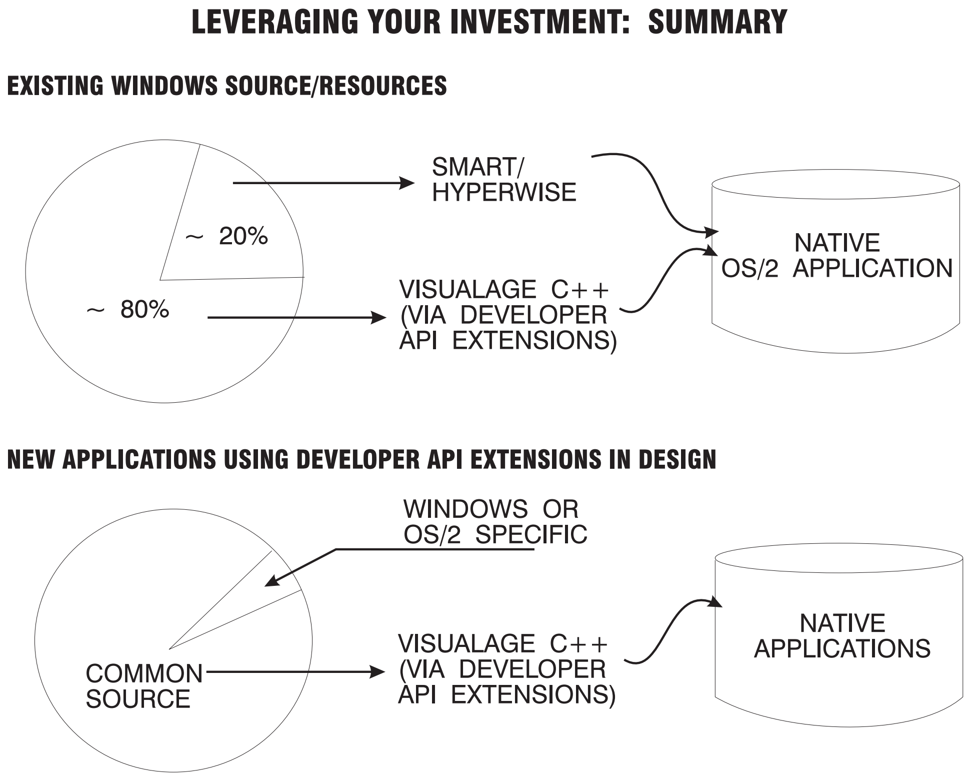


Figure 1. Leveraging your Investment

A beta level of Developer API Extensions will be available this summer. The final version will be available by the end of 1995.

Developer API Extensions Details

Developer API Extensions provides a simplified migration path to OS/2 Warp and facilitates writing to multiple platforms. Applications developed with Developer API Extensions:

- Can include both Developer API Extensions APIs and existing OS/2 Warp APIs
- Are offered APIs that are functionally equivalent to Windows NT 3.5 and Windows 95 APIs
- Can exploit OS/2 specific features and functions

Architecture: Figure 2 on page 3 shows the Developer API Extensions architecture that allows applications to:

- Make calls to either Developer API Extensions or to other OS/2 Warp components
- Maintain Windows code and enhance the functions with OS/2 Warp APIs as needed

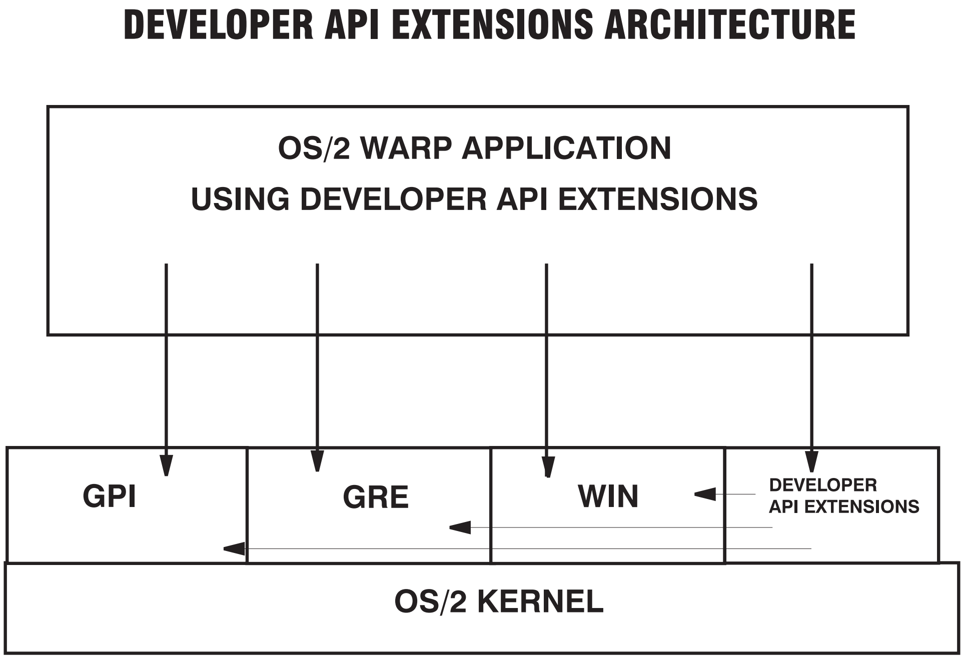

DEVELOPER API EXTENSIONS ARCHITECTURE
OS/2 WARP APPLICATION
USING DEVELOPER API EXTENSIONS
GPI
GRE
WIN
DEVELOPER
API EXTENSIONS
OS/2 KERNEL

Figure 2. Developer API Extensions Architecture

Common Code Support: Developer API Extensions provides a:

- Migration path for existing Windows applications using SMART
- Common code for Windows and OS/2 Warp applications

Windows developers will be able to use SMART to analyze their code and identify the Developer API Extensions APIs that are supported in OS/2 Warp. The APIs that are not supported can be converted to other OS/2 Warp APIs. The RTF and RC files can be converted to OS/2 Warp format by using SMART. Developer API Extensions is essential to maintaining a common code base for multiple platforms. Because Developer API Extensions covers the most commonly used Windows 32-bit APIs, applications should be designed so that platform dependent functions are separate from platform independent functions.

Due to the breadth of Developer API Extensions, the platform-dependent code is small and unique. Platform independent functions that use Developer API Extensions can be maintained as common source code. By using common source code, enhancing and maintaining applications becomes easier. See Figure 3.

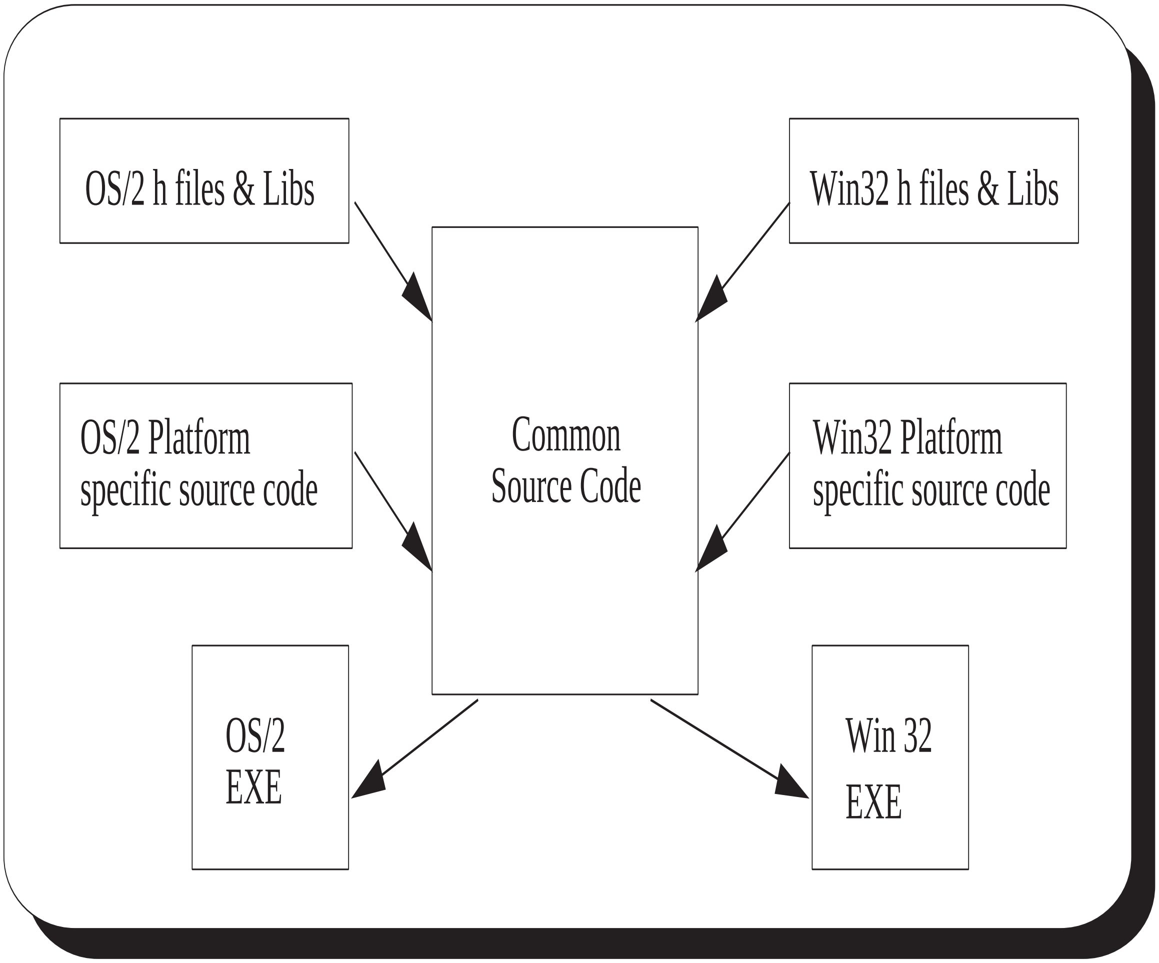


Figure 3. Developer API Extensions Common Code Support

Application Support: Developer API Extensions provides application support during development and run time. To create a Developer API Extensions application, the following are required:

- Developer API Extensions headers and libraries
- OS/2 Warp C compiler (for example, IBM C Set++ Compiler or VisualAge C++)
- OS/2 Warp Toolkit

These tools and utilities are available through the OS/2 Warp Toolkit or The Developer's Connection for OS/2. The compilers are available separately. By using these tools and utilities, application developers can migrate Windows programs to OS/2 Warp or develop Developer API Extensions applications for multiple platforms.

Developer API Extensions will be shipped as part of the OS/2 Warp operating system. Once applications are developed, the only requirement to run the application is an OS/2 Warp release that contains the Developer API Extensions enhancements.

Scope: Developer API Extensions supports the following Windows areas:

- Graphics device interface
- Windows management (Including messages, MDI, DDEML)
- System services including:
  - Date and time
  - File I/O
  - Memory management
  - Module management
  - Printing
  - Processes and threads
  - Registry
  - Resource management
  - Synchronization
  - Environment

Developer API Extensions does not currently support the following Windows areas:

- Communication
- Exception handling
- File compression extensions
- Multimedia extensions
- OLE
- Pen windows extensions
- Print spooling extensions
- Security

However, most of these extensions are provided by functionally equivalent OS/2 Warp APIs.

Source Migration Analysis Reporting Toolset (SMART) Overview

SMART makes migrating code to OS/2 Warp easier than ever. This innovative tool set, developed by One Up Corporation, provides tools to migrate Windows code to OS/2 Warp code.

SMART assists in the migration of source code from:

- 16-bit OS/2 to 32-bit OS/2 code
- 16-bit Windows to 32-bit OS/2 code
- 32-bit Windows to 32-bit OS/2 code

SMART provides assistance through the following tools:

- **Analysis and Reporting Tool:** Analyzes the conversion effort and provides reports that are a migration road map.
- **Source Migration:** Allows migration of API and message code and provides progress reports and online references to assist in migration.
- **Resource Translator:** Takes Windows resources, including cursors and icons, and converts them to OS/2 Warp resources.
- **Viewer:** Provides context sensitive online help for the SMART code migration process.
- **SLink, the SMART editor:** Integrated with the complete SMART, SLink (SourceLink), the SMART editor provides both editing and hyperlinking to the key elements in code.

- **User-Defined Migration Dictionaries (UDMD) Library:** SMART has a library of UDMDs that ease the analysis and conversion of Windows help files, resources, and source code. UDMDs are available to target the following platforms:

  – OS/2 Warp
  – OS/2 Warp for the PowerPC

  Developers can also create UDMDs to program or tailor the migration process for a particular environment. The migration command language (MiCL) is another programmable feature that provides the capability to prompt the user for data or processing options during the migration process.

  SMART is available on The Developer Connection for OS/2. Extensive information is available about each SMART component in the online documentation found in the SMART folder once SMART is installed. It is IBM's intent to provide the Developer API Extensions UDMD on CompuServe and on the IBM Solution Developer Operations (SDO) home page on the Internet in the near future. See the SDO home page at:

  `http://www.austin.ibm.com/developer`

SMART Details

SMART automates many of the changes required to migrate code. Additionally, SMART identifies and provides application developers with a detailed description of several alternative implementations for the remainder of the issues.

SMART provides three main functions to assist in the migration process:

1. Analysis
2. Code migration
3. Resource translation

Analysis: The Analysis and Reporting Tool analyzes source code and resources and provides these reports:

1. Analysis Summary Report
2. File Detail Report
3. Keyword Detail Report

SMART identifies items or keywords in the code that will require migration to OS/2 and groups them into categories. The categories are based on the level of difficulty required to migrate the items to OS/2. The "Total Category Hits" lines in the Analysis Summary Report are the number of migration items by category. The categories are:

000 Informational only
An exact match of the keyword exists in OS/2 Warp. There may be a change in the size or type (for example, a short to a long). The Developer API Extensions will fall in this category when using the Developer API Extensions UDMD.

010 Literal replacement
An equivalent definition exists in OS/2 Warp. A literal change in the keyword is required (for example, LPSTR to PSZ).

020 Replacement with parameter changes
The equivalent function exists in OS/2 Warp, but parameters or fields of a structure differ slightly from the source platform definition. This category also includes some of the parameters that are not applicable or required in OS/2 Warp.

030 Change with more/less API calls

The equivalent function exists in OS/2 Warp but the function must be implemented with more, or sometimes less, function calls. This category also includes items that map to one of several choices depending upon the type of parameter used.

040 Logic changes required

Similar function exists in OS/2 Warp but the logic required to do the function must be reworked.

050 Functionality does not exist

The function is not available on OS/2 Warp.

999 Dictionary entry not defined

The SMART database does not currently have a complete description for the migration of the keyword.

Migration: Through the extensive keyword dictionaries supplied with SMART a majority of source code is automatically migrated. The migration command language (MiCL) and user defined migration dictionaries (UDMD) allow the developer to further automate the migration process.

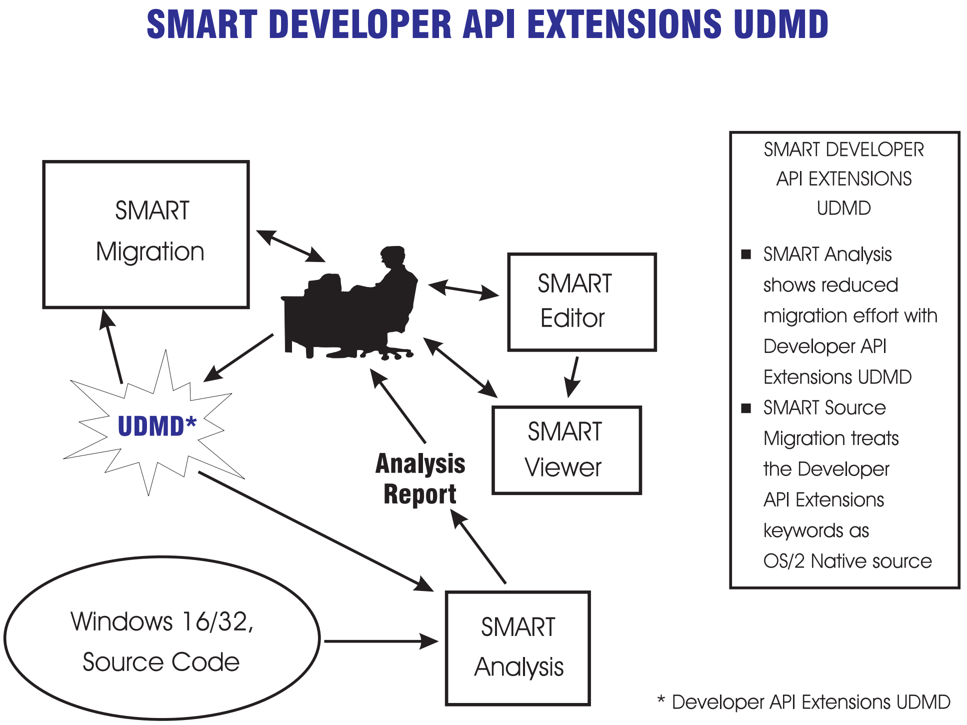
The MiCL allows the developer to program the migration process for each keyword affected. By using MiCL, SMART provides the capability to prompt for data or processing options when an occurrence of a migration keyword is detected.

The use of a UDMD provides the flexibility to extend and tailor the analysis and conversion function of the SMART migration tables. The developer can override equivalent migration items or keyword entries in SMART tables. A UDMD may also be used to expand the list of migration items or keywords SMART will convert automatically. This allows the definition of keyword mapping from the Windows platform to OS/2 Warp once; then, the UDMD will map the migration for multiple instances of the keyword in the code. The "Total Hits Processed with UDMD" in the Analysis Summary Report is the number of keywords processed for which the mapping or migration is specified in the UDMD. By using the Developer API Extensions UDMD with the Windows SMART table, a migration analysis can be generated. The Developer API Extensions UDMD dramatically reduce the amount of effort needed to migrate a Windows application to OS/2.

Part of the analysis generated shows the savings based on the various categories and the **Net Migration Effort** is the net approximate effort it may take to migrate a program to OS/2 Warp. Often, this number is the approximate number of hours. However, the migration effort is based on the programmer's experience in both Windows and OS/2 Warp. The migration and analysis process is iterative.

Because the Developer API Extensions broaden the API set available in OS/2 Warp, SMART will now recognize these extensions as native OS/2 API. The analysis done by SMART identify the Windows APIs that do not conform to the Developer API Extensions.

Figure 4 on page 8 illustrates how SMART and the Developer API Extensions work together. Existing source code is used as input to the SMART Analysis Tool, along with the Developer API Extensions UDMD. The SMART Analysis Tool generates a report which categorizes all the keywords found into different groups, depending on the effort necessary to migrate the keyword to a native OS/2 keyword. The report also estimates the effort necessary to complete the migration to OS/2.



SMART DEVELOPER API EXTENSIONS UDMD

- SMART Analysis shows reduced migration effort with Developer API Extensions UDMD
- SMART Source Migration treats the Developer API Extensions keywords as OS/2 Native source

* Developer API Extensions UDMD

Figure 4. SMART UDMDs

The SMART Migration Tool can be used with the appropriate UDMD to generate OS/2 source code from existing Windows source code for those APIs that require conversion. The SMART Editor and Viewer are provided to assist developers in migrating those keywords which are beyond the scope of the Developer API Extensions.

The Keyword Detail Report shows that there are a few items that are not automatically migrated. To analyze these items, use the SMART Viewer, which can be invoked directly from SLink, the SMART editor, through a REXX macro provided with the SMART package.

The SMART Viewer provides context sensitive online help to SMART Migration. Use the SMART Viewer to search on a topic name and information on that topic is displayed on a SMART Viewer window. It also provides "Template," "Prototype," "Example," and "References" for a migration keyword.

Resources Translation: SMART also converts Windows resources, cursors, and icons to OS/2 resources, cursors, and icons.

The SMART Windows to OS/2 Warp Resource Conversion utility provides source-to-source conversion of Windows resource definition files to OS/2 Warp resource definition files. Among the functions included are:

- Automatic reformatting of resource definition statements
- Generation of alternate accelerator key sequences
- Conversion of mnemonic characters
- Code page conversion of text strings
- Remapping of dialog control styles

The SMART Windows to OS/2 Warp cursor and icon conversion utility provides binary conversion of Windows' icon and cursor files to OS/2 Warp pointer files. This conversion utility may be run from:

- The SMART action bar menu
- The command line
- A batch command file

WinHelp Translation: The SMART Windows to OS/2 Warp Help Translator conversion utility converts Windows help files to OS/2 Warp help files. Use the SMARTHLP.EXE to read an HPJ file and create a Presentation Manager (PM*) IPF-compatible HLP file. SMARTHLP.EXE takes the rich text format (RTF) and bit map (BMP) components of the Windows help and converts these to an OS/2 Warp IPF file. Then, edit the IPF file or pass it directly to the IPF compiler to create a viewable INF file.

Getting Started with Developer API Extensions and SMART

To use Developer API Extensions now, take a careful look at the Developer API Extensions API list (see Appendix A, “Developer API Extensions List” on page 10) and compare this to the APIs being used in your current 32-bit Windows development. This analysis will show how close your product is to being an OS/2 Warp product. SMART can help determine which APIs are currently being used in your application code.

When Developer API Extensions becomes available, all the prerequisites will be complete for you to be able to use the Developer API Extensions to create OS/2 Warp applications. Announcement of specific availability dates will be made on CompuServe and Prodigy.

* Presentation Manager (PM) is a registered trademark of the IBM Corporation.

Appendix A. Developer API Extensions List

The following is the current target list of functions supported by IBM Developer API Extensions Warp:

AbortDoc
AbortPath
AddAtom
AddFontResource
AdjustWindowRect
AdjustWindowRectEx
AngleArc
AnimatePalette
AppendMenu
Arc
ArcTo
ArrangeIconicWindows

Beep
BeginDeferWindowPos
BeginPaint
BeginPath
BitBlt
BringWindowToTop

CallMsgFilter
CallNextHookEx
CallWindowProc
ChangeClipboardChain
CharLower
CharLowerBuff
CharNext
CharPrev
CharToOem
CharToOemBuff
CharUpper
CharUpperBuff
CheckDlgButton
CheckMenuItem
CheckRadioButton
ChildWindowFromPoint
ChooseColor
ChooseFont
Chord
ClientToScreen
ClipCursor
CloseClipboard
CloseEnhMetaFile
CloseFigure
CloseHandle
CloseMetaFile
CloseWindow
CombineRgn
CommDlgExtendedError
CompareFileTime
CopyCursor
CopyEnhMetaFile
CopyFile
CopyIcon
CopyMetaFile
CopyRect
CountClipboardFormats
CreateAcceleratorTable
CreateBitmap
CreateBitmapIndirect
CreateBrushIndirect
CreateCaret
CreateCompatibleBitmap
CreateCompatibleDC
CreateCursor
CreateDC
CreateDialog
CreateDialogIndirect
CreateDialogIndirectParam
CreateDialogParam
CreateDIBitmap
CreateDIBPatternBrushPt
CreateDirectory
CreateEllipticRgn
CreateEllipticRgnIndirect
CreateEnhMetaFile
CreateEvent
CreateFile
CreateFont
CreateFontIndirect
CreateHatchBrush
CreateIC
CreateIcon
CreateIconFromResource
CreateIconIndirect
CreateMDIWindow
CreateMenu
CreateMenu
CreateMetaFile
CreateMutex
CreatePalette
CreatePatternBrushPt
CreatePen
CreatePenIndirect
CreatePolygonRgn
CreatePolyPolygonRgn
CreatePopupMenu
CreateProcess
CreateRectRgn
CreateRectRgnIndirect
CreateRoundRectRgn
CreateScalableFontResource
CreateSemaphore
CreateSolidBrush
CreateThread
CreateWindow
CreateWindowEx

DdeAbandonTransaction
DdeAccessData
DdeAddData
DdeClientTransaction
DdeCmpStringHandles
DdeConnect
DdeConnectList
DdeCreateDataHandle
DdeCreateStringHandle
DdeDisconnect
DdeDisconnectList
DdeEnableCallback
DdeFreeDataHandle
DdeFreeStringHandle
DdeGetData
DdeGetLastError
DdeInitialize
DdeKeepStringHandle
DdeNameService
DdePostAdvise
DdeQueryConvInfo
DdeQueryNextServer
DdeQueryString
DdeReconnect
DdeSetUserHandle
DdeUnaccessData
DdeUninitialize
DefDlgProc
DeferWindowPos
DefFrameProc
DefHookProc
DefMDIChildProc
DefWindowProc

DeleteAtom
DeleteCriticalSection
DeleteDC
DeleteEnhMetaFile
DeleteFile
DeleteMenu
DeleteMetaFile
DeleteObject
DestroyAcceleratorTable
DestroyCaret
DestroyCursor
DestroyIcon
DestroyMenu
DestroyWindow
DialogBox
DialogBoxIndirect
DialogBoxIndirectParam
DialogBoxParam
DispatchMessage
DlgDirList
DlgDirListComboBox
DlgDirSelectComboBoxEx
DlgDirSelectEx
DosDateTimeToFileTime
DPtoLP
DragAcceptFiles
DragFinish
DragQueryFile
DragQueryPoint
DrawFocusRect
DrawIcon
DrawMenuBar
DrawText

Ellipse
EmptyClipboard
EnableMenuItem
EnableScrollBar
EnableWindow
EndDeferWindowPos
EndDialog
EndDoc
EndPage
EndPaint
EndPath
EnterCriticalSection
EnumChildWindows
EnumClipboardFormats
EnumEnhMetaFile
EnumFontFamilies
EnumFonts
EnumMetaFile
EnumObjects
EnumProps
EnumPropsEx
EnumThreadWindows
EnumWindows
EqualRect
EqualRgn
Escape
ExcludeClipRect
ExcludeUpdateRgn
ExitProcess
ExitThread
ExitWindows
ExitWindowsEx
ExtCreateRegion
ExtFloodFill
ExtSelectClipRgn
ExtTextOut

FatalAppExit
FatalExit
FileTimeToDosDateTime
FileTimeToLocalFileTime
FileTimeToSystemTime
FillRect
FillRgn
FindAtom
FindClose
FindFirstFile
FindNextFile
FindResource
FindText
FindWindow
FlashWindow
FlattenPath
FloodFill
FlushFileBuffers
FrameRect
FrameRgn
FreeLibrary
FreeProcInstance

GetACP
GetActiveWindow
GetArcDirection
GetAspectRatioFilterEx
GetAtomName
GetBitmapBits
GetBitmapDimensionEx
GetBkColor
GetBkMode
GetBoundsRect
GetBrushOrgEx
GetBValue
GetCapture
GetCaretBlinkTime
GetCaretPos
GetCharABCWidths
GetCharWidth
GetClassInfo
GetClassLong
GetClassName
GetClassWord
GetClientRect
GetClipboardData
GetClipboardFormatName
GetClipboardViewer
GetClipBox
GetClipCursor
GetClipRgn
GetCommandLine
GetCurrentDirectory
GetCurrentPositionEx
GetCurrentProcess
GetCurrentProcessId
GetCurrentThread
GetCurrentThreadId
GetCurrentTime
GetCursor
GetCursorPos
GetDC
GetDCEx
GetDCOrgEx
GetDesktopWindow
GetDeviceCaps
GetDialogBaseUnits
GetDIBits
GetDiskFreeSpace
GetDlgCtrlID
GetDlgItem
GetDlgItemInt
GetDlgItemText
GetDoubleClickTime
GetDriveType
GetEnhMetaFile
GetEnhMetaFileBits
GetEnhMetaFileHeader
GetEnhMetaFilePaletteEntries
GetEnvironmentStrings
GetEnvironmentVariable
GetFileAttributes
GetFileInformationByHandle
GetFileSize
GetFileTime
GetFileTitle
GetFileType
GetFocus
GetFontData
GetForegroundWindow
GetFullPathName

GetGlyphOutline
GetGValue
GetIconInfo
GetKerningPairs
GetKeyboardType
GetKeyState
GetLastActivePopup
GetLastError
GetLocalTime
GetLogicalDrives
GetLogicalDriveStrings
GetMapMode
GetMenu
GetMenuCheckMarkDimensions
GetMenuItemCount
GetMenuItemID
GetMenuState
GetMenuString
GetMessage
GetMessageExtraInfo
GetMessagePos
GetMessageTime
GetMetaFile
GetMetaFileBitsEx
GetMiterLimit
GetModuleFileName
GetModuleHandle
GetNearestColor
GetNearestPaletteIndex
GetNextDlgGroupItem
GetNextDlgTabItem
GetNextWindow
GetObject
GetObjectType
GetOEMCP
GetOpenClipboardWindow
GetOpenFileName
GetOutlineTextMetrics
GetPaletteEntries
GetParent
GetPath
GetPixel
GetPolyFillMode
GetPriorityClass
GetPriorityClipboardFormat
GetPrivateProfileInt
GetPrivateProfileString
GetProcAddress
GetProfileInt
GetProfileString
GetProp
GetQueueStatus
GetRasterizerCaps
GetRegionData

GetRgnBox
GetROP2
GetRValue
GetSaveFileName
GetScrollPos
GetScrollRange
GetStockObject
GetStretchBltMode
GetSubMenu
GetSysColor
GetSystemDirectory
GetSystemMenu
GetSystemMetrics
GetSystemPaletteEntries
GetSystemTime
GetTabbedTextExtent
GetTempFileName
GetTempPath
GetTextAlign
GetTextCharacterExtra
GetTextColor
GetTextExtentPoint
GetTextFace
GetTextMetrics
GetThreadPriority
GetTickCount
GetTimeZoneAdjustment
GetTopWindow
GetUpdateRect
GetUpdateRgn
GetViewportExtEx
GetViewportOrgEx
GetVolumeInformation
GetWindow
GetWindowDC
GetWindowExtEx
GetWindowLong
GetWindowOrgEx
GetWindowPlacement
GetWindowRect
GetWindowsDirectory
GetWindowText
GetWindowTextLength
GetWindowThreadProcessId
GetWindowWord
GetWinMetaFileBits
GetWorldTransform
GlobalAddAtom
GlobalAlloc
GlobalDeleteAtom
GlobalDiscard
GlobalFindAtom
GlobalFlags
GlobalFree

GlobalGetAtomName
GlobalHandle
GlobalLock
GlobalMemoryStatus
GlobalReAlloc
GlobalSize
GlobalUnlock

HeapAlloc
HeapCreate
HeapDestroy
HeapFree
HeapReAlloc
HeapSize
HideCaret
HiliteMenuItem

InflateRect
InitAtomTable
InitializeCriticalSection
InSendMessage
InsertMenu
InterlockedDecrement
InterlockedExchange
InterlockedIncrement
IntersectClipRect
IntersectRect
InvalidateRect
InvalidateRgn
InvertRect
InvertRgn
IsBadCodePtr
IsBadHugeReadPtr
IsBadHugeWritePtr
IsBadReadPtr
IsBadStringPtr
IsBadWritePtr
IsCharAlpha
IsCharAlphaNumeric
IsCharLower
IsCharUpper
IsChild
IsClipboardFormatAvailable
IsDBCSLeadByte
IsDialogMessage
IsDlgButtonChecked
IsIconic
IsMenu
IsRectEmpty
IsWindow
IsWindowEnabled
IsWindowVisible
IsZoomed

KillTimer

LeaveCriticalSection
LineDDA
LineTo
LoadAccelerators
LoadBitmap
LoadCursor
LoadIcon
LoadLibrary
LoadMenu
LoadMenuIndirect
LoadModule
LoadResource
LoadString
LocalAlloc
LocalDiscard
LocalFileTimeToFileTime
LocalFlags
LocalFree
LocalHandle
LocalLock
LocalReAlloc
LocalSize
LocalUnlock
LockFile
LockResource
LockWindowUpdate
LPtoDP
lstrcat
lstrcmp
lstrcmpi
lstrcpy
lstrlen

MakeProcInstance
MapDialogRect
MapVirtualKey
MapWindowPoints
MessageBeep
MessageBox
ModifyMenu
ModifyWorldTransform
MoveFile
MoveToEx
MoveWindow
MsgWaitForMultipleObjects
MulDiv

OemToChar
OemToCharBuff
OffsetClipRgn
OffsetRect
OffsetRgn
OffsetViewportOrgEx
OffsetWindowOrgEx
OpenClipboard
OpenEvent
OpenFile
OpenMutex
OpenProcess
OpenSemaphore
OutputDebugString

PackDDElParam
PaintRgn
PatBlt
PathToRegion
PeekMessage
Pie
PlayEnhMetaFile
PlayMetaFile
PlayMetaFileRecord
PolyBezier
PolyBezierTo
PolyDraw
Polygon
Polyline
PolylineTo
PolyPolygon
PolyPolyline
PostMessage
PostQuitMessage
PostThreadMessage
PrintDlg
PtInRect
PtInRegion
PtVisible

ReadFile
RealizePalette
Rectangle
RectInRegion
RectVisible
RedrawWindow
RegCloseKey
RegCreateKey
RegCreateKeyEx
RegDeleteKey
RegDeleteValue
RegEnumKey
RegEnumKeyEx
RegEnumValue
RegisterClass
RegisterClipboardFormat
RegisterWindowMessage
RegOpenKey
RegOpenKeyEx
RegQueryInfoKey
RegQueryValue
RegQueryValueEx
RegSetValue
RegSetValueEx
ReleaseCapture
ReleaseDC
ReleaseMutex
ReleaseSemaphore
RemoveDirectory
RemoveFontResource
RemoveMenu
RemoveProp
ReplaceText
ReplyMessage
ResetDC
ResetEvent
ResizePalette
RestoreDC
ResumeThread
RoundRect

SaveDC
ScaleViewportExtEx
ScaleWindowExtEx
ScreenToClient
ScrollDC
ScrollWindow
ScrollWindowEx
SearchPath
SelectClipRgn
SelectObject
SelectPalette
SendDlgItemMessage
SendMessage
SetActiveWindow
SetArcDirection
SetBitmapBits
SetBitmapDimensionEx
SetBkColor
SetBkMode
SetBoundsRect
SetBrushOrgEx
SetCapture
SetCaretBlinkTime
SetCaretPos
SetClassLong
SetClassWord
SetClipboardData
SetClipboardViewer
SetCurrentDirectory
SetCursor
SetCursorPos
SetDIBits

SetDIBitsToDevice
SetDlgItemInt
SetDlgItemText
SetDoubleClickTime
SetEndOfFile
SetEnhMetaFileBits
SetEnvironmentVariable
SetEvent
SetFileAttributes
SetFilePointer
SetFileTime
SetFocus
SetForegroundWindow
SetHandleCount
SetLastError
SetLocalTime
SetMapMode
SetMapperFlags
SetMenu
SetMenuItemBitmaps
SetMetaFileBitsEx
SetMiterLimit
SetPaletteEntries
SetParent
SetPixel
SetPolyFillMode
SetPriorityClass
SetProp
SetRect
SetRectEmpty
SetRectRgn
SetROP2
SetScrollPos
SetScrollRange
SetStretchBltMode
SetSysColors
SetSystemTime
SetTextAlign
SetTextCharacterExtra
SetTextColor
SetTextJustification
SetThreadPriority
SetTimer
SetTimeZoneInformation
SetViewportExtEx
SetViewportOrgEx
SetVolumeLabel
SetWindowExtEx
SetWindowLong
SetWindowOrgEx
SetWindowPlacement
SetWindowPos
SetWindowsHookEx
SetWindowText
SetWindowWord
SetWinMetaFileBits
SetWorldTransform
ShowCaret
ShowCursor
ShowOwnedPopups
ShowScrollBar
ShowWindow
SizeofResource
Sleep
StartDoc
StartPage
StretchBlt
StretchDIBits
StrokeAndFillPath
StrokePath
SubtractRect
SuspendThread
SwapMouseButton
SystemParametersInfo
SystemTimeToFileTime
SystemTimeToTzSpecificLocalTime

TabbedTextOut
TerminateProcess
TerminateThread
TextOut
timeGetSystemTime
timeGetTime
TlsAlloc
TlsFree
TlsGetValue
TlsSetValue
ToAscii
TrackPopupMenu
TranslateAccelerator
TranslateMDISysAccel
TranslateMessage

UnhookWindowsHookEx
UnionRect
UnlockFile
UnpackDDElParam
UnrealizeObject
UnregisterClass
UpdateWindow

ValidateRect
ValidateRgn
VkKeyScan

WaitForMultipleObjects
WaitForSingleObject
WaitMessage
WidenPath
WindowFromDC
WindowFromPoint
WinExec
WinHelp
WriteFile
WritePrivateProfileString
WriteProfileString
wsprintf
wvsprintf

Yield

ZeroMemory