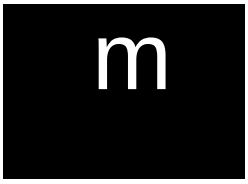
IBM United Kingdom Limited

OpenEdition MVS and VM: Overview and Porting

31st August, 1995

Mark Cathcart
IT Consultant



IBM United Kingdom Limited
Rosanne House, Bridge Road
Welwyn Garden City, Herts. AL8 6TZ
UK

About the author

Mark Cathcart is Technical Consultant to Enterprise Systems at IBM United Kingdom on Client/Server and Open Systems. Previously, he was the senior professional in VM/VSE Product Management and Technical Support. He is a "certified" IT Consultant, a Knight of the Order of VM and a member of the British Computer Society Client/Server Group.

Mark joined IBM Systems Engineering in 1987 and became a 'Poacher turned Game-keeper' after spending 13 years as an IBM customer. This included four years with responsibility for the VM systems at a leading New York bank. Mark likes to play "devils advocate" at IBM and has a unique insider's perspective.

Involved with enterprise workstations, distributed applications, and data distribution since 1983, Mark worked with the project team for America's first successful Home Banking System. More recently, Mark wrote the IBM System/390 Open Client/Server guide as well as the co-author of a number of commercially available computer books and articles.

Mark is the IBM liaison to the Interoperability and Portability domain for GUIDE SHARE Europe.

Mark can be contacted via:
the 'Net:
http://www.s390.ibm.com/sections/corner/index.htm
IBM VNET: CATHCAM at NHBVM8
InterNet: Mark_Cathcart@uk.ibm.com
Inter-Enterprise Address(IBMMail): GBIBMWJP
X400:C:GB A:IBMX400 P:IBMMAIL S:CATHCART
G:CATHCAM

Trademarks:

The following are trademarks of the companies shown:

DEC - Digital Equipment Corporation
HP-UX - Hewlett-Packard Company
IEEE, POSIX - Institute of Electrical and Electronic Engineers(IEEE)
OSF, Open Software Foundation, DCE, DFS - Open Software Foundation Inc.
Solaris , SUN, NFS - Sun Microsystems Inc.
UNIX is licensed exclusively through X/Open Company Limited
X/Open, XPG - X/Open Company Limited
X-Window - Massachusetts Institute of Technology

The following are trademarks or registered trademarks of the IBM Corporation:

- AIX, CICS, DB2, DFSMS, ES/9000, IBM, IMS/ESA, MVS/ESA, OpenEdition, OS/2, RACF, S/390, System/390, VM/ESA, VSE/ESA

About this document

This document is based on three earlier documents by the author, *Opening Up VM; Looks like UNIX, Feels like UNIX, Works like MVS;* the *OpenEdition MVS Update and Progress Report; and One Man's View of OpenEdition.*

Disclaimer

This document expresses the views of its author only.

The information contained in this document has not been submitted to any formal IBM test and is distributed on an *"as is"* basis **without any warranty either expressed or implied.** The use of this information or the implementation of any of the techniques is the reader's responsibility and depends on the reader's ability to evaluate and integrate them into their own operational environment.

Open Distributed Challenges

Most business managers today want to preserve technology investments and to address new business opportunities. What is needed are solutions for total system integration that make the resources of the IBM mainframe and LAN systems available to anyone in the customer's network when they need them.

The IBM strategy is based on the Open Blueprint and is intended to meet these objectives. Simply put, the strategy describes a cohesive environment where applications in the network can exploit the traditional strengths of MVS and VM using an integrated set of open system services

By facilitating portability and interoperability, open systems enhance the ability to protect investments. But let's take a look a those two words, Interoperability and Portability.

Portability

For many, portability is just about applications, being able to run them without any changes. This is important; but portability is also about data and about people and their skills. It is valuable to businesses for example

- to give flexibility in changing systems platforms or “rightsizing”
- to allow the same application to be used across several business units
- to make it easier for staff to move between departments and between systems

To achieve portability, interfaces must be standard across systems:

To port applications, not only standard languages, but also standard interfaces to services provided by the system for file and database access, communications, the user interface and so on.

Standardisation is rarely complete, and therefore portability is rarely perfect. It is the level at which the interface is standard, which determines the amount and ease of portability: the higher the level, the greater the portability and the greater the resulting value.

In general the higher the level, the less standards exist; the process of developing and agreeing standards has mainly worked “bottom-up”.

Interoperability

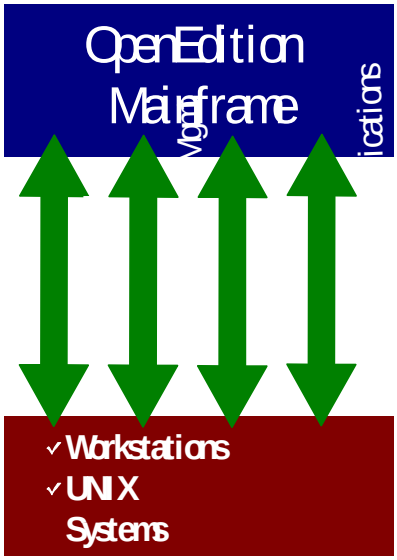
Most companies today have a mix of computer equipment, they also have an urgent business need - to enable people to share data and applications across these systems. This may be ad hoc, like the retrieval of data for use in a PC-based spreadsheet; or it may involve new integrated applications, like a banking system that provides a complete view of all customer accounts.

The return on investment in systems will be improved by wider access to them. So, heterogeneous systems must work well together, that's what interoperability means. Most customers now consider interoperability to be their highest priority for open systems.

Interoperability is about the ability to recognise and use information within data. Just as for portability, interoperability depends upon standards for functions and for interfaces; and over time, the standards are moving up the levels. Much of the interoperation, which is needed and expected by business today, is above the level which the standards have currently reached.

What is OpenEdition ?

OpenEdition combines the power of the workstation, the flexibility of open systems, and the strength of S/390 into a new environment. Offering new, open interfaces for applications and interactive users, OpenEdition supports and fosters a super-environment of larger operating systems or servers and of distributed systems and workstations that share common interfaces.



You can create client server applications primarily developed at the workstation. At the same time, many new application solutions will become available, developed by vendors, application integrators, and also by users at their own workstations.

Applications you create or buy that confirm the POSIX standards for C programs can be moved with ease among computer systems that conform to those standards, such as OpenEdition MVS and VM

With OpenEdition, existing S/390 applications and data are still supported and they are even enhanced. New applications will have access to existing data.

Using OpenEdition,

- users can switch backwards and forwards between the traditional 3270 interface and the new OpenEdition shell interface, an interface which is completely familiar to UNIX users
- UNIX skilled users can interact with the system using a familiar set of standard commands and utilities
- Skilled users can interact with the system, using familiar TSO and CMS commands and interactive menus to create and manage the new hierarchical file system files and to copy data back and forth between MVS data sets and CMS files
- application programmers and users will have both sets of interfaces to choose from and, by can choose to mix these interfaces

Prior to OpenEdition, MVS/ESA™ and VM/ESA™ supported at least 50 national and industry standards. As customer interest in standards has intensified, so has IBM's commitment to standards.

OpenEdition adds significant new functions and standards, these are concentrated in three areas:

- POSIX conformance for portability
- DCE for interoperability
- X/Open XPG/4 for standards compliance
- X/Open for UNIX branding[1] and conformance, greater portability and improved interoperability.

What's the view ?

One of the primary objectives for this document was to provide a measure of the value for the features of and issues relating to OpenEdition. In each major section you will find a summary that discusses the effect and value it brings to OpenEdition.

POSIX

POSIX is a set of standardised services that has been established by the Institute of Electrical and Electronic Engineers(IEEE). There are a wide range of POSIX standards that cover both basic operating system function and function needed by languages and things like common test methods. Some of the standards are now final, formalised and approved standards, some are still in the *drafting* stage.

OpenEdition services for MVS and VM performs the control program services for POSIX.1 process management, file system management, and communication.

[1] Commonly known as Universal Unix, SPEC 1170, XPG 4.2 etc.

In addition to POSIX.1 function, OpenEdition provides a POSIX.2 Shell and Utilities. Using this applications can invoke shell scripts, which in turn invoke shell utility programs. Some of these utility programs make operating system requests and thus, to the operating system, are seen as applications. Process management is controlled by the "kernel" address space. The kernel address space is started as part of OpenEdition initialisation.

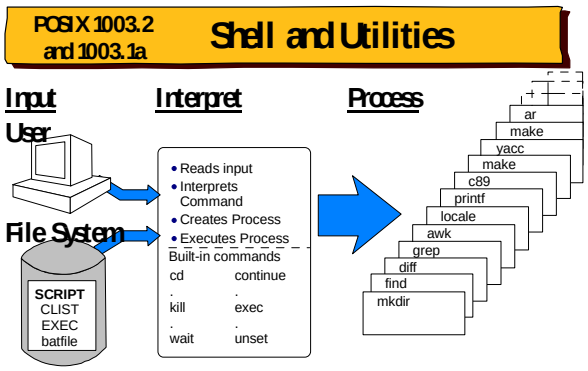
POSIX 1003.1 System Application Program Interface

The System API consists of over 175 functions, many of these are significant, such as the file system support, some are insignificant[2] such as the *tolower()* function. The functions fall into the following areas:

Process Control:	Creation, Execution, Termination, Signals, Timer Operations
Process Environment:	Process/Group/User id's, Time, Environment/System Variables
Files and Directories:	Directories, File Creation/Removal, File Characteristics
Input/Output Primitives:	Pipes File Descriptors Read/Write/Control
Device & Class-Specific Functions:	Terminal Interface Terminal Interface Control Functions

POSIX 1003.2 Shell and Utilities

OpenEdition offers interactive users choice and ability to switch between the familiar interface of the POSIX.2 shell or command interpreter and the traditional 3270 TSO/E and CMS panel or command interface.



Workstation users can be connected to TSO/E or VM/CMS (and the OpenEdition shell) through 3270 emulation or through TCP/IP. User can also now directly access the shell without first going through TSO. An installation can customise logging on so that a user who prefers to work with a UNIX-like interface is put directly into the familiar POSIX.2 shell. From there, the user can choose to switch to the less familiar TSO/E environment (ISPF panels or TSO/E commands) when desired or required.

An interactive user can also choose to log on through the TSO/E or VM/CMS interface and switch to the shell when desired.

The OpenEdition shell is a super-set of the UNIX Korn shell. It provides the ability to write shell scripts, an interpreted programming environment, and can be run interactively or in batch.

There are some 80+ utilities included with the shell. Many of these are built-in and do not require a fork and are therefore quicker. The names of many of the shell utilities will be familiar to UNIX and workstation users and include: mkdir; find; grep; awk; printf; c89; make etc.

POSIX 1003.4a Threads

In POSIX.1 a number of C language functions for controlling program processes are provided so that they can start or interact with other processes.

---

[2] As measured by the amount of programming taken to implement them.

A process can start a second process using the *fork()* function. The original process (or address space) is known as the parent. The new process is known as the child. The child process is almost identical to the parent. Pipes can be used to communicate between processes.

OpenEdition MVS also provides functions from POSIX.1a and a subset of POSIX.4a for threads.

This support provides interfaces to support multiple flows of control called 'threads' within a POSIX.1 Process. The interfaces are provided as POSIX function calls from a C program or as assembler callable interfaces. Threads are defined as multiple separately dispatchable units of work within the process.

In traditional MVS terms a POSIX process is the same as an Address Space and a POSIX thread is an MVS Task or TCB.

OpenEdition VM Threads

VM will implement POSIX.1c Threads , it will substitute the POSIX.1 *fork()* function with the *spawn()* function from POSIX.1d. *spawn()* provides a fast, low-overhead mechanism for creating new POSIX processes.

One Mans View

For existing CMS or TSO users the addition of POSIX to MVS is of little or no use, except for the ability to choose from a wider set of business applications than may have previously been available. As far as the end-user is concerned POSIX just adds a whole set of new but totally unmemorable commands such as Grep, Awk and LS.

However, the addition of POSIX standards is a good move. Many of the other features of OpenEdition build upon POSIX conformance and the POSIX interfaces. POSIX is an essential element needed for UNIX conformance.

Hierarchical File System

The Hierarchical File System is part of the POSIX standard. However, for a number of reasons it is worth separate consideration. The file system OpenEdition file system can be used to create, store, and access files in a tree-like structure containing both files and directories. The hierarchical file system will be familiar to UNIX™, DOS and OS/2 users.

The file system has the concept of a working directory and POSIX commands for creating, changing and removing directories are provided. Data is stored as byte streams, this allows the application to define the data structure.

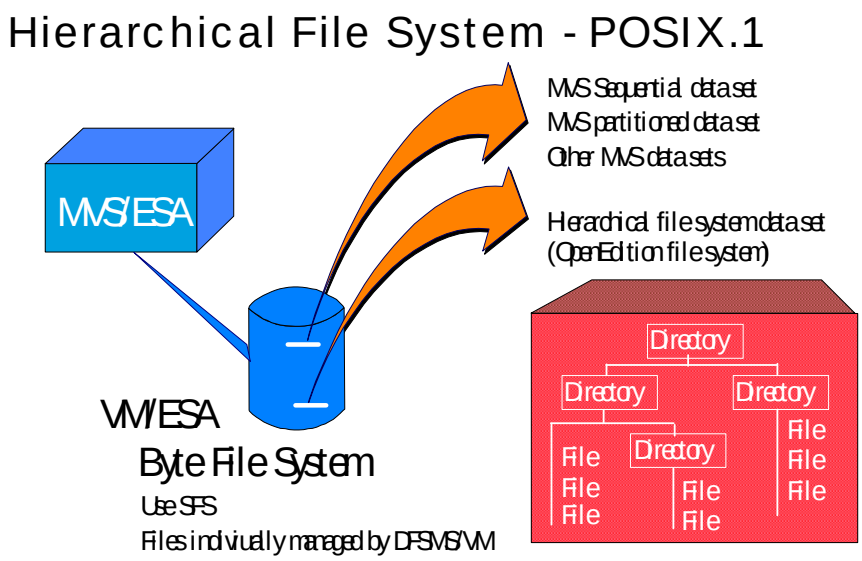
HFS has the following attributes:

- POSIX syscall semantics
- Byte range locking
- Long file names (1024 characters)
- Symbolic links (aliases)
- Pipes
- Concurrent write
  (to the same file from multiple address spaces)
- Permission control

HFS in MVS is a mountable file system. Every MVS system has a single root directory but can have multiple HFS data sets. Each HFS data set can contain its own root directory and hierarchical structure, the HFS data set root directory can be mounted at any point within the system root structure. Each user can create their own mount points for the same or different HFS data sets.

OpenEdition allows the user to take advantage of the many capabilities in MVS/ESA and related products (including RACF and DFSMS™) to:

- Associate files with individual and group ids and limit and control access to assigned users



- Define procedures for automatically backing up and periodically migrating file systems to archival storage
- Define a policy for the storage placement and use of the file systems
- Have built-in integrity associated with MVS/ESA

With the hierarchical file system support provided by OpenEdition MVS

- Terminals and communication pipes are accessed as files
- The hierarchical structure encourages an enterprise and its users to organise sets of files in a natural way (for example, by area, by department, by group of users, by user, and so forth).

Files stored in the HFS can take advantage of the many MVS capabilities and products, including RACF and DFSMS,

- to control access to files to individual users, or groups of users
- to automatically backup data and periodically migrate file systems to archive storage
- to define policies for storage placement and use of file systems
- and of course, the data integrity that you already get with MVS.

Communication services are provided for telecommunication and networking products to be used with or by OpenEdition MVS and generally relate to end user terminal attachment or for program to program communication over a network. So both TCP/IP and ACF/VTAM fall into this category.

VM/ESA Byte File System

OpenEdition VM will provide a hierarchical byte file system compatible with the POSIX standards and integrated with the existing VM Shared File System. A notable difference between the VM/ESA BFS and MVS/ESA HFS is that the VM/ESA BFS will have files managed individually by DFSMS/VM.

Exchangeability of MVS and CMS with POSIX-Organised Data

MVS datasets and CMS files can be copied to or from the OpenEdition POSIX-conforming file system. An ASCII encoded POSIX file can be uploaded or ported into an archive, translated into EBCDIC encoding for MVS or CMS, and stored and processed either in a hierarchical file system or copied to an MVS sequential data set or partitioned data set member or CMS file.

One Mans View

Hierarchical file systems offer a number of benefits not available to traditional TSO and CMS users. The ability to group file by project or by type is a big benefit and is more in-line with the way users work. Adoption by traditional TSO users will be slow though because it is different.

Adoption by CMS users of both the POSIX environment in general, as well as the VM Byte File system will likely be much quicker. I'm certainly looking forward to using the hierarchical file structure that I have today in the VM Shared File System but in a much more familiar way through the use of a current directory and commands such as *MKDIR, CD etc.*

The ability for the VM BFS to manage storage on a file-by-file basis gives it an additional benefit and may offer a significant enough benefit to allow some installations to move some files from workstation systems to VM. This though will depend on the availability of an enhanced NFS Servers for VM.

Distributed Computing Environment

The Open Software Foundations (OSF) Distributed Computing Environment (DCE) differs significantly from the other standards discussed in this document. This is because it is *not* a paper standard. The source code for all of DCE is available from the OSF.

IBM recognises the importance of DCE technologies as the fundamental building blocks required to implement an open distributed system.

DCE Provides

- Operating system independence
- Communication independence
- Location independence
- greatly improved interoperability and portability through a set of common pervasive API's.

These are being incorporated into OpenEdition. The key services of:

- Remote Procedure Call
- Directory Services
- Security Services
- Time Services

will all be provided and integrated into OpenEdition services.

MVS/ESA SP 5.1 and VM/ESA V2.1 will provide DCE Base Services based on the 1.0.3 level of OSF DCE, allowing interoperability with any other DCE enabled system. Additionally a User Data Privacy feature will be provided to support encryption. Currently MVS has committed to implement OSF DCE 1.1.

OpenEdition MVS Application Support

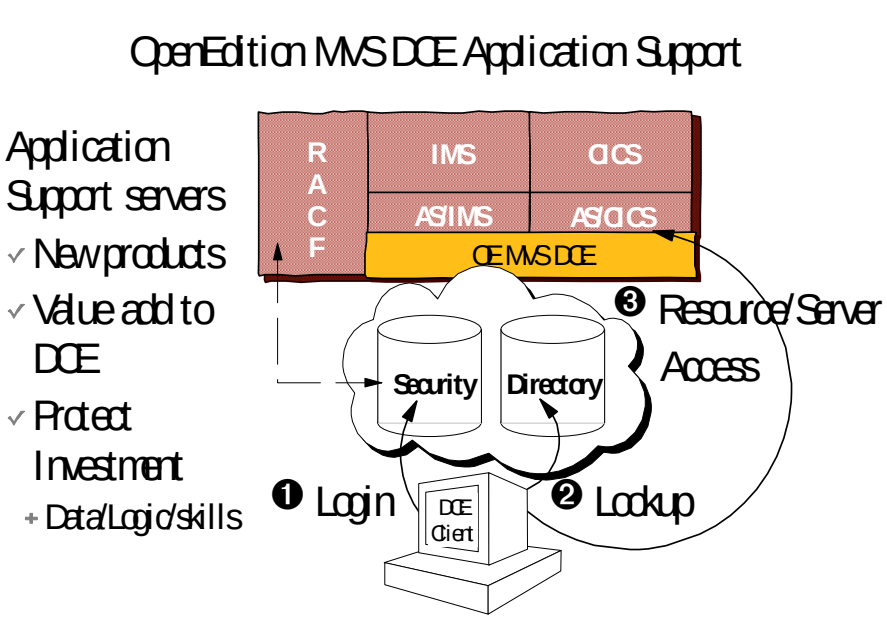
The OpenEdition DCE Application Support servers for IMS and CICS bridge the new, heterogeneous world of DCE to the traditional world of MVS. There are other products on the market today that allow workstations to access CICS and IMS data, and in some cases, their transaction logic as well. But those products involve communication gateways that convert other network protocols into LU 6.2 in order to present CICS and IMS with familiar requests.

The new AS/CICS and AS/IMS servers allow workstation users, using standard DCE Remote Procedure Calls to access CICS and IMS without the communication gateway products.

The application servers are designed to maximise the current investment in COBOL skills and transactions. A structured transaction can be made available as a DCE server in a very short time. New CICS and IMS servers can be written as COBOL transactions. The Client code is written in C as a DCE RPC, which is consistent with the workstation environment. The application servers handle the conversion of C and COBOL data types automatically. The new servers can coexist with all other ways of accessing CICS or IMS from workstations.

One Mans View

The addition of DCE to MVS and VM is indeed very useful. Currently DCE is the best solution to



heterogeneous application interoperability. The range of services offered transparently across a wide spectrum of computing systems make it extremely attractive as a base for distributed application.

In OpenEdition VM, DCE enabled applications will also be able to use and access existing CICS and IMS data and logic via the new Application Support servers.

Application Portability

Using the POSIX and DCE support of OpenEdition, an application programmer can write an application that will run on any conforming operating system.

The Source code can be moved from one system to another and then recompiled to run on the new system. Because operating system interfaces within the application are standardised, this activity of moving (*or porting*) an application from one computer to another computer system is relatively easy and straightforward.

For example, a program could be written and tested on a workstation system and later ported to run (after recompilation) at a host operating system. Or a server program written to run initially at one computer could be ported to another computer. The amount of work to port such a program is typically far less than to rewrite the program for each unique operating system.

An application programmer can choose to take advantage of unique non-standard operating system interfaces in exchange for the advantage of being conforming and more easily portable. The programmer needs to determine that these advantages outweigh the disadvantages of the extra work required to port the program to another system later, however.

In general, an OpenEdition application is also a POSIX application. In terms of how easily an application can be ported to other POSIX-conforming operating systems, POSIX defines three levels of POSIX.1 conformance that an application can have:

- Strictly conforming POSIX.1 application. This is an application that requires only the facilities described in the POSIX.1 standard and in the applicable C language (ANSI) standard. This kind of application must be designed to handle or avoid any kind of behaviour defined in the standard as unspecified or implementation-defined, as well as accept any symbolic constant in the range permitted by the standard.

- Conforming POSIX.1 application. This is similar to the strictly conforming application, except that this application can have option and limit dependencies permitted by the standard. (A statement of conformance must describe these particular dependencies.)

- Conforming POSIX.1 application using extensions. This is an application that differs from the previous kind only in that it uses

non-standard facilities that are consistent with the standard. Such an application must describe all its requirements for these extensions.

The degree to which you want an application to conform to POSIX depends most directly on how likely it is that the application will need to be ported to and run on another operating system in the future.

One Mans View

Currently there are very few strictly POSIX and DCE conforming applications. The improved portability offered by POSIX and DCE conformance is **definitely** a welcome addition to MVS and VM. However, it is increasingly likely that applications will be X/Open XPG compliant rather than just POSIX and/or DCE compliant.

X/Open

X/Open started in Europe and has spread well beyond its early boundaries. Its goal is to establish a set of Application Programming Interfaces(APIs) that both enable portability and interoperability but also facilitate internationalisation. Currently X/Open standards are expressed through documents called X/Open Portability Guides(XPG). The most recent guide is XPG4 and it requires POSIX.1 conformance.

The internationalisation of programs permits running a program in different language/country environments or "locales" without itself being changed. All information regarding languages and cultures is not hard coded in the program but stored externally.

In October 1993, Novell agreed to give the rights to the UNIX name to X/Open so that all vendors could develop to the UNIX standards and use the UNIX name for their products. X/Open will use conformance testing to approve applications and systems that want to be *UNIX* branded. In order to establish what UNIX standards are, X/Open drafted and formalised a new standard. The standard, called during its draft stage SPEC1170, and now commonly referred to as Universal UNIX, has been incorporated into level-2 of the XPG4 standard.

One Mans View

MVS/ESA 5.2.2, which shipped in September 1995, contains around 90% of the APIs within the universal UNIX specification. OS/390 achieved The Open Group (X/Open) UNIX'95 branding in 1996.

Given the current level of industry and user interest in the single Unix specification it is likely that a large number of *UNIX* conformant applications will be available in 1996. Although applications will still need to be recompiled to run on most UNIX systems, support for Universal UNIX APIs will increase the number and selection of applications available.

Although it is still early days for OpenEdition VM, its successful adoption is bound to lead to the inclusion of additional standards.

End User Access/Interface

With the addition of a POSIX conforming shell and utilities in OpenEdition, the upcoming addition of *Raw-mode* or Character-mode terminals to OpenEdition MVS via its support for ASCII terminals, a number of issues arise over the user interface.

The end-user interface is a critical success factor in the use and adoption of any application. Support for terminals that can be driven character by character (raw), through the use of a curses library, is essential to support the majority of UNIX based applications.

On the other hand, there are a wide range of terminals and workstations already in use within the enterprise, including real and emulated 3270 terminals, so how do people see OpenEdition ?

First, the shell is what an end user see at the terminal. It includes the layout of the screen, the

commands available and their syntax. The POSIX shell (as defined by the 1003.2 standards) is the **korn** shell. The POSIX shell will be familiar to UNIX users and provides at least some functions that will be familiar to PC-DOS and OS/2 users.

In its first implementation, the shell limited to 3270 terminals using block-mode terminal support. This means that the end-user types in characters and then presses either the ENTER key, a PF key, or other 3270 Action key such as PA2, CLEAR etc. At this point, an interrupt is sent to the mainframe system and the system reacts accordingly. The addition of ASCII mode terminals or raw-mode, every character typed or every move of the cursor can trigger an effect on the program that is running. Raw-mode is also known as *non-canonical* screen handling.

Raw-mode support allows users to *telnet* or *rlogin* from remote character mode terminals via a TCP/IP sockets network. This eliminates the need for the user use the access a TSO session before starting the shell. To manage the workload that could be generated by raw-mode terminals, MVS has announced support for a communications server. This is an IBM RISC System/6000 based server that multiplexes the characters from many terminals and provides them as a buffer to MVS, allowing MVS to more efficiently process each keystroke. In addition, some of the TCP/IP network processing is also offloaded to the communications server.

Support for ASCII or raw-mode terminals from programs is provided by the *curses* library. Curses can be used to highlight portions of the screen, control colours, the movement of the cursor etc. An often quoted example of raw-mode terminals is the UNIX *vi* editor.

Terminal Device	Software	Connection	Shell	Mode
3270 Terminal		FEP 3174 etc.	OMVS	Block
Workstation	Emulator (PC3270, TN3270)	FEP 3174 etc.	OMVS	Block
	Rlogin	FEP/ OCS	Rlogin	Block/ Raw
	Telnet	OCS	Telnet	Block/ Raw
	X-Server	FEP	X-client	Graphical
ASCII		OCS	OCS	Block/ Raw
X-Terminal	Rlogin	FEP/OCS	Rlogin	Block/ Raw
	Telnet	OCS	Telnet	Block/ Raw
	X-Server	FEP	X-Client	Graphical

The preceding table shows how vary devices can access the OpenEdition shell.

One Mans View

Actually logging-in to the OpenEdition shell is an odd experience for existing TSO and VM/CMS users. In either raw-mode or block-mode the system looks unlike anything they are familiar with. In the longer term, if OpenEdition is successful, the CMS and TSO session services that we use today are also likely to become things of the past. People will give them up to use the OpenEdition shell to launch applications and work with their files in a more natural way than either CMS or TSO currently provide.

The skills and application portability provided by the shell is a big advantage. Support for Raw-mode terminals is also crucial, otherwise application portability will be impacted. The communications server seems an ideal comprise between application need and system performance. In a large enterprise of say 3,000 ASCII terminals it would be unrealistic to expect the main

processor to directly handle all the user key-presses. Over time, I guess my wish would be to see the communication server function moved on board the mainframe. This would be done in the same or similar way as the function of LAN servers is provided by the Open Systems Adapter. There are of course cases where this just wouldn't work, where currently it works better and makes more sense to have the communications server local to the users, especially when they are not close to the mainframe.

Software Developers/Vendors

OpenEdition is part of a much wider industry based solution to the huge variety of application programming interfaces, commands, and facilities available on computer systems today.

Often, too much emphasis is placed on *Open* in Open Systems Standards. The real benefit comes from the adoption of *standards* on a wide range of platforms. There are no losers in the adoption of open standards.

Development organisations maximise their investment in training, education and its developers by using the skills they have to develop on many systems; because developers are able to develop code once and then port it to many platforms in the same language and often using the same tools, code will inevitably improve in quality, less defects because things are done once and done right the first time; because there are less defects the developer is free to work on new features and new platforms rather than defect work.

Users of the applications will also benefit, there will be a wider portfolio of applications to choose from; instead of buying a system to run the application you buy an application to run on the systems that you already have; the users have to learn less in order to use systems provided because the provide broadly similarly functions that work in a broadly compatible way. Finally, since the quality of applications is improved development organisations should be able to lower overall software costs.

But what is the reality ?

One Mans View

From the S/390 application developers/vendors it is common to find one of two views: "*What has all this got to do with me ?*" or "*Wow, this is neat stuff, you mean I can do this on the mainframe?*".

I've actually spent time with a number of lead architects and strategists discussing what and how they might use OpenEdition. I'm pleased to say that I know a number of organisations that are developing a mid-term strategy to migrate their products to a common development base, using the standards included in OpenEdition and others. They've seen the light.

On the other hand, I've come away from 2-hour presentations and discussions with software vendors and developers who didn't know what this had to do with them, and their biggest problem remains that you *can* do all this with a mainframe, which they find confusing.

I've also visited, discussed and presented on OpenEdition to a number of UNIX software developers/vendors. The surprise is that they have the same reaction but for completely opposite reasons. Those that are not really interested in moving beyond departmental, or even individual UNIX workstations have the problem, you *can* do this with a mainframe. Others who see the opportunities OpenEdition presents for their products and skills have the view, "*Wow, this is neat stuff, you mean I can do this on the mainframe?*".

What has been the most pleasant surprise has been the realisation and adoption of standards within IBM. This has primarily been as a result of the Open Blueprint and the efforts within IBM's development organisations to standardise on the data formats, communication protocols and application programming interfaces included in the Blueprint. Products are being developed with

portability across the IBM range a key design objective.

A good example of this is the implementation of the OMG CORBA compliant, distributed object request broker being built for MVS. This was first developed for IBM's AIX operating system and then ported to MVS. It uses many of the interfaces discussed in this paper.

Summary - One Mans View

Clearly OpenEdition offers a number of interesting new facilities to the user and programmer alike. In time we will see if OpenEdition is a success. The same time will also show us if the adoption of open standards will also succeed or fail and by what measure. We all have a role to play in this.

As users we need to demand that our applications are developed to open standards to make sure we get the flexibility we need in our business applications.

As developers we must take the opportunity of using the skills we have on the widest range of systems, the adoption of open standards should ensure that we can spend more time developing and less time porting, rewriting function not available, and less time on maintenance and debugging.

As organisations developing applications we must maximise on the investment in development by insuring applications run on the widest range of systems possible.

Finally, if you feel like I do, then we must make sure that we communicate our beliefs that the adoption of open standards are a real benefit and not something that can be measured, in the short-term, on a financial basis. Insist on the standards and features you want from OpenEdition!

References and Bibliography

1 - Introducing OpenEdition MVS
GC23-3010 - available from IBM

2 - OpendEdition MVS DCE Presentation Guide
MDCE package on IBM MKTTOOLS disk.

3 - Wide-open VM by M.Cathcart
VM/ESA User's and Applications Handbook
Published by McGraw-Hill
ISBN 0-07-023703-4.

4 - IBM ES Client/Server & OpenEdition Update.
Jan 24th 1994 by M.Cathcart.

5 - New Thinking in ES Management Servers,
IBM CUA Annual Conference, April 1994 by M.Cathcart

6 - IBM S/390 Open Client/Server Guide,
by M.Cathcart and Leslie Estroff.
GU20-5146

7 - Porting Applications to OpenEdition MVS
IBM ITSC Red-book GG24-4473

8 - Object Technology for MVS
Cathcart's corner on the 'net
http://www.europe.ibm.com/go/s390/e2tech/mc/index.htm

IBM®

Printed in the United Kingdom

Feel free to visit the author on the Net:
http://www.s390.ibm.com/sections/corner/index.htm