Automatic, Network-directed Operating System Software Upgrades: A Platform-independent Approach

by John R. Lawson, Jr.

ABSTRACT

The initial system load (ISL) capability of Digital's layered-product POLYCENTER Software Distribution (formerly known as RSM) version 3.0 provides OpenVMS system managers with a network-directed tool for performing automatic operating system software upgrades. The design of the POLYCENTER Software Distribution product integrates a number of new and varied software architectures to perform the ISL. A description of the POLYCENTER Software Distribution implementation of the ISL for the OpenVMS operating system details the steps of the ISL process. The software's modular ISL mechanism can be expanded for use on other Digital and non-Digital operating systems and hardware platforms.

INTRODUCTION

The POLYCENTER Software Distribution version 3.0 product provides automatic, centrally delivered, network-directed operating system software upgrades through a process called the initial system load.[1] The term initial system load (ISL) has existed for a number of years in various forms and has come to describe the act of loading the operating system software onto brand-new (virgin) systems without the need for locally attached tape drives or other removable media devices. This term, more loosely applied, may also be used to describe other operations such as operating system upgrades.

The ISL technology provides many advantages over the traditional means of performing upgrades. Typically, upgrades are performed one system at a time, at each console by the system manager, who must maintain the correct set of installation media for each client system's unique set of peripherals and answer each question as the upgrade procedure prompts. In a network managed by the POLYCENTER Software Distribution product, operating system upgrades are performed simultaneously. Any number of ISL operations can be invoked by using a single installation medium and often by issuing a single command. In addition, the ISL mechanism can be used for system disk maintenance operations, such as upgrade, replacement, replication, backup, or compression.

The POLYCENTER Software Distribution product can be extended for use with non-Digital operating systems and hardware platforms all controlled from its one user interface. In most cases, no backups

need be performed on the clients' system disks. A halted client system, of course, must be launched into the first step manually.

This paper describes the POLYCENTER Software Distribution version 3.0 product. It begins by discussing the software environment and the software technologies used by the ISL process. It then states the project team's goals for the product. The paper next relates the ISL scheme implemented by the POLYCENTER Software Distribution version 3.0 product for the OpenVMS operating system. The paper concludes with a discussion of the details of expanding the ISL to other platforms. It is assumed that candidate operating systems are capable of at least simple task-to-task communication through the DECnet network (or some emulation), but other communication mechanisms could be devised instead.

SOFTWARE ENVIRONMENT

The POLYCENTER Software Distribution product defines operating system software as everything on the volume that is typically called the system disk. This includes boot files, data files, configuration files, utilities, compilers, layered products, and customizations. It even includes user directories, if they exist on that system disk.

The POLYCENTER Software Distribution product allows the individual operating system to determine the definition of upgrading the operating system software. For the OpenVMS operating system, upgrading could mean the complete replacement of the contents of the system disk volume, including the utilities, compilers, database, and customizations. For the OpenVMS AXP operating system, it could also mean the installation, without touching the rest of the volume, of only newly acquired OpenVMS operating system files and images. For other platforms, it could mean any one of several other techniques. Each platform can define what is needed to perform operating system upgrades or installations.

A network managed by the POLYCENTER Software Distribution product consists of one or more centrally located server systems; each server is responsible for performing certain operating system maintenance functions on its assigned set of client systems. The server system runs the OpenVMS operating system. Client systems run any of a number of operating systems. The POLYCENTER Software Distribution version 3.0 software supports backups, user authorizations, and layered-product installations for clients running the VAX VMS, OpenVMS VAX, OpenVMS AXP, and ULTRIX operating systems. The software includes support for ISL procedures to clients running the VAX VMS, OpenVMS VAX, and OpenVMS AXP operating systems. The software's ISL architecture, however, supports expansion to other operating systems and platforms.

DESIGN OF THE POLYCENTER SOFTWARE DISTRIBUTION PRODUCT

The design of the POLYCENTER Software Distribution version 3.0 product integrates a number of new and varied software architectures. The software development required the cooperation and synchronization of two layered-product and two operating system development groups.

Software Technologies

No single software technology is capable of automatically upgrading system disks. Several must be used in combination. A brief description of a number of such technologies that could be used to implement the ISL process follows.

- Maintenance Operations Protocol (MOP) is a network protocol used to download system software into the memory of adjacent network nodes.

- Remote triggering enables one client system to cause another client system to reboot. The client system must have triggering enabled and have a triggering password defined and known to the server node.

- The load assist agent is a shareable image running in the context of the maintenance operations monitor (MOM) process on the server system. This code permits a server to control and customize (if necessary) the system software being downloaded to the client.

- The local area disk (LAD) protocol allows locally attached disks or container files on server systems to be presented to the local area network (LAN) for use as virtual disks on client systems. The InfoServer is the most common server of the LAD protocol. OpenVMS systems running the POLYCENTER Software Distribution product can also act as LAD servers. A system can be a LAD virtual disk.

- The processor-specific primary bootstrap is a low-level program that is loaded into the memory of a booting client. This program can be loaded from a disk drive, a tape drive, the network interconnect (NI), or read-only memory (ROM). A small but self-contained program, it is capable of communicating with the machine's console subsystem, most of the machine's internal resources, and the system disk from which it loads a secondary bootstrap or the full operating system.

  Note that some operating systems (OpenVMS included) claim that their bootstrap programs are processor-independent. However, if the operating system is under development,

support will be added eventually to this bootstrap for new CPU models and/or hardware variations. Thus the processor-independent bootstrap program from an earlier version of an operating system may not support all the processor types supported by a later version of this same program. Therefore, processor independence is tied to the set of processors supported by that particular version of the operating system. For this reason, the POLYCENTER Software Distribution product specifically stores an image of the bootstrap program in a private directory alongside the container file that houses a virtual system disk (a bootable snapshot of the version of the operating system).

- The system start-up command procedure is a command script that is responsible for bringing the recently booted operating system to its full configuration. In an ISL, this command procedure is tailored to configure only the resources needed to perform the ISL. Sometimes, limited command-level access is allowed, but seldom is full user access or timesharing permitted.

- In the OpenVMS operating system, the BACKUP/IMAGE command can duplicate a system disk either directly from disk to disk, or indirectly from a saveset file (which might be located on tape or across the network) to disk.

- Standalone BACKUP is a self-contained, diskless operating system capable of executing BACKUP/IMAGE commands but not capable of network operations.

- The SYS$UPDATE:VMSKITBLD.COM procedure in the OpenVMS operating system is used to create a generic system disk, using the current system disk software as a model.

- The POLYCENTER Software Installation (PCSI) utility is capable of creating or upgrading a system disk from a configuration file and a description file (possibly located at a remote network location) or the current system disk.

All these software technologies are utilized in the POLYCENTER Software Distribution ISL, except standalone BACKUP and SYS$UPDATE:VMSKITBLD.COM, because the former cannot be used remotely, and the latter produces uninteresting system disks. Anyone expanding the ISL, however, can use whatever techniques they choose, including those two.

Goals for the ISL Process

The development team had the following goals for the POLYCENTER Software Distribution implementation of the ISL process.

- The process must be totally automatic; only halted client systems are permitted to require human intervention.

- Multiple ISL processes must run concurrently. No specific limits should be placed on the number running in parallel (except for practical performance reasons).

- The software library must store several operating system images. They can be images of different operating systems and/or different versions of the same operating systems.

- Client systems must not be restricted to specific peripheral hardware.

- The software must make no assumptions about the hardware to which it is delivering software. Whatever configurations are legal to a particular operating system must also be supported by the POLYCENTER Software Distribution product.

- The client software must make no assumptions about the server system directing the ISL. Therefore, it would be inappropriate to store operating system images in BACKUP saveset files, which are unique to the OpenVMS operating system.

- Client software, including temporary system disks, must be taken from the clients themselves. Prepackaged operating system software is discouraged because it becomes obsolete as new versions of the operating system are developed, and because it is rarely capable of being customized by the user.

- The ISL process should be expandable to other operating systems and hardware platforms without changes to the current product.

- The POLYCENTER Software Distribution product should be able to use Digital-supplied distribution media as operating system images, such as the PCSI-based OpenVMS AXP version 6.1 CD-ROM.

- The operating system image should occupy as little disk space as possible.

- The ISL process should work over all valid DECnet network configurations. This requirement was only partially achieved: the LAD protocol works over the LAN only.

From these requirements, two achievements were gained: the totally modular organization and the compatibility with the OpenVMS AXP operating system distribution CD-ROM. The former permits support for other operating system and hardware platforms to be added incrementally. The latter enables system managers to

simply load the latest copy of the distribution media, invoke the PCSI utility to record their configuration choices, and then enter a single command to upgrade all their client systems at once.

DESCRIPTION OF THE ISL STEPS

Regardless of the operating system or hardware platform, the ISL process requires the following simple steps:

- Load a processor-specific primary bootstrap into the memory of the client system.
- Boot a (usually read-only) version of the operating system from some form of temporary system disk.
- Determine the parameters of the ISL to be performed.
- Move the operating system software to the target system disk.
- Initiate the cleanup, configuration, tuning, and reboot of the target system disk (which would then contain the new version of the operating system software).

Figure 1 shows the steps of the ISL process in the POLYCENTER Software Distribution Installation.

[Figure 1 (ISL Process in the POLYCENTER Software Distribution Installation) is not available in ASCII format.]

This section provides a description of each step in the ISL process, contrasting the OpenVMS implementation with the POLYCENTER Software Distribution implementation of the ISL for the OpenVMS operating system. The discussion includes both the traditional standalone installation based on the BACKUP command and the OpenVMS-defined ISL or upgrade based on the PCSI utility. The PCSI-based upgrade very closely resembles the ISL process of the POLYCENTER Software Distribution version 3.0 product.

The modular layout of the POLYCENTER Software Distribution implementation and the extension of the ISL to other operating systems are discussed in the section Platform Independence.

Processor-specific Primary Bootstrap

The primary bootstrap is responsible for establishing the connection needed between the client system and the temporary (or virtual) system disk, wherever that might be maintained.

OpenVMS Implementation. If the distribution medium is local,

then the ROM bootstrap or a bootstrap file on the medium is sufficient to boot the operating system contained there. If the distribution medium is served to the LAN by an InfoServer system, then the bootstrap image must be downloaded from an adjacent DECnet node with service enabled on its NI circuit common to that client, using MOP. In OpenVMS AXP, this image is called APB.SYS; in OpenVMS VAX, it is ISL_SVAX.SYS or ISL_LVAX.SYS (for small or large VAX systems, respectively).

The system manager requests the image to be downloaded (at the console of each client system) by entering special processor-dependent boot commands. The MOM process on the adjacent node drives the MOP delivery of the bootstrap image to each client. Before the connection between the client system and the temporary system disk can be established, the system manager must navigate a series of menus to select the name of the InfoServer service under which that distribution medium is presented.

POLYCENTER Software Distribution Implementation. The client system boots from its NI adapter, generating a MOP load request. The server keeps the client's hardware NI address in its database so it can detect and process this request. This activates the load assist agent (LAA) under the MOM process. The LAA retrieves the various answers to the operating system configuration questions from the POLYCENTER Software Distribution library. It then passes those answers plus the operating system version-specific primary bootstrap (APB.EXE for OpenVMS AXP or ESS$ISL_VMSLOAD.EXE for OpenVMS VAX) back to the MOM process to be downloaded to the client's memory. Among these configuration answers are the name and password of a LAD service, presented by the server in a file containing the temporary system disk. This primary bootstrap establishes the logical connection between the client system and this virtual system disk.

Temporary System Disk

The temporary system disk is a tailored copy of the operating system to be installed. This system disk is usually customized in such a way that its only purpose is to perform the ISL.

OpenVMS Implementation. If it is mounted locally, the temporary system disk is the distribution medium. If the medium is presented to the client by an InfoServer service, then the temporary system disk is a virtual disk bound to that LAD service. In either case, the temporary system disk is mounted as read-only.

The operating system booted from that medium is either standalone BACKUP (for traditional installations) or OpenVMS (for PCSI-based ISL installations or upgrades). Since standalone BACKUP can

perform only BACKUP operations, there is an extra, time-consuming step. The system manager must enter an appropriate BACKUP/IMAGE command to move a portion of the operating system software (the so-called REQUIRED saveset file) to the target system disk and then boot onto the target system disk (containing this partial OpenVMS operating system) to continue the installation.

POLYCENTER Software Distribution Implementation. The temporary system disk always contains the full operating system to be installed. In most cases, this temporary system disk is actually a fully functional image of a model system disk taken from another client system by an earlier FETCH OPERATING_SYSTEM command.[2] The fetch process (discussed later) has replaced this temporary system disk's system start-up command procedure with a script that runs the remaining steps of the ISL process.

Previous versions of this product (also known as RSM) included a prepackaged temporary system disk with a fixed contents that was built by hand. Software developers routinely captured the latest versions of OpenVMS system disks inside boot container files as small as 14,000 blocks! Although an interesting academic achievement, this proved to be an impractical approach. Digital releases new processors from time to time, and each new processor requires a new minimum version of the OpenVMS operating system. The system disks captured in the boot container could not be easily upgraded in the field. An engineering change order was required for the POLYCENTER Software Distribution product each time support was added to the OpenVMS system for a new processor.

These previous versions also stored the operating system image in a BACKUP saveset file. This method could be more space-efficient (page, swap, and dump files consume no space in a saveset file), but it violates one of the design goals.

In version 3.0, the software developers eliminated the concept of separate BACKUP saveset files and boot containers. Since the operating system support for the new processors exists in the software saved in the operating system image, the clients can be booted directly from that image. The POLYCENTER Software Distribution version 3.0 product stores the image of the model system disk directly into a container file. This approach produced an interesting side effect. If a particular processor is not supported by the version of OpenVMS saved in the operating system image, it is not possible to boot that processor into the ISL. As a result, an older version of the OpenVMS operating system cannot be installed on hardware that requires a newer version.

Parameters of the ISL

When operating system software is being installed, system

configuration choices must be selected from a number of variables. At a minimum, the name of the target system disk must be known. Answers might also be needed for questions such as: which subsets of the operating system files are to be installed? The ISL procedure must be capable of obtaining these answers, either by prompting a user at the console of the client system or by some automatic means.

OpenVMS Implementation. At this point, the OpenVMS operating system is running, and a special system start-up command procedure has control. The system manager now answers a series of prompts at the console. Only rarely does the upgrade procedure ask all its questions at once (and state that it is finished asking questions) before commencing any time-consuming tasks. If it did, the system manager could leave the console of one machine to move to the console of the next machine and so on. In this way, multiple upgrades could be performed concurrently.

POLYCENTER Software Distribution Implementation. The parameters of the ISL were downloaded along with the primary bootstrap image. The system start-up procedure of the ISL executes a program that locates the list of parameters in memory and returns them as logical names, which are easier for command procedures to manipulate. (Other operating systems would use their own easily accessible data storage mechanisms.)

The system start-up procedure starts the DECnet networking software and establishes a network connection with the POLYCENTER Software Distribution server system, permitting access to larger amounts of data than might fit into the bootstrap image. The BACKUP saveset file used by previous versions of the POLYCENTER Software Distribution product was accessed through this DECnet connection.

Move the Operating System Software

Each operating system has specific requirements for creating or duplicating system disks. This step uses the client operating system's standard procedure to duplicate or upgrade the target system disk, generally using the temporary system disk as its source (or model). However, another means, such as network files or library files, may be used.

OpenVMS Implementation. From this point, there is no difference between an upgrade and an installation using the traditional standalone OpenVMS mechanisms.

The OpenVMS mechanism now performs a series of complex file replacements in a peculiar order, which requires several reboots to complete. This maximizes the existing free space on the target

system disk. After all the reboots have completed, the old operating system files will have been deleted, and the new files will have been delivered.

The PCSI-based upgrade does not need to perform the several reboots, since the target system disk is treated as a data disk. Its operating system files are simply replaced with new versions taken from the temporary system disk. This is one reason that the PCSI-based OpenVMS upgrade is faster than the traditional OpenVMS upgrade.

POLYCENTER Software Distribution Implementation. Since full OpenVMS (including DECnet) is running, all the resources of the OpenVMS operating system are available for manipulating the target system disk, which is also treated as a data disk. Alternatives such as VMSKITBLD.COM (which creates duplicate basic system disks), the BACKUP/IMAGE command (which duplicates system disks in their entirety), and the PCSI utility (which upgrades system disks in place) could be utilized at this point.

The BACKUP/IMAGE command moves the image of the temporary system disk to the target system disk. The PCSI utility replaces the operating system files on the target system disk with the new operating system files from the temporary system disk. In the BACKUP/IMAGE case, any system-specific customizations or layered-product files that were saved into the container file by the FETCH OPERATING_SYSTEM process are now in place. In the PCSI case, however, all system-specific customizations or layered-product files are left undisturbed.

Cleanup, Configuration, Tuning, and Reboot

Any final changes needed before allowing the client to use its new system disk are performed during the cleanup, configuration, tuning, and reboot phase. The client now boots from its newly upgraded target system disk, and the temporary system disk is no longer needed.

OpenVMS Implementation. As a final step, the AUTOGEN procedure tunes the operating system parameters to the hardware on which it is intended to run. Any other configuration issues (such as the network node name and address) remain as exercises for the system manager to perform at some later time. The system reboots one last time. For traditional installations, this reboot may have been the fifth or sixth. Some of these may have been manual reboots, which require the system manager to issue nonstandard, processor-specific console commands.

POLYCENTER Software Distribution Implementation. When the BACKUP/IMAGE command is used, customizations specific for the

ISL, which are all stored under the [RSM0.] directory tree, must be removed from the target system disk, which, before this step, is a perfect image of the temporary system disk. In addition, the DECnet software must be reconfigured. The DECnet databases still contain the configurations saved in the temporary system disk; these must be updated to reflect the hardware on this client. As a final step, the client reboots onto the target system disk, and the temporary system disk is no longer required. With the PCSI-based ISL, no cleanup is required.

FETCHING AND INSTALLING OPERATING SYSTEM SOFTWARE

The POLYCENTER Software Distribution product keeps images of model operating systems in its private software library. The act of placing software into the library is called a fetch. The act of delivering that software to a client system is called an install. The operating system commands are FETCH OPERATING_SYSTEM and INSTALL or UPGRADE OPERATING_SYSTEM.[3] A model system disk cannot be installed without first being fetched from a client system or suitable distribution medium.

The Fetch Operation

The FETCH OPERATING_SYSTEM command takes a parameter that is the symbolic name of the operating system to be fetched. The POLYCENTER Software Distribution product uses and records this symbolic name because it is the key to a naming scheme used to activate program modules later.

Table 1 lists several symbolic names and the operating systems they might represent. It is important to remember that there is no built-in mapping between these names and the operating systems to which they are mapped. This list is a theoretical sampling of what mappings could be configured on a particular server system.

Table 1 Required Files for Sample Operating Systems

Symbolic Name	Operating System	Required ISL Files
VMS	OpenVMS VAX or VAX VMS	SYS$SHARE:RSM$ISL_INSTALL-VMS.EXE RSM$SDS_DATA:RSM$FETCH-VMS.DSK SYS$SHARE:RSM$ISL_LAA-VMS.EXE
AVMS	OpenVMS AXP	SYS$SHARE:RSM$ISL_INSTALL-AVMS.EXE RSM$SDS_DATA:RSM$FETCH-AVMS.DSK SYS$SHARE:RSM$ISL_LAA-AVMS.EXE
ULTRIX	ULTRIX	SYS$SHARE:RSM$ISL_INSTALL-ULTRIX.EXE RSM$SDS_DATA:RSM$FETCH-ULTRIX.DSK

		SYS$SHARE:RSM$ISL_LAA-ULTRIX.EXE
OSF1	OSF/1	SYS$SHARE:RSM$ISL_INSTALL-OSF1.EXE RSM$SDS_DATA:RSM$FETCH-OSF1.DSK SYS$SHARE:RSM$ISL_LAA-OSF1.EXE
VMS5	OpenVMS VAX with DECnet Phase V	SYS$SHARE:RSM$ISL_INSTALL-VMS5.EXE RSM$SDS_DATA:RSM$FETCH-VMS5.DSK SYS$SHARE:RSM$ISL_LAA-VMS5.EXE
AVMS5	OpenVMS AXP with DECnet Phase V	SYS$SHARE:RSM$ISL_INSTALL-AVMS5.EXE RSM$SDS_DATA:RSM$FETCH-AVMS5.DSK SYS$SHARE:RSM$ISL_LAA-AVMS5.EXE
WINDOWS	MS-DOS running Microsoft Windows	SYS$SHARE:RSM$ISL_INSTALL-WINDOWS.EXE RSM$SDS_DATA:RSM$FETCH-WINDOWS.DSK SYS$SHARE:RSM$ISL_LAA-WINDOWS.EXE

When processing an INSTALL OPERATING_SYSTEM AVMS command, the POLYCENTER Software Distribution product uses the OpenVMS run-time library routine LIB$FIND_IMAGE_SYMBOL to dynamically activate the shareable image SYS$SHARE:RSM$ISL_INSTALL-AVMS.EXE. This image is called the ISL Director. It is used for both fetch and install operations. The POLYCENTER Software Distribution product calls the ISL Director routine RSM$ISL_FETCH and passes to it a context data structure (described in the section Platform Independence). This routine uses the software's remote command execution agent (CEA) to issue Digital command language (DCL) commands on the client system. Non-OpenVMS clients would need to implement their own communications mechanism, so that the server system could direct the client to perform any required actions.

These DCL commands cause the client to mount the LAD virtual disk presented from the fetch toolkit container file RSM$SDS_DATA:RSM$FETCH-AVMS.DSK. The client executes the command procedure [RSMV3.0]RSM$ISL_BOOT-AVMS.COM from the fetch toolkit virtual disk. This command procedure

- Determines the size of the client's system disk
- Reports that system disk size to the ISL Director

  The server creates an appropriately sized LAD container file to receive the snapshot of the client's system disk and serves it to the client.
- Mounts the new virtual disk
- Issues a BACKUP/IMAGE command to copy the system disk to the virtual disk
- Provides the server with access to the processor-specific primary bootstrap image (APB.EXE)

The server saves the APB.EXE image in its library alongside the newly created container file.

- Customizes the virtual disk so it can be used as the temporary system disk during an ISL

  The boot command procedure uses programs and command procedures from the fetch toolkit virtual disk to accomplish this step. In a FETCH OPERATING_SYSTEM AVMS, this final step includes creating a special system root [RSM0.SYSEXE], placing a private system start-up command procedure [RSMV3.0]RSM$ISL_STARTUP-AVMS.COM, installing a program to retrieve the parameters of the ISL [RSMV3.0]RSM$ISL_CLIENT-AVMS.EXE, and installing a command procedure to remove these customizations [RSMV3.0]RSM$ISL_CLEANUP-AVMS.COM.

The two virtual disks are then dismounted, and the server closes the container file and makes it available, write-protected, for ISL operations. These LAD services can be accessed with binary passwords known only to POLYCENTER Software Distribution servers, so no casual access to the data contained within is ever allowed.

The Install Operation

The POLYCENTER Software Distribution product retrieves the symbolic name of the operating system (e.g., AVMS) from the database. The software product uses the symbolic name to activate the ISL Director image (SYS$SHARE:RSM$ISL_INSTALL-AVMS.EXE) and passes control to its universal routine RSM$ISL_INSTALL. This routine enables the LAA (SYS$SHARE:RSM$ISL_LAA-AVMS.EXE) and prepares a data file RSM$SDS_WORK:ISL_client.DAT for use by the LAA after the client system requests it to be downloaded.

If a DECnet connection is possible between the server system and the client system, then the command execution agent issues appropriate shutdown and reboot commands to launch the ISL. If not, the POLYCENTER Software Distribution process assumes that the client is halted and that the system manager will launch the client into the ISL manually.

When the MOM process detects the client's NI address, it activates the LAA and passes control to the routine at offset 0000 in the image. The parameters to this procedure call (which are described in the section Platform Independence) include the node name of the client system and the address of a callback routine used to deliver the bytes of the bootstrap image to the client. The callback routine

- Reads the RSM$SDS_WORK:ISL_client.DAT file (described in the section Platform Independence)

- Retrieves the processor-specific bootstrap image (APB.EXE) from the library
- Locates and writes the parameters of the ISL into the bootstrap image's work space
- Releases these bytes to MOM for delivery to the client

Once this is downloaded, the server system assumes a passive role, waiting for the client to announce its own completion.

The processor-specific bootstrap image has control of the client system. It locates the LAD service name and password in the parameters of the ISL to establish the connection to the temporary virtual system disk (which is being presented by the server system) and boots the OpenVMS AXP operating system.

The system start-up command procedure (RSM$ISL_STARTUP-AVMS.COM) then receives control and

- Starts enough of the OpenVMS operating system to mount local disks and start the DECnet networking software
- Executes the program RSM$ISL_CLIENT-AVMS.EXE to retrieve the ISL parameters

With the parameters of the ISL stored in logical names, the system start-up procedure then

- Configures the target system disk
- Initializes the target system disk if necessary
- Starts the DECnet networking software
- Solicits further instructions (if any) from the server system
- Issues a BACKUP/IMAGE command to move the operating system software from the temporary system disk to the target system disk
- Executes the RSM$ISL_CLEANUP-AVMS.COM command procedure to remove the customizations specific for the ISL

The target system disk now appears to be identical to the model system disk just before the fetch operation.

The UPGRADE OPERATING_SYSTEM and FETCH CONFIGURATION Commands

The contents of the PCSI-installable distribution medium for OpenVMS AXP bears a striking resemblance to a POLYCENTER Software Distribution temporary system disk. This is no coincidence. The

OpenVMS AXP development team modeled the distribution medium after the POLYCENTER Software Distribution boot container, so the product would be plug-compatible. The obvious difference, however, is that the system start-up procedure invokes the PCSI utility instead of the BACKUP/IMAGE command.

The client system boots from the distribution medium under the direction of the POLYCENTER Software Distribution product. Next the procedure starts the DECnet network software using the parameters of the ISL. Then the PCSI configuration answers are taken from the server system rather than being prompted manually at the console. Everything else is the same.

Before any of this is possible, however, the system manager invokes the PCSI utility to record the answers to all the configuration questions using the RSM$TRIAL_INSTALL.COM command procedure. The PCSI configuration file is then inserted in the POLYCENTER Software Distribution library using the FETCH CONFIGURATION command.

Note that when recording configuration files, the PCSI utility permits users to defer answers until installation time. Unfortunately, because of the product's stipulation that no human intervention be required, such deferrals cause the ISL to fail.

PLATFORM INDEPENDENCE

The following section details how the ISL process can be expanded to other platforms and operating systems. Table 1 gives a sample list of symbolic names and their corresponding operating systems. The POLYCENTER Software Distribution version 3.0 kit provides only the VMS (for the VAX VMS and the OpenVMS VAX operating systems) and the AVMS (for the OpenVMS AXP operating system) ISL kits.

To add ISL support for other operating systems and/or hardware platforms, the following requirements must be met.

- The operating system must be bootable from a read-only LAD virtual disk. (Among others, the MS-DOS, ULTRIX, and DEC OSF/1 operating systems are known to have this capability.)

- The hardware platform must be MOP downloadable. (Most Digital processors have this capability.)

- The operating system's processor-specific bootstrap image must have an LAA-writeable scratch area for the parameters of the ISL.

- The parameters of the ISL must be retrievable by the operating system's system start-up command procedure.

- The operating system must have a mechanism for moving the contents of the temporary system disk to the target system disk, which will never be identical media. (Most operating systems have this capability.)

- The ISL Director shareable image (SYS$SHARE:RSM$ISL_INSTALL-opera.EXE), containing entry points RSM$ISL_FETCH and RSM$ISL_INSTALL, must be active on the server system (running OpenVMS).

- The contents of the fetch toolkit container file (RSM$SDS_DATA:RSM$FETCH-opera.DSK) need be known only to the ISL Director. This file resides on the server system (running OpenVMS) but is read only by the client system and only during a fetch operation.

- The load assist agent (SYS$SHARE:RSM$ISL_LAA-opera.EXE) must be capable of delivering the operating system's processor-specific primary bootstrap image (plus the parameters of the ISL) to the client system, which runs on the server system (running OpenVMS).

Table 1 lists the names of the files required to support various operating systems. Note the naming scheme for the files. Each set of three files, which compose a single ISL kit, implements the entire ISL fetch and install functionality. The ISL Director routine RSM$ISL_FETCH works in conjunction with the fetch toolkit. The ISL Director routine RSM$ISL_INSTALL works in conjunction with the load assist agent. Table 2 gives the naming convention used for all resources shared between these three files. The term opera identifies the symbolic name of the operating system. The term server identifies the DECnet node name of the server system, and the term opsys identifies the user-defined pseudonym for the fetched operating system image.

Table 2 Naming Conventions Used by ISL Resources

Name	Description
RSM$ISL_INSTALL-opera.EXE	Shareable image containing the ISL Director routines, which runs on the POLYCENTER Software Distribution server (running OpenVMS).
RSM$ISL_LAA-opera.EXE	Shareable image containing the load assist agent, which runs on the POLYCENTER Software Distribution server (running OpenVMS).
RSM$FETCH-opera.DSK	Container file containing the fetch toolkit, which resides on the POLYCENTER

	Software Distribution server system (running OpenVMS) but is read only by the client system.
RSM$FETCH_server-opera	LAD service name for the fetch toolkit, which is served by the POLYCENTER Software Distribution server (running OpenVMS) to the client.
RSM$ISL_BOOT-opera.COM	Command procedure responsible for actually performing the save of the operating system software from the client's system disk to the virtual disk, which runs on the client system.
RSM$SDS_OS_LIBRARY: [opsys.OPERSYS]SYS0.DSK	Container file for the fetched operating system, which resides on the POLYCENTER Software Distribution server system (running OpenVMS). (This directory may also be used to store the bytes of the processor-specific bootstrap image so the load assist agent has easy access.)
RSM$ISL_server-opsys	LAD service name of the temporary system disk containing the fetched operating system image, which is served from the POLYCENTER Software Distribution server system (running OpenVMS) to the client system.
RSM$ISL_STARTUP-opera.COM	Command procedure responsible for actually delivering the operating system software from the temporary system disk to the target system disk. It runs on the client system but is booted from the temporary system disk.
RSM$ISL_CLEANUP-opera.COM	Command procedure for removing customizations specific to the initial system load from a temporary system disk. It runs on the client system but is booted from the temporary system disk.
RSM$ISL_server	LAD service name of the VAX "boot container" for operating systems fetched prior to version 3.0, which is served from the POLYCENTER Software Distribution server system (running OpenVMS) to the client system (which in this case must be running OpenVMS VAX or VAX VMS).
RSM$ISL_server_EVMS	LAD service name of the AXP "boot

container" for operating systems fetched prior to version 3.0, which is served from the POLYCENTER Software Distribution server system (running OpenVMS) to the client system (which in this case must be running OpenVMS AXP).

The ISL Director

The ISL Director is a shareable image activated by LIB$FIND_IMAGE_SYMBOL; therefore it need not have transfer vectors, as long as the two required entry points are declared UNIVERSAL. These two routines are called in user mode. They are passed a single parameter, the address of a data structure called the QENTRY.

The pertinent fields of the QENTRY data structure passed to RSM$ISL_FETCH are

```
.
.
.
char pseudonym[64];
.
.
.
char client_node[128];
.
.
.
char library_node[128];
.
.
.
char opera_house[8];
.
.
.
```

and the pertinent fields of the QENTRY data structure passed to RSM$ISL_INSTALL are

```
.
.
.
char ethernet[19];
.
.
.
char client_node[128];
```

```
.
.
.
char library_node[128];
.
.
.
char opera_house[8];
.
.
.
```

In both routines, the field called opera_house contains the symbolic name of the operating system (e.g., AVMS).

RSM$ISL_FETCH is responsible for copying a bootable snapshot of the client's system disk into the LAD container file SYS0.DSK. The LAD virtual disk should be organized into the native format of the operating system being fetched. The server system will never attempt to read these files. To the server system, this container is simply a large series of bytes, whose meaning (to the client system) is unimportant. This routine is responsible for obtaining the size of the container file to be created, creating that container file, and then serving it, writeable, to the LAD. Once the fetch operation has concluded, the container should be served again in read-only format.

RSM$ISL_INSTALL is responsible for enabling the LAA for the new client system. Since the LAA runs under the MOM process, which is a non-POLYCENTER Software Distribution environment, this routine should also collect any and all information (such as the DECnet node name and address of the server system) needed by the LAA, and store that information in the file RSM$SDS_WORK:ISL_client.DAT. The content of this file is shared only between RSM$ISL_INSTALL and the LAA; therefore, the format of the file is implementation-dependent.

The Fetch Toolkit

The fetch toolkit is also a LAD virtual disk organized in a format that is native to the client's operating system. Again, the server system will never read this virtual volume. This virtual volume contains the native operating system pieces necessary to save a snapshot of the model system disk, make it bootable as the temporary system disk, and restore it to its original state. These are usually three separate command procedures. The command procedure that saves the system disk image must also store the bytes of the operating system's primary bootstrap image for future access by the LAA.

The Load Assist Agent

The LAA delivers the bytes of the processor-specific primary bootstrap image to the client system. The MOM process activates this shareable image dynamically, but not using LIB$FIND_IMAGE_SYMBOL. Therefore, the one required entry point to this image must occur at offset 0000 in the image. (The name of the entry point is unimportant.) This is best accomplished using a single transfer vector.

This routine is called in user mode with three parameters, the addresses of three data structures: the MOMIDB, the MOMARB, and the MOMODB.

The offset MOMIDB$A_PARAM_DSC contains any text from the NCP load assist parameter field. This field contains arbitrary text that RSM$ISL_INSTALL placed there. Normally, this field contains a handle used to retrieve the file RSM$SDS_WORK:ISL_client.DAT. A good handle is the DECnet node name of the client system.

The offset MOMARB$A_SEND_DATA is the address of a routine to deliver data to the client. The LAA need only collect and/or generate the data to be delivered to the client; this callback routine delivers it to the client. Its two parameters are a string descriptor identifying which and how many bytes are to be delivered, and the relative address in the client's memory to place these bytes. This callback routine may be called repetitively.

The offset MOMODB$L_TRANSFER_ADDRESS must be filled with the relative transfer address of the processor-specific bootstrap image that was loaded into the client's memory by MOMARB$A_SEND_DATA. For OpenVMS VAX, this offset is traditionally zero, because certain older VAX processors are not capable of using any other value. That is one reason why the transfer address for ISL_SVAX.SYS is always zero.

SUMMARY

The ISL mechanism installs, maintains, and upgrades operating system software. These simple descriptions provide the framework for expanding the ISL process implemented in the POLYCENTER Software Distribution version 3.0 product to platforms other than OpenVMS VAX and OpenVMS AXP operating systems. This expansion can make work easier for system managers of multiple platforms and may even start a de facto standard for performing operating system upgrades.

ACKNOWLEDGMENTS

I would like to thank Richard Bishop and Charlie Hammond in the OpenVMS AXP Development Group for allowing me to unify the POLYCENTER Software Distribution version 3.0 ISL and the

PCSI-based OpenVMS AXP version 6.1 upgrade.

NOTE AND REFERENCES

1. POLYCENTER Software Distribution is the new name for Digital's Remote System Manager product. The installed software continues to use its traditional acronym RSM.

2. POLYCENTER Software Distribution Management Guide (Maynard, MA: Digital Equipment Corporation, Order No. AA-JG05E-TE, May 1994).

3. POLYCENTER Software Distribution Command Reference (Maynard, MA: Digital Equipment Corporation, Order No. AA-JG03E-TE, May 1994).

BIOGRAPHY

John R. Lawson Jr. John Lawson joined Digital in 1984. He has been a member of the OpenVMS VAX Development Group and the POLYCENTER Software Distribution Development Group. His code exists in several layered products and in the OpenVMS VAX and OpenVMS AXP operating systems. He holds a B.M. degree from the Eastman School of Music (1984) and a B.S. in software engineering from the University of Rochester (1986). He is currently pursuing an M.S. in mathematics and computer science from the Colorado School of Mines. John has a U.S. patent pending for a unique sorting algorithm.

TRADEMARKS

The following are trademarks of Digital Equipment Corporation: AXP, DEC, DEC OSF/1, DECnet, Digital, InfoServer, POLYCENTER, OpenVMS, ULTRIX, VAX, and VMS.

Microsoft is a registered trademark and Windows is a trademark of Microsoft Corporation.

OSF/1 is a registered trademark of the Open Software Foundation, Inc.