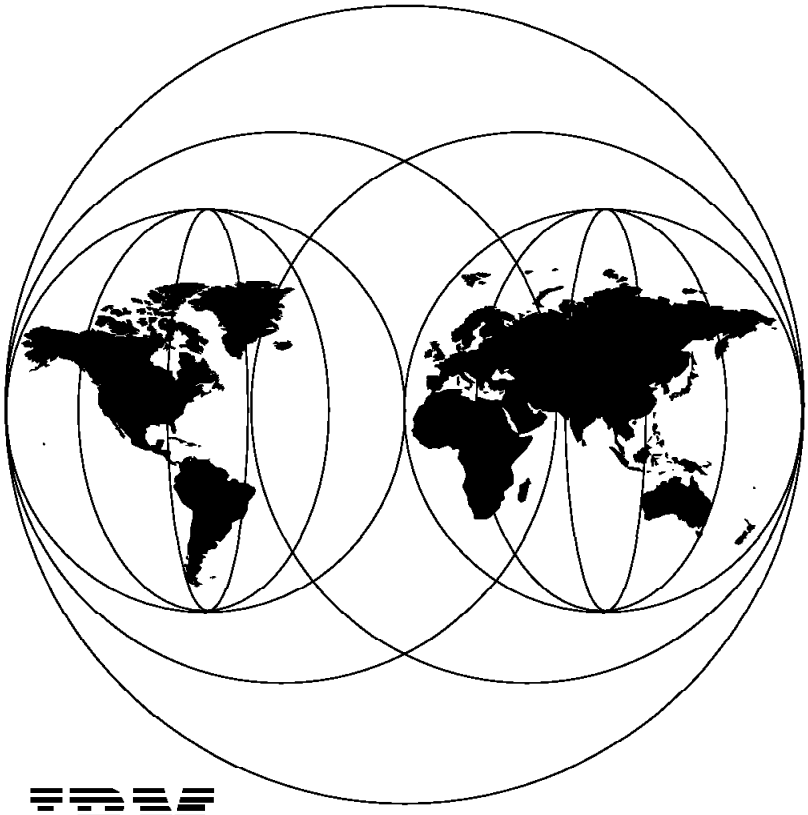
SG24-4775-00

OS/390 Software Management Cookbook

October 1996



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4775-00

OS/390 Software Management Cookbook

October 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix F, “Special Notices” on page 213.

First Edition (October 1996)

This edition applies to Release 1 of OS/390 (5645-001) and to all subsequent releases and modifications until otherwise indicated in new editions.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Preface

This redbook describes the tasks necessary to maintain an OS/390 environment, using SMP/E and IBM custom-built offerings. It focuses on topics that include maintenance philosophy; system design for maintenance; and software installation, testing, and implementation. The entire document may be read to gain an overall understanding of large systems maintenance. Alternatively, specific sections may be read as the need arises during a system update.

This redbook is written for systems programmers involved in the installation and maintenance of an OS/390 operating system. Some knowledge of OS/390 is assumed.

How This Redbook Is Organized

This redbook contains 241 pages. It is organized as follows:

- Chapter 1, “Introduction”

  This chapter introduces you to software management, definitions, and delivery offerings.

- Chapter 2, “Maintenance Philosophy”

  This chapter helps you determine the most appropriate software management strategy for your business.

- Chapter 3, “Maintenance Environment”

  This chapter describes the environment you should set up to best enable you to maintain your software.

- Chapter 4, “Installation”

  This chapter helps you determine whether to upgrade or replace your existing software, and the IBM offerings that are available to help you accomplish this task.

- Chapter 5, “Testing”

  This chapter briefly examines answers to the questions, “What is testing and why should I do it?”

- Chapter 6, “Production and Implementation”

  This chapter describes the many activities needed to get your upgraded software into production.

- Chapter 7, “Subsystem Topics”

  This chapter describes those software management tasks that are associated with maintaining a subsystem environment.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

This project was designed and managed by:

Bob Haimowitz	International Technical Support Organization, Poughkeepsie Center
Paul Rogers	International Technical Support Organization, Poughkeepsie Center

The authors of this document are:

Robert Ayres	Lloyds Bank UK
Paola Bari	IBM SEMEA
Fatima Cavichione	IBM Brazil
Edilson Fuzetti	IBM Brazil
Stuart Hawley	IBM UK
Chris Hollis	IBM UK
Elba Van Zyl	IBM South Africa

This is book is a major revision of the *MVS Software Management Cookbook*, GG24-3481-02. The authors of the this book wish to thank the authors of the previous book.

Thanks to the following people for their invaluable contributions to this project:

Rainer Assling	IBM Informationssysteme GmbH, Hamburg, Germany
Dave Clitherow	IBM UK
Rich Conway	ITSO Poughkeepsie
Greg Daynes	IBM US
John Eells	IBM US
Tony Newling	ITSO Poughkeepsie
Linda Richter	IBM US

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Introduction

This chapter discusses the following topics:

- Why read this book?
- Issues
- Software management definition
- OS/390 Software Delivery offerings
- Reader's guide
- Some limitations of this book
- Assumptions

1.1 Why Read This Book?

The 1990s are presenting a number of challenges to enterprises using information systems. As your business becomes more dependent on information systems, the requirements for high availability and continuous operation become more important. Also, in difficult economic times, there are increasing pressures to lower the costs of doing business. These costs include those of installing, maintaining, and operating information systems.

A key discipline that assists in addressing both of these challenges is software management, which is planning for, installing, and maintaining systems software. This publication examines the tasks of software management, and what can be done to simplify those tasks. Simply stated, the better your software management, the higher the availability of your system, and the lower your costs to install and maintain the system.

One of the goals of OS/390 is to help the customer improve their software management by supplying a fully tested software platform, thus helping them to reduce the cost and complexity of their software management.

As in the past, IBM supplies both upgrade and replace paths for customers to upgrade their OS/390 systems. However, the emphasis with OS/390 is very much towards replace as the preferred path. This path will best enable the customer to reap the maximum benefit from the integrated, tested software platform that is OS/390.

1.2 Issues

The issues that must be dealt with in large systems include:

- The increasing rate of change of large systems software. IBM is delivering "more function, more often." New function should be easy to install and absorb, but outdated software management techniques can impact your ability to install and use new function.
- The increasing complexity of software configurations. With a growth in multiprocessing and multiple processor complexes, the software environment becomes very complex and its management becomes a problem. Changes to one processor may impact all others in the installation. In addition, complex environments are no longer the domain of only the very

large enterprise. An enterprise with one ES/9000 or with S/390 Parallel Enterprise servers may have a complex environment. It may execute several operating environments in logical partitions provided by the PR/SM feature or running several sysplex images, for example.

Further, as you install and use more heterogeneous environments, such as host connected LANs, mid-range systems, and so on, you have to install and maintain more software to provide the additional connectivity.

- New tools and techniques that would simplify software management tasks are not being fully exploited.
- A simple approach to system maintenance has never been fully understood. As a result, there are:
  - System programmers dedicated to system maintenance, while they could be more productive working in areas such as performance and capacity management.
  - Unnecessary system outages and the inability to meet service level agreements.
  - Unnecessary calls to the IBM Support Center, which has the effect of increasing response time for resolution of urgent problems.

The bottom line is that to maintain your competitive edge, you have to be able to conduct your business as efficiently as possible. You have to be able to implement new and cost-effective solutions, using information systems. Those solutions must be installed quickly, simply, and without negative impact. The key is effective software management.

1.3 Software Management Defined

Software management is a management discipline that dictates how you will make changes to your systems software. Specifically, the discipline deals with the tasks shown in Figure 1.

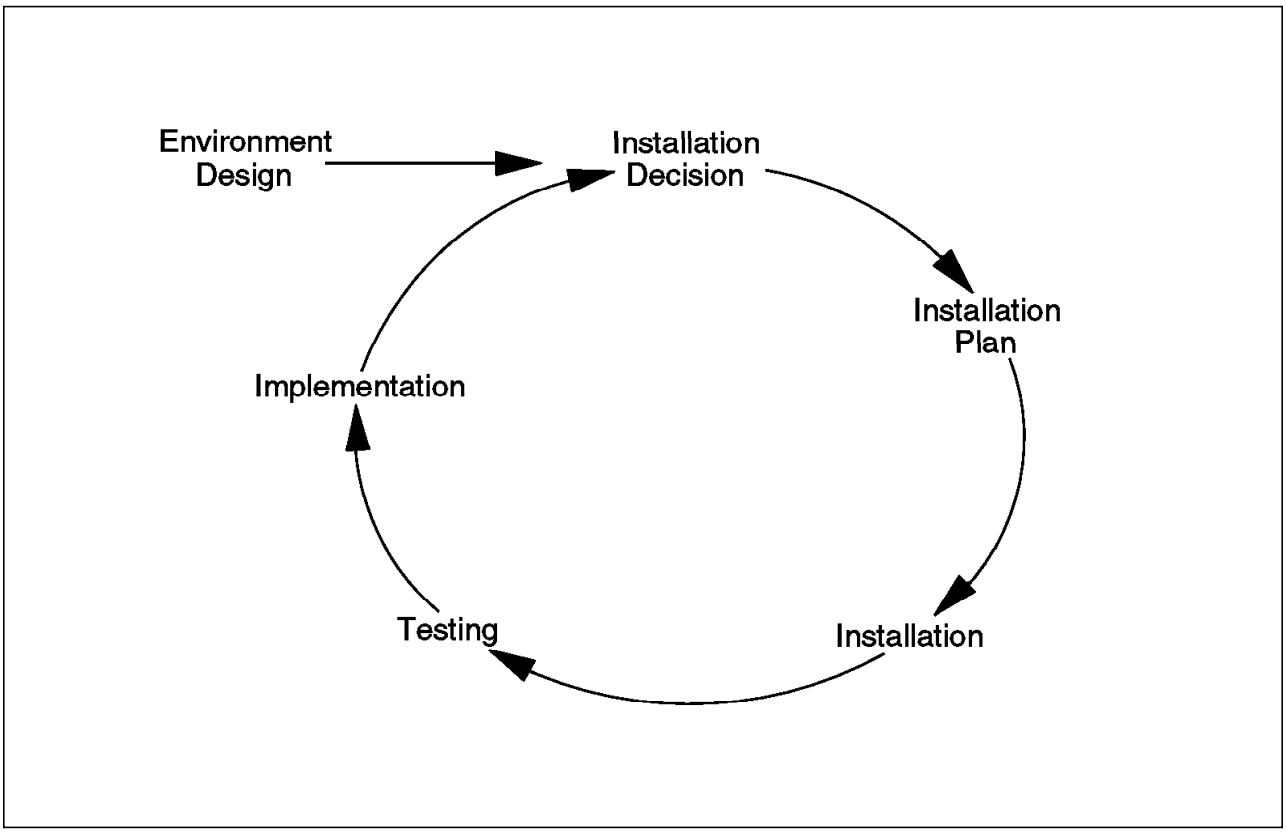


Figure 1. Software Management Tasks

These tasks, briefly defined, are:

Environment Design Design and implementation of a "maintenance environment." Here you will decide how to set up your system so that you can install OS/390 and maintain the elements within OS/390. This topic is discussed in Chapter 3, "Maintenance Environment" on page 39.

Installation Decision You make a business decision to make changes to your systems software. Such changes may be new releases of OS/390 or elements within OS/390, service upgrades, activating of additional features, preventive maintenance or perhaps installing corrective service. The process of making specific installation decisions is not discussed in this book. However, Chapter 2, "Maintenance Philosophy" on page 9 describes how the discipline of "software management" will impact how you make installation decisions.

Installation Plan Once you have decided to perform an installation, you will have to do some planning. The amount of time and effort put into planning will have a direct effect on the cost of implementing change. With OS/390, the research and planning required in the past to make things fit together properly has been reduced. Planning requires *management commitment* and good project management skills. To implement a change to the operating environment requires a plan that includes identifying key activities, who is responsible for those activities, and what

is the expected completion date. Some of the planning steps you have to perform are further discussed in Chapter 4, “Installation” on page 75.

Installation Installing IBM software via the selected distribution medium is discussed in Chapter 4, “Installation” on page 75.

Testing Determining whether the installation has been successful. Testing activities are an integral part of any change plan and must be designed and controlled. When installation plans are drafted, all too often the word “testing” is all that appears on the task list. More thought and design should be devoted to testing activities so that proper resources and schedules can be allocated. Testing is discussed in detail in Chapter 5, “Testing” on page 123.

Implementation Making the installed changes available for productive use by your customers, the users of the information system. Implementation can mean propagation of a “test” system to a “production” system. This subject is dealt with extensively in Chapter 6, “Production and Implementation” on page 139.

A separate chapter, Chapter 7, “Subsystem Topics” on page 151 discusses those software management tasks that are pertinent to OS/390 subsystems.

Once you have the right environment design for your enterprise, you can stay with that design for many years. However, you will find that software management is a cyclic process. Once you have implemented one set of products and maintenance, you will find yourself getting ready to implement a new set.

1.4 What Makes Up OS/390?

This session introduces the new shipment package of OS/390.

1.4.1 OS/390 Elements and Features

OS/390 consists of *base elements* that deliver essential operating system functions. Most of the elements are products that have been available for some time and you will have been running some of them. To distinguish the element from its prior or preceding product the term *root product* can be used. The OS/390 level of an element can be any of the following:

- A repackaging of the root product
- The root product with some additional function
- The root product unchanged

In addition to the base elements, OS/390 has *optional features* that have an affinity to the base. There are two types of feature. One type is shipped with OS/390 whether you order the feature or not. These features support *dynamic enablement* . If you order the feature they are shipped *enabled*, otherwise they are shipped *disabled*. If you later want to use a disabled feature, let IBM know and enable it dynamically through PARMLIB. The second type of feature does not support dynamic enablement and is not shipped with OS/390 unless you specifically order it in addition to the base.

IBM′s direction is to have all optional priced features capable of being dynamically enabled and disabled and hence shipped with the base.

1.4.2 Exclusive and Non-Exclusive Elements and Features

Some elements and features contain new function that is available only within OS/390. Such an element or feature is termed *exclusive*.

Other elements exist within OS/390 and also as separately orderable products. New function can be obtained within OS/390 or by ordering the separate product. Such elements are termed *non-exclusive*.

IBM′s direction is to make functional enhancements only within OS/390. Hence the number of exclusive elements and features will increase.

1.5 OS/390 Software Delivery Offerings

The increasing complexity of software has led to a strong demand for customized software services. The new standard of customized packages or CustomPacs offers solutions to this problem by assisting the customer in the management of his software environment in a less complex and time consuming way.

The CustomPac dialogs are used to install all of the software delivery offerings. The CustomPac dialogs contain code and interactive system productivity facility (ISPF) dialog elements that enable you to install CustomPac systems and subsystems.

IBM offers the following methods for installing OS/390:

- ServerPac for OS/390
- CBPDO
- IBM fee-based customized packages

A summarized description of the above offerings follows. For detailed information on each of these offerings, see 2.5.1, “Replacement Methods” on page 23 and section 2.5.3, “Upgrade Methods” on page 26. In addition, IBM provides a number of electronic facilities to aid in software management and maintenance. These facilities are described in Appendix A, “Electronic Software Support Offerings” on page 191.

1.5.1 ServerPac for OS/390

The ServerPac is the recommended system replace vehicle and comes free with your OS/390 license. ServerPac for OS/390 provides selection of OS/390 base elements and optional features as well as other products and features. It is delivered in a dump-by-data set format and provides a system that is IPL and IVP tested prior to shipment. Subsystems have already been tested during the OS/390 Systems Integration Test. ServerPac can be used to install an OS/390 system for the first time or to replace an existing MVS system. ServerPac can be ordered separately for subsystems such as CICS, IMS, DB2 or NCP.

1.5.2 CBPDO

The Custom Built product Delivery Offering(CBPDO) updates an existing system instead of replacing it. The CBPDO offering comes free with your OS/390 license. This software package consists of a logically stacked SMP/E RELFILE tape that contains the base elements of OS/390, the optional features ordered by the customer and the optional features that can dynamically be enabled.

1.5.3 IBM Fee-Based Customized Packages

Depending on your country, you can purchase one or more of the following customized packages either in addition to the above offerings or separately. All deliverables are individually tailored using customer supplied data, and delivered on the IBM recommended service level.

- **IBM SystemPac/MVS(SystemPac).** This software package consists of installed products for a ready-to-IPL system. Some of the products have been customized in response to information provided to IBM. SystemPac can be used to install an MVS system for the first time or to replace an existing MVS system.
- **IBM ExpressPac/MVS(ExpressPac).** This software package consists of installed products for a ready-to-IPL system. Some of the products have been customized in response to information provided to IBM. The ExpressPac package is the same as a SystemPac.
- **IBM FunctionPac/MVS(FuntionPac).** This software package consists of a predefined group of installed products that provide a specific software function or application. (A FunctionPac may contain non-IBM products as well as IBM products.) Some of the products have been customized in response to information provided to IBM.
- **ProductPac/MVS (ProductPac).** This software package upgrades the product and service content of an existing system or subsystem, based on the contents of the SMP/E zones to be upgraded.
- **ProductPac/E (ProductPac/E).** This software package upgrades the product and service content of an existing system or subsystem, based on the contents of the SMP/E zones to be upgraded. This package is the same as a ProductPac but all SMP/E processing is done at the production center.
- **ServicePac/MVS (ServicePac).** This software package upgrades only the service content of an existing system or subsystem, based on the contents of the SMP/E zones to be upgraded.
- **Selective Follow-On-Service (SFS).** This service is provided to all SystemPac, ProductPac, FunctionPac, and ServicePac customers. IBM will deliver customized PTF service to correct High-Impact or Pervasive APARs (HIPER PTFs), and PTFs to correct PTFs in Error (PE PTFs). SFS service packages are provided at 30- and 60-day intervals after delivery of each SystemPac, ProductPac, or ServicePac.

1.6 Reader′s Guide

When it comes to planning the installation of a system, a new software product, or just a few PTFs, there is no shortage of information to help you with the task. Reading all the information can be a very time-consuming task! Also, it can be a little difficult to figure out just what pieces of information you ought to read.

Figure 2 describes how this book fits in with all the other information that is available to assist you.

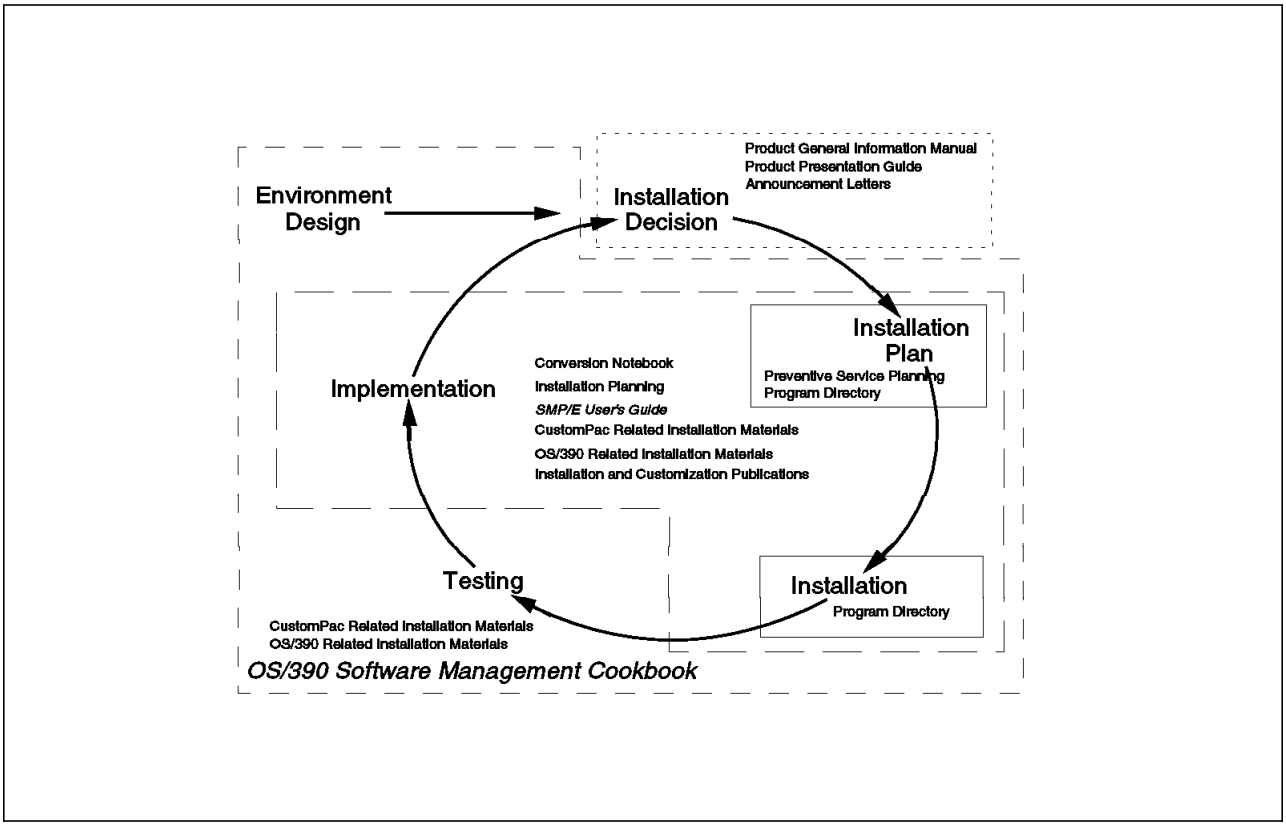


Figure 2. Software Management Information

Each of the key tasks of software management, as described in section 1.3, "Software Management Defined" on page 2 are shown, along with the key sources of information for each task.

This book deals with the concepts and overall discipline of software management. Once you have an understanding of the concepts presented here, you can move on to the specific information listed in Figure 2.

1.7 Some Limitations of This Book

Like a cookbook, this publication should not be treated as "the only way to do things." Just as you might modify a recipe to suit your personal tastes, so you will have to choose a software management approach that best suits the operations of your enterprise. The intent of this publication is to give you some ideas about how to proceed.

Use the cookbook in the way it was intended, using a dash of salt to suit your taste.

1.8 Assumptions

The following assumptions have been made:

- MVS/ESA SP 4.3 is the last release that supports MVSCP. This means that you have to use an IODF to IPL your OS/390 system. IBM assumes that you have already migrated your operating system to use HCD. Refer to *MVS/ESA SP V5 HCD: User's Guide* for migration from MVSCP and IOCP to HCD.
- With OS/390, IBM assumes that customers are already using a restructured RACF data base which implies that RACF 1.9 or higher is already installed.
- The examples in the book are based on the assumption that customers are running OS/390 in a Parallel Sysplex environment. However, there is no reason why the solutions discussed cannot be implemented in any OS/390 environment.

1.9 Read On

This book addresses the issues raised in section 1.2, “Issues” on page 1, and presents a software management methodology that makes change easier. Read on.

Chapter 2. Maintenance Philosophy

In this chapter important concepts of software management are discussed, including:

- Why manage software?
- How current should I be?
- An approach for keeping current.
- Installation strategy.
- Current package formats.
- Implementation strategy.
- Concurrent maintenance concepts.

The concept of “minimum risk” is examined, concentrating on how best to implement it in your enterprise. This discussion will help you determine which is the most appropriate software management strategy for your business.

Figure 3 gives an overview of which areas are discussed in this chapter with respect to the overall software management cycle introduced in 1.3, “Software Management Defined” on page 2. Since the discussion in this chapter covers all aspects of software management, none of the areas are discussed in great detail. However, the ideas presented in this section will assist you in forming your own software management strategy. That strategy will impact how you undertake the individual tasks, which are further outlined in the remaining chapters of this book.

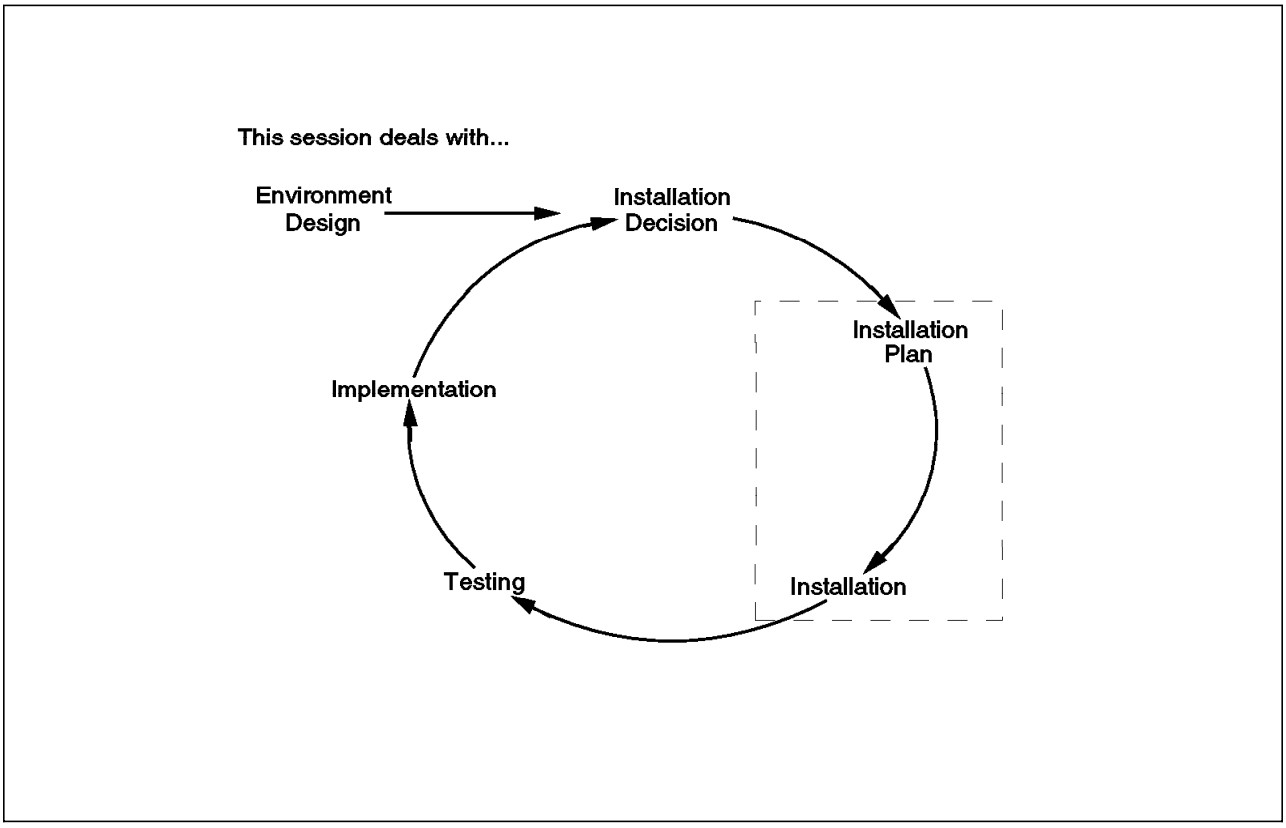


Figure 3. Maintenance Philosophy

2.1 Why Manage Software?

Software management deals with how you implement changes to your system software.

2.1.1 Management Cycle

There are several types of changes that must be considered, as shown in Figure 4. As you are no doubt aware, this is a continuous cycle of events. Your position within this cycle, at any particular time, is dependent upon two major factors;

- The last software management activity performed
- The software management strategy of your enterprise

Software management strategy is further discussed in section 2.3, “Approach for Keeping Current” on page 17.

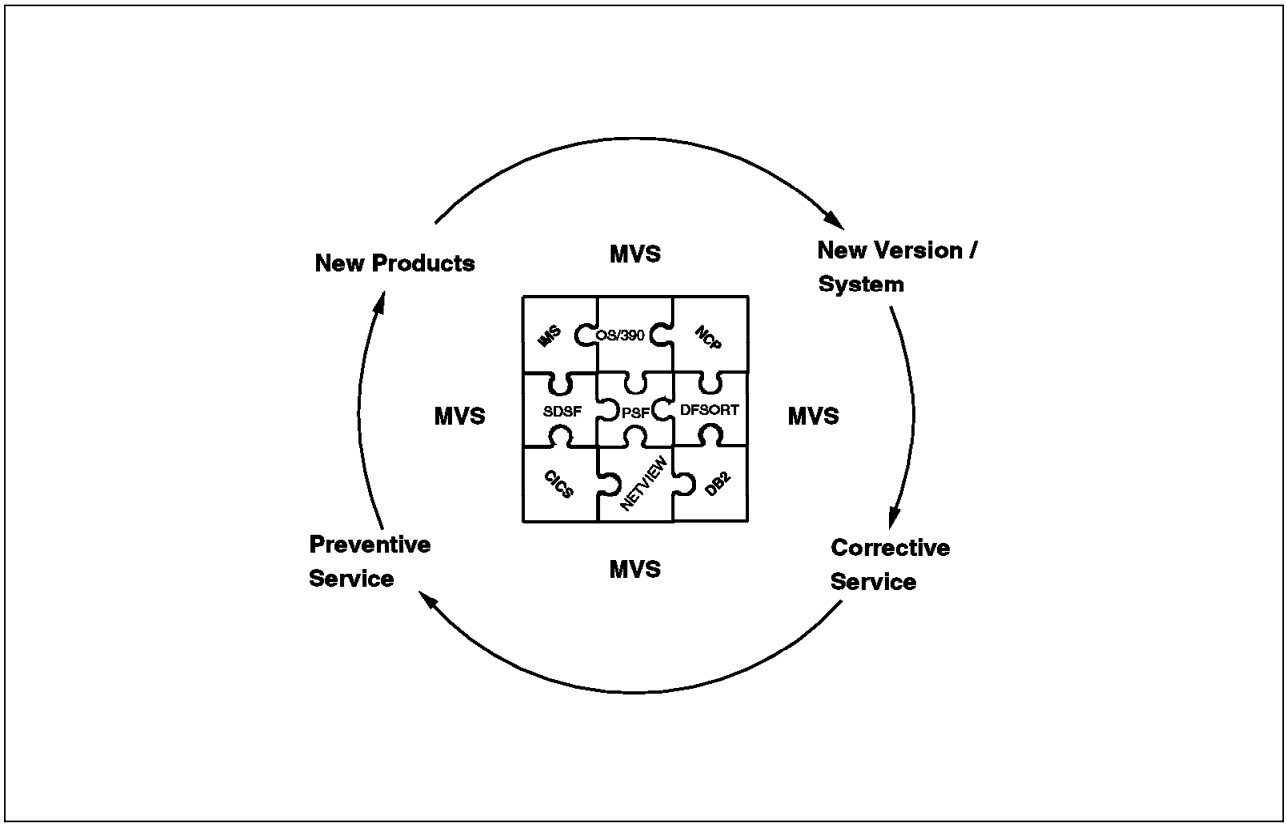


Figure 4. OS/390 Software Management Cycle

2.1.2 The Need for Change

Implementing changes, whether they be hardware or software, to an operating environment is often considered a “risky business.” Generally, the thought process behind this is that there is high business cost associated with “change.” That is, “implementing no change means no interruptions to the information system, so the business cost of using I/S remains stable.” The concept that “not changing keeps costs down” is very misleading.

There are many reasons why changes are required in your environment. Let us look at some of those reasons:

- New function
- Hardware support
- Maintenance vs replacement
- Central support

As the use of computer systems has helped your business keep or gain a competitive edge, you no doubt appreciate that it is necessary to implement new function in order to maintain this position. This need is most evident during periods of business growth or change in the business environment. The implementation of new function often has a major impact on the rest of the system, both hardware and software. If possible, this impact must be minimized. For this to happen, your systems have to be at a current level of maintenance and product version and release. This saves you from having to install new levels of products, or do large amounts of maintenance to your environment in order to support a new function.

For example, new hardware is enabled in new releases of operating system programs, or (more frequently) through enhancements packaged as PTF service or Small Programming Enhancements (SPEs). If your enterprise chooses not to install this service, the new facility may not be available to your users. When you later choose to install the new function, the amount of service required to implement it will be greater.

The arguments for both maintaining and replacing your system are fairly vigorous. In the majority of environments, many external influences act upon the system that result in some element of change having to taking place. The changes may be minor, such as a new user wanting to use an ISPF panel that had not previously been used. Since this panel was never tested there is a higher risk of a problem arising. In order for you able to minimize the risk of these external influences causing a problem, regular preventive maintenance must be performed. This approach helps to reduce the amount of corrective maintenance and therefore disruptive maintenance, you have to perform.

In addition to the requirement for new function there is another reason for implementing a new version or release of a software product. This reason has been brought about by a series of quality initiatives within the IBM Software Development Laboratories. These initiatives were introduced for three primary reasons:

1. Customer satisfaction
2. Reduction in service costs
3. Process improvements

Obviously, one way for IBM to increase customer satisfaction is to improve product quality. The process improvements that are being put into place now mean that future versions and releases can only get better. In order to take advantage of these improvements, you will have to install the latest version or release of a product.

A reduction in service costs means that there will be a reduction in time and effort required to install maintenance. This reduction in time and effort is brought about by a reduction in the amount of service you have to install. This is a direct result of the improvements in the code quality. Needless to say, in order to take advantage of these improvements you will also have to be on the

latest release that has been subject to the quality initiatives in the Software Development Laboratory.

Finally, IBM is more quickly withdrawing program support for older releases and versions of software. In order to avoid being in the position where you are running your critical business applications on system software that is no longer supported, you will have to upgrade to new levels.

Now that we have seen that some element of change is inevitable, we have to adopt an approach to software management that minimizes the cost of change to your overall business. Many things make up this total cost, such as:

- Cost to the business of not implementing the change
- Cost of system programmers to implement the change
- Cost of an outage (if required) to implement the change
- Cost associated with the change itself

The development and implementation of a suitable software management strategy can help to minimize this total cost.

2.2 How Current Should I Be?

Whenever the subject of large systems maintenance is discussed, a number of common questions are asked. These questions include:

> "How current should my system be?"
>
> "What are the guidelines for OS/390 maintenance levels?"
>
> "Should I put preventive maintenance or fix-on-fail?"
>
> "Are there recommended service levels, and are they the same for all IBM products?"

Questions such as these indicate that some guidelines are required to assist with the management of large systems software.

As discussed in section 2.3, "Approach for Keeping Current" on page 17, it is not possible to define one maintenance approach that will work for all enterprises. The approach that is taken in this publication is to help you to define the type of enterprise you have. From that point, you can determine the most appropriate techniques and tasks for your situation.

One thing that is common to all enterprises is the risk of change. You may have heard the phrase "if it ain't broken, don't fix it." Unfortunately, that phrase has been re-interpreted in many cases, so that it is now interpreted as "don't change."

For a growing enterprise, change is inevitable. Simple mechanisms are required to manage that change through the development of a maintenance strategy and implementation of change control procedures.

2.2.1 The Risk of Not Changing

What is not immediately obvious to many enterprises is that there is a risk of *not* changing the system. A growing business faces the challenge of processing an increasing number of business transactions. There is also the challenge of doing so faster and more efficiently. This requires both new hardware technology and new software technology. An operating system environment, which manages all of this, must therefore change frequently to implement new function and allow the use of new hardware.

The risk of not changing your system is illustrated in Figure 5.

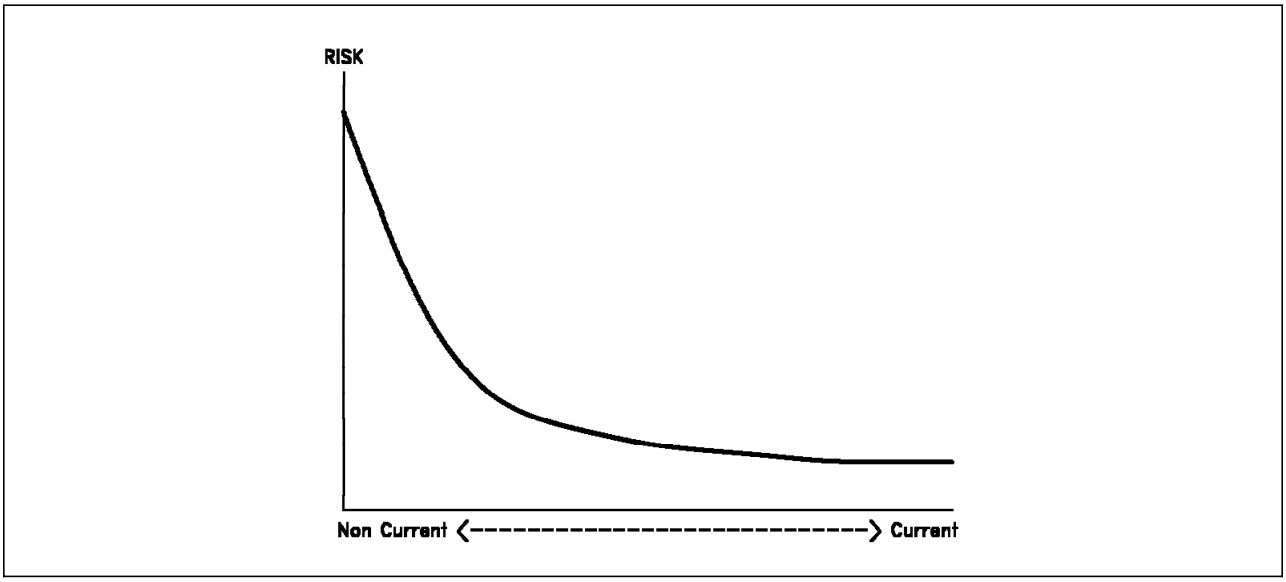


Figure 5. Risk of Staying Behind

The risks of staying non-current involve the following:

- New function is not implemented.
- New hardware cannot be implemented easily.
- New releases of programs may not function due to synchronization problems with other products.
- Problems already resolved by IBM may be rediscovered in your enterprise, resulting in unnecessary outages and wasted resources.

2.2.2 The Risk of Changing

To many enterprises, the risk of changing is intuitive, and could be defined as shown in Figure 6. The figure implies that, the more “current” the programs and service that your enterprise installs, the more likely your enterprise is to experience a failure or outage. These failures would be the result of discovering *new* problems in programs or service. One understandable reaction to this risk is not to install recently available programs and service. The rationale is that your enterprise will never discover a new problem. Other enterprises will do all the new problem discovery, diagnosis, and resolution; and then you will install programs and service, taking advantage of the knowledge gained by others.

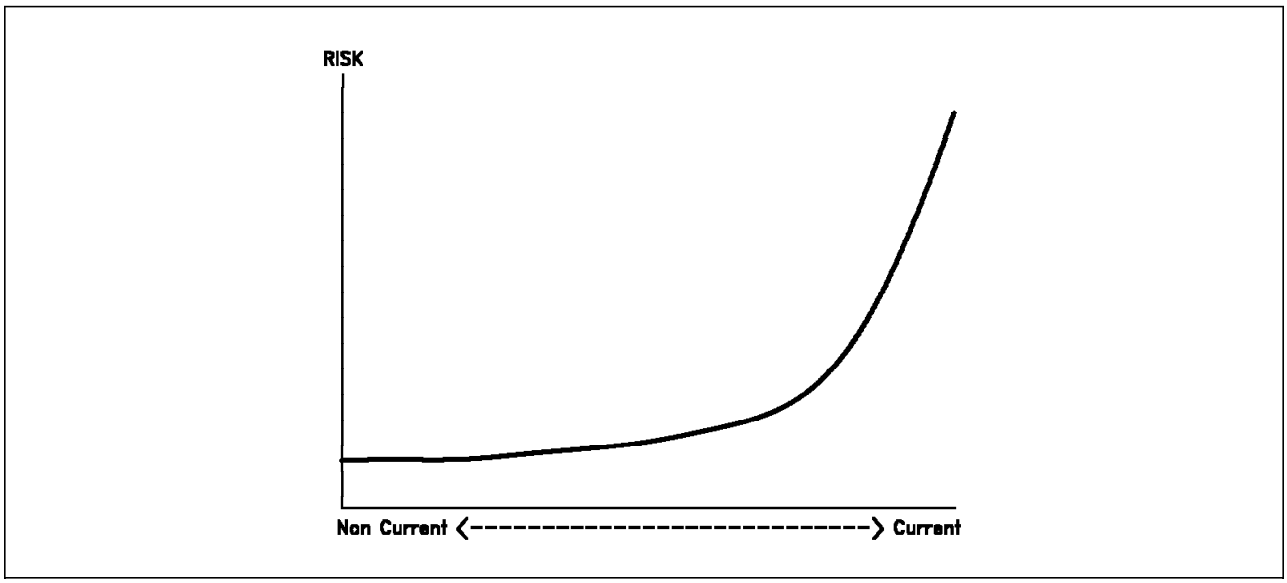


Figure 6. Risk of Staying Current

In Figure 7, the validity of the “stay behind” approach is tested. The plot shows the average time at which PTFs in Error (PEs) are discovered after their shipment. You can see that the overall trend is that fewer and fewer new PEs are discovered as time goes on. Note also that most PEs are discovered within 30 days of the PTF becoming available.

The same shape of curve is also seen in a plot of the number of problems reported for a component against time since general availability.

It may seem, then, that we are in a “no-win” situation. Staying current exposes you to the discovery of new problems, while staying behind increases the risk of rediscovering existing problems.

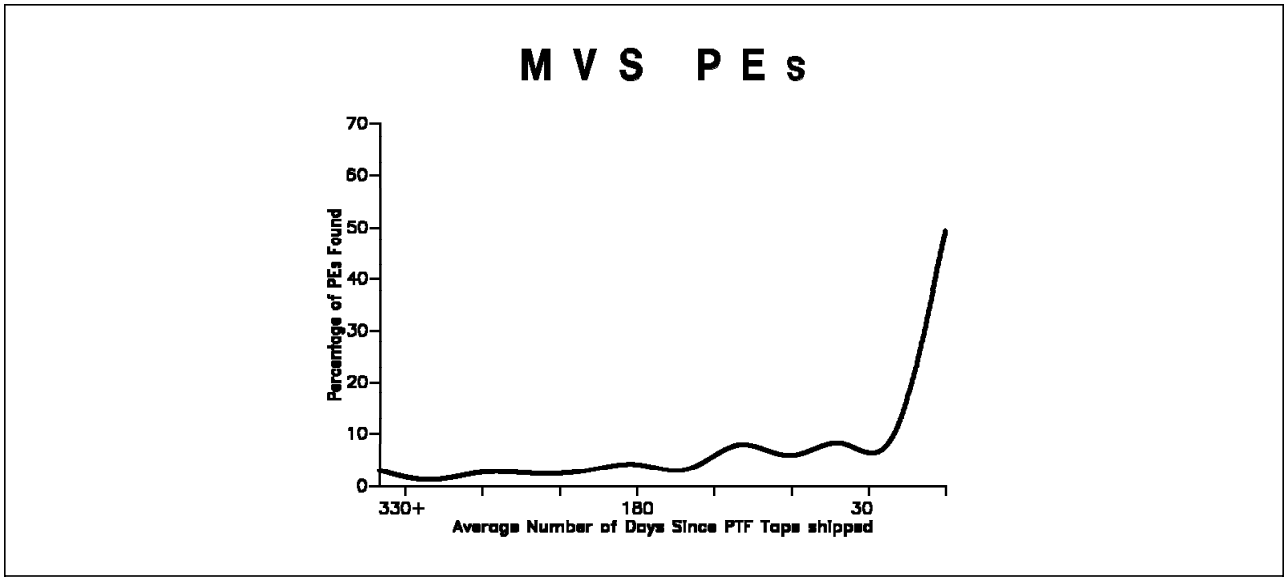


Figure 7. PE Discovery

2.2.3 The Minimum Risk Point

If “the risk of staying current” (Figure 6) is overlaid with “the risk of staying behind” (Figure 5), it is evident that there is a “middle ground” point of minimum risk. This concept is illustrated in Figure 8.

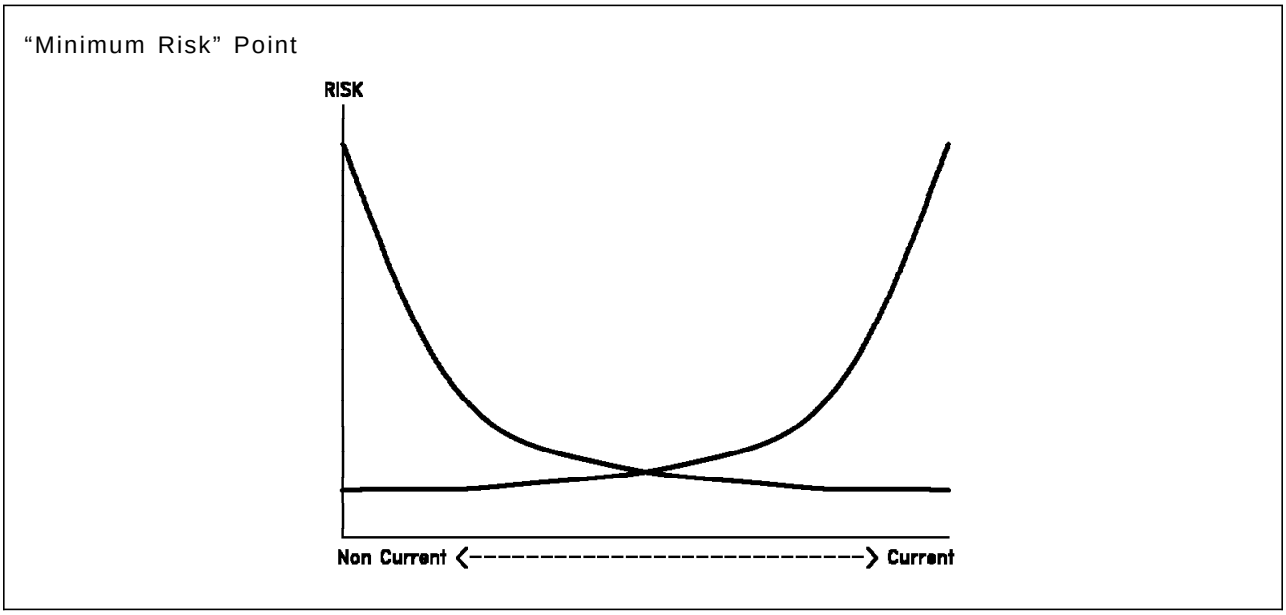


Figure 8.

So the point of minimum risk is neither very current, nor very far behind current. Rather, your enterprise should install programs and service “somewhere in the middle.” Before we define where that point might be, let us examine a few important items.

2.2.3.1 No Risk

It is not possible to eliminate the risk of outage from your environment. It *is* possible to reduce the risk to a very low point. Reducing the risk requires investments in equipment (hardware), software, people, time, and testing, among others. How much you choose to reduce this risk is a business decision. That assessment will compare the cost of the investment to the cost of the assessed risk. For example, a company that assesses the cost of an outage at 1 million dollars per hour might be willing to invest 0.99 million to reduce the risk of outage by 0.011%. This represents saving a single 1-hour outage over a year. By accepting a computer system, your enterprise has made a business decision to accept a certain risk of outage in return for dramatically increased productivity. This redbook shows you simple, low-cost ways of reducing the risks inherent in software management.

2.2.3.2 What If Everyone Stays Behind?

If every IBM user chooses to install products and service at the point of minimum risk, then both risk curves shift to the left, as shown in Figure 9.

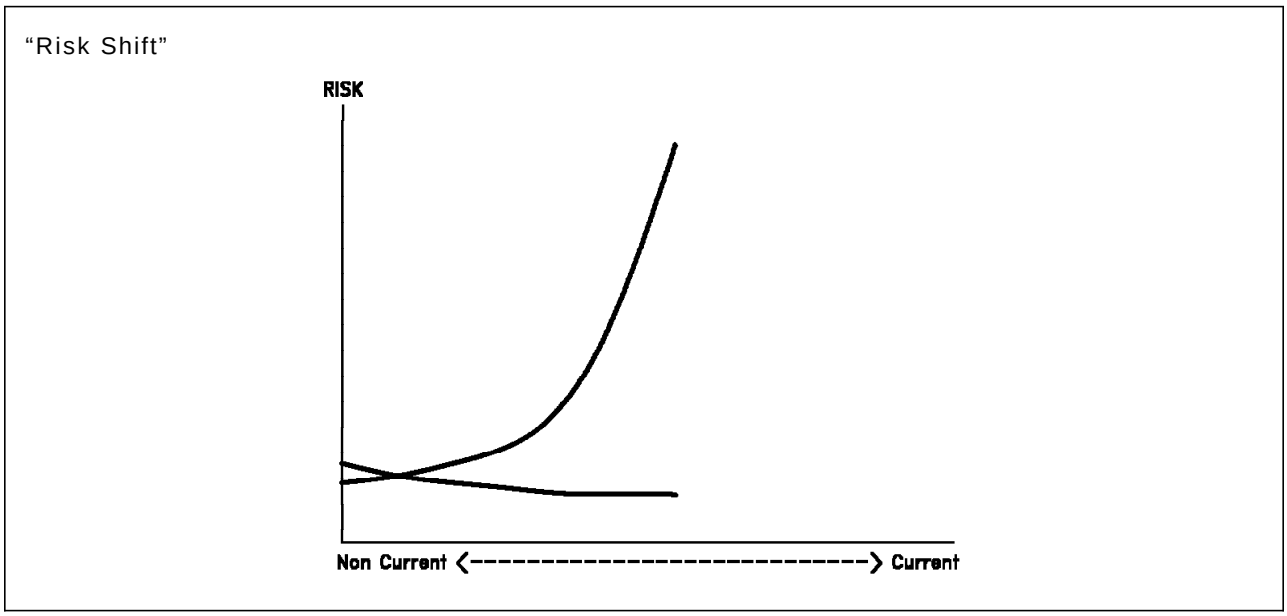


Figure 9.

This approach exposes all users to a higher risk than they expected. You should be aware that a number of processes are in place to prevent this from occurring, for example:

- OS/390 is a pre-tested system. Also the subsystems are tested during the OS/390 System Integration Test.
- New IBM programs are tested by the development organization.
- PTF service is tested by the PTF developer, by the original problem discoverer, and by I/S organizations that have agreed to do testing. IBM Software Manufacturing, which installs PTF service to maintain the ServerPac database, does some testing to ensure that a system built from the serviced distribution libraries can IPL and run Installation Verification Procedures (IVPs).
- Platform Evaluation Test (PET), a group within IBM, has a goal of providing customers with error-free software solutions. One of their key objectives is to improve IBM internal error detection. They achieve this by system-testing and production-testing maintenance. An added benefit of this testing is that they are in a position to provide a list of fixes that should not appear on regular suggested maintenance.
- Some enterprises require new function and service as early as possible. For business reasons, they accept a higher risk to implement new function quickly, and pass back to IBM any problems they encounter.

Your enterprise should assess what business requirements it has, and from that determine what to install and when. Do not delay the installation of products and service hoping that they will "improve with age."

How you decide what to install, and when, is discussed in section 2.3, "Approach for Keeping Current" on page 17.

2.2.3.3 Minimum Risk Point Determination

The minimum risk point is not simple to define. In order to calculate the minimum risk point for your enterprise, you must look at what the Information Technology (IT) requirements of your business are.

With the introduction of the CustomPacs, IBM has defined a recommended service level for you (see section 2.5, “Upgrade or Replace?” on page 21). This level has shown, in practice, that there are significant benefits to keeping at, or close to, this level. All of the CustomPac offerings are built to this IBM recommended service level.

2.3 Approach for Keeping Current

As we have already established in section 2.1.2, “The Need for Change” on page 10, change is inevitable. In the next topic we explore what is meant by “software management strategy” and look into some of the factors that will shape the strategy you adopt.

2.3.1 Today′s Strategy

Every enterprise has a software management strategy today, but the strategy in place may not be the most effective. The following discussion will help you to decide whether the strategy you use today can be improved on in your environment.

OS/390 is also introducing some new factors that can affect software management in a traditional OS/390 shop, such as:

- OS/390 plans to provide a new release approximately every six months.
- Different OS/390 releases can coexist (N, N+2) and they do not need to be installed sequentially.
- Each new release of OS/390 becomes the only available release.
- Each OS/390 release will be out of currency after three years.
- Service upgrades are integrated into each new release. Some maintenance, defined as Fix-in-next-release, will not distributed but directly integrated in the following OS/390 software.

In your enterprise you can manage these changes with different approaches and different tools. Following is a list of the more common approaches:

- ServerPac
- CBPDO
- ServerPac plus consequent CBPDO
- Leave it alone (Fix on Fail)

A frequently used policy for software management is to, on an annual basis (or so), replace the entire system with a new one built from a ServerPac or CBPDO. This method is quite acceptable on the basis of how many elements you need to upgrade in your current system. Sometime you can be involved in customization efforts even against elements that have not changed from the previous OS/390 release. This customization effort may be considerable, especially as regard the ISV products (S/390 Solution Provider). In addition, if no preventive maintenance is planned for, there is an increased likelihood of corrective service being necessary. Since corrective service, by definition, occurs after the problem has

been encountered, it can be more disruptive and costly than scheduling regular preventive maintenance upgrades into your overall strategy.

OS/390 preventive service will be delivered using the same processes that are used today. For example, Enhanced Service Offering (ESO), CBPDO and ServicePac. The new ServerPac for OS/390 delivery vehicles integrates service not only for the OS/390 Base and Communication Server, but for all orderable products on the OS/390 system replace checklist. The corrective service follows the traditional distribution procedures.

A good approach for preventive service is to use CBPDO. In a CBPDO, the service provided is at a far higher level. The benefit of this is that you can get the “average” fix applied before you experience the “average” problem. However, CBPDO delivers such up-to-date service that you might not want to put it all on, because you would be increasing the risk of your business finding a problem for the first time. The effort in determining what to put on is not all that great since SMP/E has some powerful functions to assist you, such as using SOURCEID and GROUPEXTEND. On the down side, there is a lot of manual research required to investigate all the HOLDERROR conditions encountered by APPLY CHECK. This often requires several weeks of elapsed time plus many calls to the IBM Software Support Center (ISC).

Another good approach for preventive service is using one of the packages belonging to the CustomPac offerings. Both the ServicePac and SFS solution can be helpful in doing preventive maintenance. ServicePac provides maintenance at the IBM Recommended Service Level for all the software installed on the system. The SFS package addresses specifically the installation of the HIPER maintenance and the correction of potentially PE PTFs.

One of the approaches to software management today is to leave the system alone. This really is an adoption of a “fix-on-fail,” or corrective-only strategy. In such a strategy, a PTF is applied to the system only after a high impact problem occurred. This strategy is not invalid, but is applicable only in an environment where there is a very low rate of change, and the volume of corrective maintenance is also low. This implies that the business environment in which you are working can be defined as being *static*.

2.3.2 Rate of Change

There are two major factors that will influence any decision regarding the most appropriate software management strategy to adopt, namely:

- The rate of change
- The nature of the change

If there is a high rate of change in your environment, you ought to consider an aggressive policy towards preventive maintenance in your software management strategy. Failure to do so may well result in an increase in problems found. This, in turn, causes a higher workload because of the large numbers of corrective PTFs to be applied and because you will spend additional time diagnosing each problem with the IBM Support Center. Where the rate of change is high, the nature of the change has a bearing upon the decisions you make. If the changes are such that each one is likely to exercise a previously unused piece of the code, preventive maintenance provides you the only safety net.

In such a *dynamic* environment, it is strongly recommended that you install preventive maintenance on your systems as regularly as possible, working with the resources you have available to you. It may even be cost effective to provide further investment for additional resource in order to protect the availability of your systems.

2.3.3 Dynamic Versus Static

So far we have mentioned the two extremes of static and dynamic environments. A static environment is one in which no discernible level of change exists. For this to be the case, there must be no external influences acting upon your environment. Something that may appear trivial, such as adding a new user to the system, may be sufficient to cause a problem to occur. Given the fact that you are likely to be some way behind current on service, the PTF to fix your problem will probably be available. However, because you are behind on service, you may well introduce the long-prerequisite-chain problems.

Clearly, the dynamic environment is the opposite, where change is occurring at such a rate that preventive maintenance is the only way to stop the amount of corrective maintenance from becoming too great.

Trying to decide whether your enterprise is *static* or *dynamic* is no simple matter. Most sites probably think they lie somewhere in the middle and this is probably true. So, given your knowledge of your environment, and some information on the types and rate of changes going on, you should, through careful planning, be able to estimate how often you should be upgrading the maintenance level of your software. Experience has shown that, for the majority of enterprises, upgrading the maintenance or product level of a system or subsystem from two to four times a year is not unreasonable.

You also have to consider changes to implement new function, such as new DASD or a new application. Good planning is therefore essential to be able to predict what changes are going to occur over the next, say, 12 months. The strategy you adopt has to take all these considerations into account and yet be flexible enough to take care of the unexpected.

Typically, we see that sites that have adopted good software management strategies spend less time in resolving problems with the code. This releases valuable skill for use in increasing the value of the system to the business.

2.4 Installation Strategy

Depending on the all the previous considerations and on the issues for the changes, your enterprise could decide different installation strategy paths to manage the software based on the following methods:

- System replacement
- System upgrade

One key decision, “upgrade or replace,” is examined; the discussion in 2.5, “Upgrade or Replace?” on page 21 will help you determine how to make this decision.

IBM provides a variety of packages to help you replace or upgrade the system. We describe the available OS/390 software manufacturing offerings, concentrating on the CustomPacs.

Once you have made a decision, the specific tasks to perform for replacement and upgrade, are described. Installation is outlined, using CustomPacs as an example. Due to the nature of the CustomPacs, which are customized to fit only on your system, a more general layout is presented. However, details are provided when necessary to help you understand the concept.

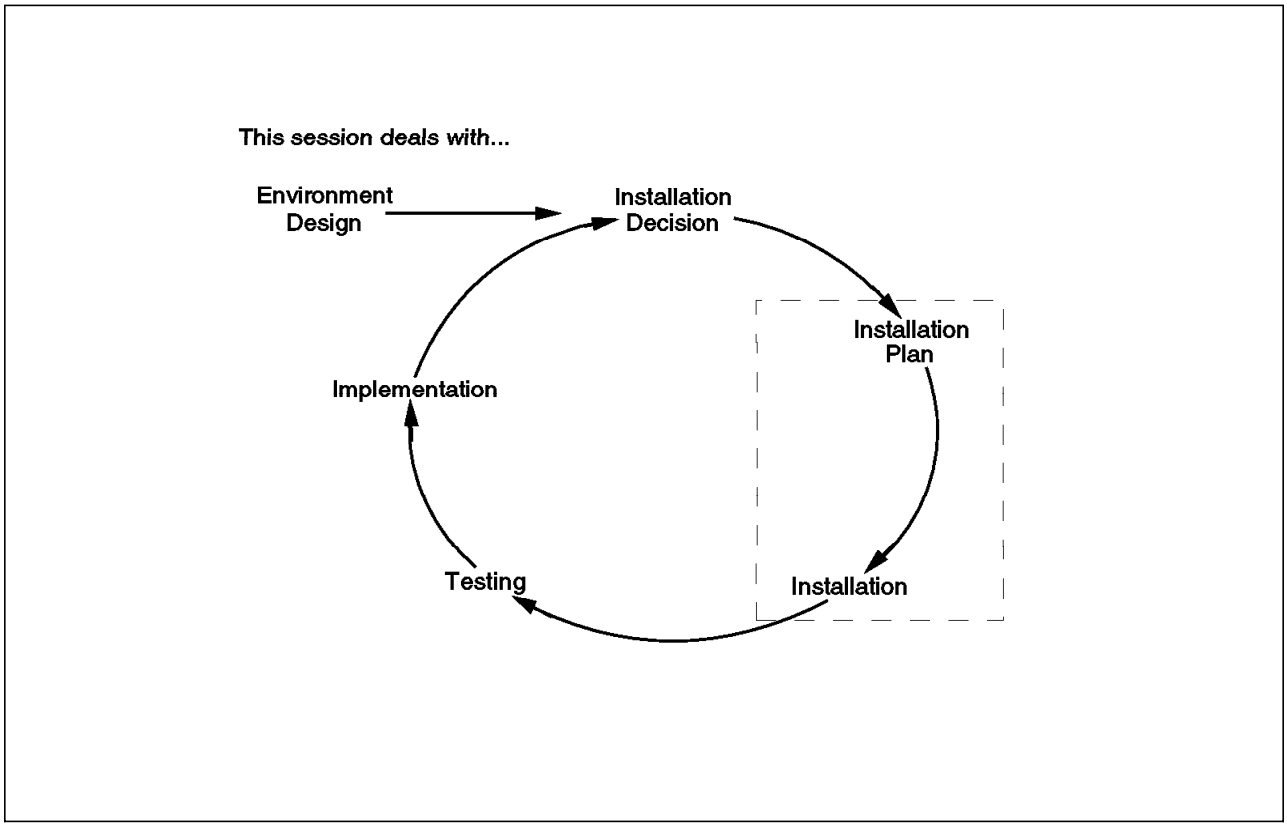


Figure 10. Installation

2.4.1 How Should I Manage My Software?

There is no single, recommended way of updating a system. All OS/390 users have different configurations, product sets, business requirements, support personnel, and so on. In other words, every user is unique, and a recommendation that would work well for one user may be unworkable for another. It is not the intention in this document to provide a single approach. It is the intention of this document to give you general guidance on the tasks that are necessary to manage your system. You will have to apply that guidance to your own unique environment.

There are some general decisions you will have to make about managing your system. You can use this paragraph to make those key decisions. Then, use it to direct you to appropriate sections of the document for guidance on the tasks required.

2.5 Upgrade or Replace?

One key question that you will have to answer when discussing installing changes on your system is: Is it quicker and less disruptive to replace the system, or to upgrade the existing software? The answer to this question depends on both the characteristics of your enterprise and the complexity of the changes being installed. When deciding whether to replace your system or upgrade it, consider the following general factors:

Service Level

Service level is simply a description of how much preventive and corrective maintenance is installed on your system. It is usually expressed as PUTyynn. For example, if your system is at PUT9601, you have installed the PTFs from the 9601 PUT tape and all previous levels. But this terminology can be confusing for the following reasons. Some PTFs from PUT9601 (and previous PUT tapes) may not be installed because they are in error, or dependent on a PTF in error. In addition, some PTFs more recent than 9601 may be installed. So your system is not actually at 9601; it is at 9601 plus some PTFs and minus some other PTFs. The point is that “9601” is not a definitive description of the system level, it is merely an approximation. The only definitive statement that can be made is for any one element, where SMP/E lists the last service to the element.

IBM Recommended Service Level

The IBM recommended service level (ISL) is based on available and tested PTFs from the Software Manufacturing Center (SMC) database. ISL includes current level PTFs, PTF in error (PE) corrections, HIPER (High Pervasive) fixes, and device support PTFs. CustomPacs are always shipped at the IBM recommended service level. Prior to integrating service into a CustomPac, care is taken to avoid installing any service that is known to be in error. “Known to be in error” means marked in error (PE) at the time the research was done for the service level. The service level date of each CustomPac is noted in the package.

To upgrade your system using CustomPacs, your service level must be within two years of current IBM service. This requirement is due to a limit in the amount of service kept in the SMC database. In fact, if your service level is more than one year behind, an upgrade will be complex and time consuming. If the system is very out-of-date, it is often quicker to replace it with a pre-serviced system than to attempt to install all outstanding maintenance.

If you are running a version of MVS that is earlier than MVS/ESA SP 4.3, IBM advises you to use the ServerPac or purchase IBM services (if available) to install OS/390.

Number of Elements Exclusive to OS/390

The more exclusive elements you use, the more likely a replace option; otherwise each element would have to be individually installed.

Number of Products or Elements to be Added or Updated

The more products or elements that are to be installed or updated, the more complex an upgrade becomes. With a larger number of products or elements to be added or updated, a replacement might be simpler, as much of the installation work is done in the

replacement package. With fewer products or elements to install, upgrading minimizes the impact on the parts of the system you are not changing.

There is no “magic number” or dividing line between upgrade and replacement. Use the number of products or elements as a factor to help you decide, along with the other characteristics listed here.

ISV (Independent Software Vendor) Software

The dependency of your environment on ISV (Independent Software Vendor) software, modifications that interface with the system and the compatibility of the ISV software with the new system, are major factors that affect your decision in upgrading or replacing. Before an upgrade or replace, make sure your level of the product supports the OS/390 environment. If possible, place the ISV software in separate sets of libraries and SMP/E zones. This implies that you will not need to reinstall it after a replacement. However, if modifications include interface with the system, you have to reinstall the modifications. An upgrade minimizes the impact on your modifications that interface with the system.

Product Complexity

Products that require many installation tasks and extensive customization work should be considered complex. The more complex the product being installed, the greater the chance of success with a replacement package. Complex products include subsystems such as CICS,IMS and DB2. The earlier the release of your software, the more complex to upgrade.

Product Availability

Within OS/390, the levels of all elements reflect the level of the OS/390 product itself. The products that make up the base of OS/390 are elements. New releases of OS/390 are scheduled approximately every six months, which imply new elements and features. Although the existing elements and features whose functions are integrated into OS/390 R1 remain available as stand-alone products at their current release levels, future functional enhancements will be shipped only as part of OS/390.

Most products, not integrated into OS/390, will be available on some of the replacement or upgrade offerings.

System Management

The sophistication of your system management disciplines is a key factor in the decision. Change Management, Problem Management, Performance Management, Capacity Management, and so on, are critical if you use the upgrade path. Because you are changing an existing system, you will need to know what the effect will be and how to recover if there is a problem. If you do not have these disciplines in place, replacing the current system will give you a known base on which to implement good systems management disciplines.

Number of System Modifications

How much you have modified the system will be a factor in your decision between upgrade or replace. If your system has many user modifications, user exits, or CSECTs that are link-edited into other load modules, it could be easier using an upgrade method than a total replacement. An upgrade minimizes the impact on your

modifications. If your system is modified, the modifications will be lost if the system is replaced.

In summary, replacements are disruptive but easy to install, while upgrades minimize disruption but require greater skill. Use the sliding scale in Table 1 to determine the most appropriate method for your enterprise. The factors that might influence your decision of whether to upgrade or replace are listed in the center of the figure. You should determine, for each factor, whether an upgrade or replacement is indicated.

The answer you arrive at will vary, depending on the factors, and most likely, some unique characteristics of your installation will also influence the decision. You may or may not arrive at the same answer each time—the effect of each factor will change, as will the number of pertinent factors. Make sure you take all of these things into account when deciding whether to upgrade or replace your system, and keep them balanced against the whole. Unless it is, by itself, overwhelming, no single factor should form the basis of your decision. Take all the factors into account, and decide which are the most important.

Table 1. Upgrade or Replace?

Upgrade		**Replace**
Current	SERVICE LEVEL	More than one year behind
Less OS/390 elements	NUMBER OF EXCLUSIVE ELEMENTS	More OS/390 elements
Less than eight products	NUMBER OF ADDITIONAL PRODUCTS	More than eight
Simple	PRODUCT COMPLEXITY	Complex
Required at GA	PRODUCT AVAILABILITY	Not Required at GA
Strong, disciplined	SYSTEM MANAGEMENT	Weak or None
Many	SYSTEM CUSTOMIZATION	Few, small system impact
Many modifications interface with system	ISV SOFTWARE	Few modifications interface with system

Having made the key decision of whether to upgrade or replace, you have to choose now between the different upgrade and replacement methods, and plan the tasks required.

2.5.1 Replacement Methods

To “replace” a system means that you have obtained a new version of the operating system ready to install. There are several ways to replace an OS/390 system. Available worldwide are packaging offerings that provide complete replacements for IBM system software:

- ServerPac
- SystemPac
- ExpressPac
- SoftwareXcel System Installation Express (SIE)

Some installation offerings are country-specific. Confirm with your marketing representative whether any other installation offerings exist in your specific country.

2.5.1.1 ServerPac

ServerPac is a system replace vehicle that comes free with your OS/390 licence. ServerPac for CICS, IMS, DB2 and NCP are also free with your OS/390 licence. Refer to Chapter 7, “Subsystem Topics” on page 151 for a discussion of subsystem replacement methods.

ServerPac is delivered in dump-by-data set format (not a physical volume dump) of distribution libraries, target libraries, and SMP/E libraries already generated. The OS/390 (including all optional features that are capable of being dynamically enabled), additional products and their service are integrated into distribution and target libraries. All optional features support dynamic enablement. If you ordered them, they are shipped enabled. See 2.5.3.5, “Optional Feature Activation” on page 28 for a detailed description of the enable/disable feature in OS/390. Other products in the OS/390 that you might have ordered come fully integrated into ServerPac.

IBM has IPLed the system and executed all installation verification programs(IVPs) prior to shipment. System integration tests are done with additional emphasis on key subsystems:

- CICS
- IMS
- DB2
- IRLM

You receive a set of dialogs that produce jobs that unload the tape to DASD. Through the dialogs, you can name the data sets and place them in the catalogs you specify. The accompanying documentation tells you how to use the dialogs. A merge-catalog tool that allows you to incorporate entries in your existing master catalog into the catalog that IBM ships, is also included in the ServerPac. All unintegrated service is available on a service tape.

IBM delivers a “packed” version of hierarchical file systems (HFS) for some parts of OpenEdition MVS Services as part of the ServerPac. You unload these data sets when you unload the distribution and target libraries during the installation. Refer to 4.4, “OpenEdition Installation” on page 105 for more information.

IBM recommends that customers choose ServerPac because of the overall ease of installation.

2.5.1.2 SystemPac

SystemPac, a fee-based CustomPac offering, includes prebuilt systems tailored to your hardware and software configuration. SystemPac are available for:

- OS/390
- NCP
- DBS (IMS & DB2)
- CICS

SystemPac delivers a complete operating system. The SystemPac build process creates TARGET, DISTRIBUTION, and SMP/E libraries. The libraries are delivered with all products and service already installed. The service is to the IBM recommended service level. Each OS/390 SystemPac is individually IPLed and IVP quality tested during production.

SystemPacs are available in two formats:

FULL DUMP format Recommended for customers new to OS/390

DUMP-BY-DATA SET format Allows a high degree of product tailoring during installation.

A SystemPac provides an OS/390 that is built to your system specifications such as I/O configuration data, volume names for target and distribution libraries, catalog names, and console addresses. Dialogs and Related Installation Materials (RIMs) are provided to help with the installation.

2.5.1.3 ExpressPac

An ExpressPac is functionally equivalent to a SystemPac. Confirm with your marketing representative which offering exists in your specific country. For a technical description see 2.5.1.2, “SystemPac” on page 24.

2.5.1.4 SoftwareXcel System Installation Express (SIE)

SoftwareXcel Installation Express (SIE) is a software offering that can help you install new software levels more rapidly in OS/390 environments. SoftwareXcel Installation Express (SIE) provides on-site planning, installation, and post-installation support services for 30 days following installation. It is a fee-based package.

SIE researches any requirements for your Independent Software Vendor (ISV) products for your new OS/390 environment. SIE provides prebuilt system packages tailored to your hardware and software configurations. The system packages are tailored to the licensed programs you select and the customization data you provide. All products and services included in the system package are integrated into a functional system at an IBM recommended level of service. Once built, the system package is IPL′d and all applicable Installation Verification Programs (IVPs) run to ensure a quality IPLable system.

Assistance of an on-site IBM representative is included in the package. Confirm with your marketing representative if this offering exists in your specific country.

2.5.2 Replacement Tasks

If you choose to replace a system or subsystem, Table 2 lists the tasks you will have to undertake, and where to look for assistance.

Table 2. Replacement Activities

	Task	See
1.	Develop a plan	Section 4.1.1, “Planning System Replacement” on page 76.
2.	Design maintenance environment	Chapter 3, “Maintenance Environment” on page 39.
3.	Install a CustomPac offering	Section 4.1.3, “Installation” on page 79.
4.	Test the new system	Chapter 5, “Testing” on page 123.
5.	Put the new system into production	Chapter 6, “Production and Implementation” on page 139.

2.5.3 Upgrade Methods

For simplicity′s sake, we will use the term “upgrade” in this topic to mean either an upgrade of elements or products or an upgrade of service only. Sometimes the installation of product upgrades may deviate from the installation of maintenance upgrades. The main difference, compared to a replacement, is that you do not start from the beginning every time. Many of the system functions remain untouched when you upgrade.

There are several ways to upgrade an OS/390 system with new products. They are:

- ProductPac/MVS
- ProductPac/E
- FunctionPac/MVS
- CBPDO
- Optional feature activation

There are several ways to upgrade maintenance on an OS/390 system. Each has some characteristics that make it appealing to different customers having different system maintenance philosophies. They are:

- Enhanced Service Offering (ESO)
- Product Tapes
- Program Update Tapes (PUT)
- ServicePac/MVS
- Service-only Custom-Built Product Delivery Offering (CBPDO)
- Selective Follow-on Service

Note: Program Update Tapes (PUTs) may have been withdrawn in some countries.

Each of these methods is appropriate in different circumstances, and at different times, as discussed in the following sections.

2.5.3.1 ProductPac/MVS

ProductPac, a fee-based CustomPac offering, provides products and service in a manner that goes beyond CBPDO and product tapes alone. A ProductPac delivers new versions or releases of software (not OS/390 elements) to be installed in an existing OS/390 system. ProductPacs are quality tested to ensure installability.

Customers supply a copy of the SMPCSI, which is used as the basis of ProductPac production. With a ProductPac a full upgrade option is available in addition to the service for the new products. This implies that service for all elements and products in the related SMP-zone will be included.

Documentation accompanying the order includes space requirements, reports, and installation instructions. Jobs and dialogs for completing the installation are also supplied. All the service research is done for you, including selection of up-to-date PTFs to fix HIPER APARs and PEs by default. Service is updated in the Software Manufacturing (SMC) database, on which ProductPac is based daily, and is immediately included in packages as appropriate.

ProductPac supplies installable products plus installable service at the IBM recommended service level. There is no need to perform product research. ProductPac supplies customized products and service with documentation that includes research.

2.5.3.2 ProductPac/E

ProductPac/E is a fee-based offering that delivers new versions or releases of software (including OS/390 elements) to be installed in an existing OS/390 system.

This offering is equivalent to a ProductPac but with the following enhancements:

- All SMP/E processing is done at the production center; therefore there is no need for any SMP/E jobs to be run at the customer site.
- Reduced installation time. It is comparable to a SystemPac.

2.5.3.3 FunctionPac/MVS

FunctionPac is a fee-based CustomPac offering consisting of either a predefined group, or customer-specified group, of installed products that provide a specific software function or application. Some of the products have been customized in response to information provided to IBM when the package was ordered. They provide benefits similar to those of SystemPacs even though FunctionPacs do not contain complete systems or subsystems.

The package contains target libraries as well as distribution libraries; there is no need to individually install the products in the package. It is delivered with all products and service already installed. Service is to the IBM recommended service level.

Customization is done by IBM in response to information supplied by customers; this up-front work reduces the need for on-site customization after the package is installed.

RIMs and documentation are also tailored to each customer as well as to the selected products; this simplifies the task of installing and customizing the package. Installation dialogs are provided to help you install and customize the FunctionPac. The dialogs can be extended to include customer-unique post-installation steps, which can then be used in processing subsequent packages.

Follow-on service is available, which simplifies the task of maintaining the package. FunctionPac is available in COPY format and allows a high degree of product tailoring during installation.

Because a FunctionPac consolidates the products needed to support a specific software function or application, it offers a broader range of products than the other custom-built offerings. FunctionPacs may include:

- SMP/E installable products and non-SMP/E installable products
- Host based products and programmable workstation based products
- IBM products and selected non-IBM products

2.5.3.4 CBPDO

The CBDPO offering come free with your OS/390 licence. CBPDO delivers installable products and service for all the products listed under your customer number for a particular System Release, or “SREL.”

CBPDO can be ordered for:

- OS/390
- CICS
- NCP

- DBS(IMS & DB2)

In your CBPDO package you receive one logically stacked SMP/E RELFILE tape that contains the base elements, the optional features you ordered and the optional features that you can dynamically enable. Depending on your customer profile, you receive uninstalled service. You also receive:

- Sample jobs to help you install OS/390 and service, including a job that builds the IFAPRD00 PARMLIB member that enables elements and features, based the customer order.
- Custom-Built Offerings Plan/Install and program materials that describe your CBPDO order.

CBPDO service is placed in one of three categories, using SMP/E ASSIGN statements:

- General service released on a PUT tape (PUTyynn)
- Recommended service (SMCREC)
- Corrective service (SMCCOR)

SMP/E HOLDDATA is supplied in conjunction with the ASSIGN statements. This allows PTF selection based on several criteria. PTFs may be selected for installation by service category; or excluded because they are in error, cause documentation changes, or require system programmer or application programmer action, to name a few. The SMC database on which CBPDO is based is updated weekly; products, service, ASSIGN statements, and HOLDDATA, when added to the database, are available in CBPDO immediately.

Usually the CBPDO offering is used to upgrade your OS/390 system. However you can use this offering also for system replacement, even though it is not the recommended way to perform system replacement.

2.5.3.5 Optional Feature Activation

The optional feature activation is a new concept within an OS/390 environment, which is described below.

OS/390 consists of base elements that deliver essential operating system functions. In addition to the base elements, OS/390 has optional features that have an affinity to the base.

There are two types of features. One type is shipped with the product whether or not you ordered the feature and the second type is not shipped with the product when you order the base, unless you specifically order it in addition to the base. Because you have ordered this feature, you can use it once you install OS/390.

The features support dynamic enablement, which allows you to dynamically enable and disable them. If you order the features, they are shipped enabled for use; otherwise they are shipped disable for use.

Once OS/390 is installed it takes only one step to enable you to access the optional functions delivered with OS/390. The optional functions can be turned on for use, based on your business need. IBM supplies a tailored IFAPRD00 PARMLIB member that defines the product enablement policy for a system. Use and enablement of the feature is subject to the OS/390 license terms and

conditions and must be done with the knowledge of your asset manager according to the terms and conditions for OS/390.

2.5.3.6 Enhanced Service Offering (ESO)

Enhanced Service Offering (ESO) is a fee-based offering that is recommended for updating existing systems by customers who are transitioning from the traditional PUT. Program Update Tapes (PUT tapes) have been withdrawn in some countries.

IBM enhanced the PUT to provide software service in a way that meets the needs expressed by PUT customers. The enhancements include:

Twelve monthly service levels. ESO offers a monthly service level.

Customer decides when to order. The traditional PUT has always been shipped to customers periodically on a schedule determined by IBM. ESO allows customers to receive their maintenance on a schedule that meets their needs.

Sourceids. The PTFs in an ESO package have the capability of being identified with multiple *Sourceids* to enhance the customer's service installation. The following *Sourceids* can be use:

- PUTyymm - Identifies the service level of the PTF.
- SMCCOR - Identifies PTFs in the service package that have corrective service but whose service cycle is not yet complete.
- HIPER - Identifies high impact PTFs.
- SPE - Identifies PTFs that are Small Program Enhancements.
- PRP - Identifies PTFs that resolve PTFs in error.

ESO on request. Obtaining service via ESO on request will give a customer maximum flexibility in controlling the content and the delivery of the ESO.

ESO via Basic subscription. The basic subscription is recommended for those customers who prefer a deliverable equivalent to the traditional PUT, since it will be shipped at predetermined intervals and will contain PTFs that have been correctively available for a minimum of 90 days.

ESO via Tailored subscription. The tailored subscription offering allows customers to tailor their subscriptions to meet their individual service needs.

The same naming convention is being used to identify ESO service levels. For example, PUT9601 will represent the service level for PTFs accumulated in January 1996. Preventative service will be available earlier with ESO. Systems Programmers can install ESOs via standard SMP/E dialogs.

The ESO package consists of software packing list and ESO tape that contains:

- File 1: ++ASSIGN statements for each PTF in the package.
- File 2: Installation and usage instructions.
- File 3: A Softcopy packing list of all PTFs shipped in file 1.
- File 4: SMP/E ++HOLD and ++RELEASE statements for ERROR holddata.
- File 5: All UCLIN applicable to PTFs included in file 1.

2.5.3.7 Product Tapes

Product tapes are available for products that are not part of OS/390 product tapes, and can be ordered when you have a product license. Each new version of a product requires a new license, but new releases of the same version may be ordered at no additional charge. Each product is packaged separately. The service is shipped on cumulative service tapes (CUM tapes) with the products, or integrated into the product by service updates. Regardless of the existence of a CUM tape, you might need more current service from another source to finish installing the products you ordered.

Product orders supply the product either by itself, or with some service that is directly related only to the product. Product tapes are usually available at General Availability (GA).

2.5.3.8 Program Update Tapes

Program Update Tapes, or “PUTs”, provide service and service information in one package for all the products listed under your customer number. PUTs package service for all SRELs together, and supply PTFs and HOLDDATA. No SMP/E ASSIGN statements are provided, but SOURCEIDs may be assigned by the system programmer when the PTFs are received using SMP/E.

PUTs are released approximately every ten weeks by IBM Software Manufacturing and Distribution, and sent to all customers of record. The service on a PUT has been tested by IBM′s Platform Evaluation Test Center. PTFs in error, and PTFs that require PTFs in error, are removed from the PUT tapes prior to shipment.

PUTs provide a single package of all the service and service information available to your entire installation. PUT is not a recommended way to maintain your software; there are more efficient offerings available.

Note: Program Update Tapes (PUTs) may have been withdrawn in some countries.

2.5.3.9 ServicePac

ServicePac, a fee-based CustomPac offering, provides a service upgrade to all FMIDs in a related SMP-Zone. All service is quality tested for installability.

A ServicePac order requires that a copy of your current SMP/E Consolidated Software Inventory (CSI) be shipped to IBM. Your CSI is used by SMC to identify the service content of the system that is to be updated. All the service research is done for you, including selection of up-to-date PTFs to fix HIPER APARs and PEs by default. Jobs for completing the installation are supplied.

Informational and action reports are provided with the ServicePac in order to complete the installation at the customer site:

- System Holds and resolved System Holds reporting all the appropriate actions to be done in order to match the maintenance system level.
- Installed service, open PE′s which could not be solved, solved PE′s, Hiper PTF′s and Ahead service not included but shipped on the service tape. You can use this report to understand what is missing, and why it is missing, from your ServicePac.
- SMP/E regression, condensed version of the SMP APPLY run identifying problems that could not be resolved at the production center. Carefully check this report and try to solve all the regression items on-site.

ServicePac is not provided by IBM if the current level of the software is older than two years.

2.5.3.10 Service-Only CBPDO

Service-only CBPDO delivers installable service for all the products listed under all the customer numbers listed for your installation for a particular System Release, or “SREL.”

Service-only CBPDOs are recommended for updating existing systems by customers who are familiar with CustomPac Installation Dialogs. Related Installation Materials (RIMs) are provided with each service-only CBPDO. Systems programmers must install service-only CBPDOs using dialogs supplied with the RIMs.

CBPDO service is placed in one of three categories, using SMP/E ASSIGN statements.

- General service released on a PUT tape (PUTyynn)
- Recommended service (SMCREC)
- Corrective service (SMCCOR)

SMP/E HOLDDATA is supplied in conjunction with the ASSIGN statements. This allows automated PTF selection based on several criteria. PTFs may be selected for installation by service category; or excluded because they are in error, cause documentation changes, or require system programmer or application programmer action, to name a few. The SMC database, on which CBPDO is based, is updated weekly; PTFs, ASSIGN statements, and HOLDDATA that are added to the database are available in CBPDO immediately.

Service-only CBPDO provides installable service, and service information pertaining to one SREL.

2.5.3.11 Selective Follow-On Service (SFS)

SFS delivers a service upgrade for all FMIDs in a related SMP-Zone, for a previously shipped order (SystemPac, ProductPac, FunctionPac or ServicePac). SFS is a follow-on maintenance package that fits on top of the previously installed packages.

During the production of CustomPacs a profile is produced that lists the FMIDs and service from both the SMPCSI, used to build the Pac, and any SYSMODs shipped as part of the Pac. This profile is used as the basis for SFS production.

SFS can be used to reduce the risk of an outage due to a missing HIPER PTF, or a PTF that is already applied but is now found to be in error. The SFS will also update the HOLDDATA on your system with the latest available information.

All service is quality tested for installability. SFS is often referred to as CRMO (Customer Refresh Monitoring Option) and will show up on most of the dialog panels rather than SFS.

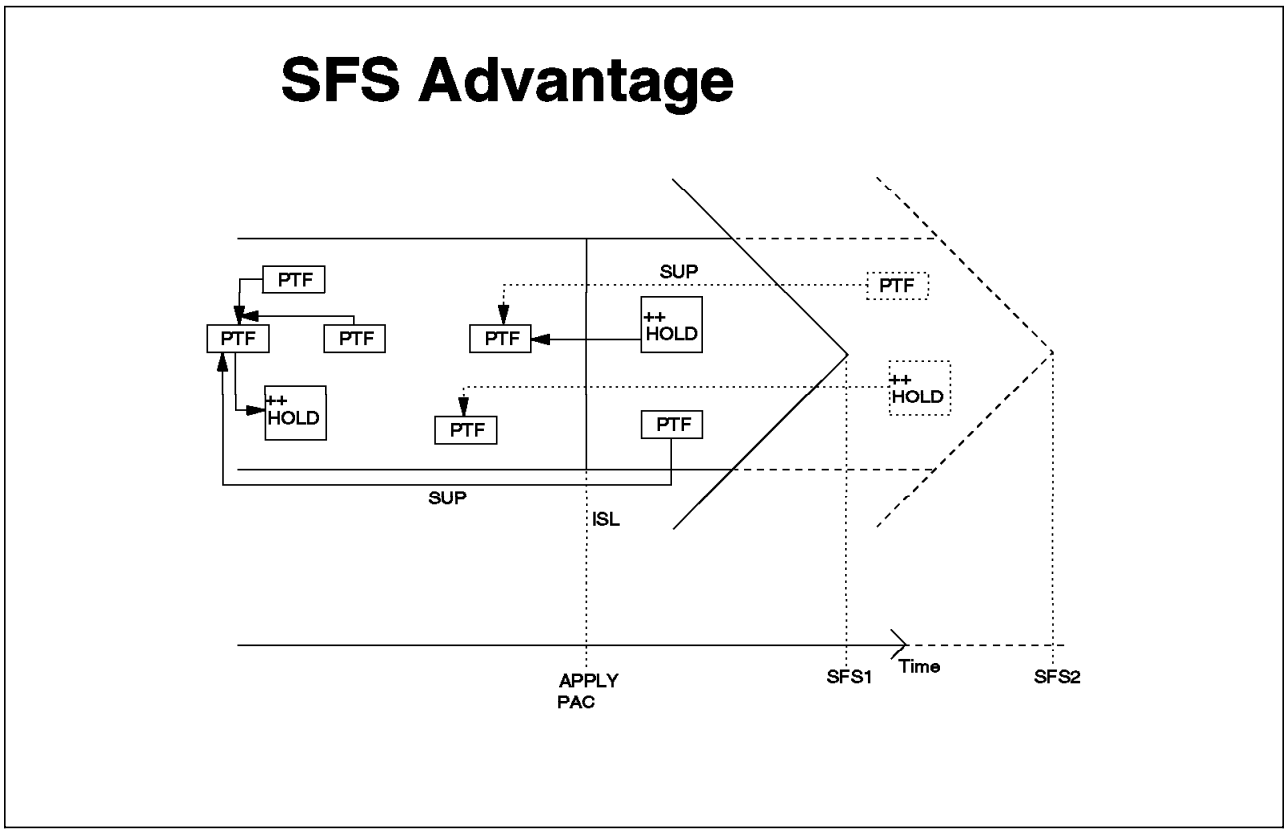


Figure 11. SFS Advantage

In Figure 11 we see that the SFS1 package will supply any fixes to known problems on your system. This may in turn allow even more maintenance, which was previously held off by the PTF in error, to go on. In addition to fixes to known problems, the holddata is refreshed. The SFS also ships any new HIPER PTFs identified since the CustomPac was installed.

Any subsequent SFS packages will do the same again. Since the original CustomPac was built against a copy of your CSI, and the SFS packages are built against the original CSI plus the contents of the CustomPac, your exposure to PE PTFs and HIPERs is minimized.

All CustomPacs contain installable maintenance at an IBM recommended service level.

2.5.4 Upgrade Tasks

If you choose to upgrade a system or subsystem, Table 3 lists the tasks you will have to undertake, and where to look for assistance.

Table 3 (Page 1 of 2). Upgrade Activities

	Task	**See**
1.	Develop a plan	Section 4.1.1, “Planning System Replacement” on page 76.
2.	Design maintenance environment	Chapter 3, “Maintenance Environment” on page 39.
3.	Install a CustomPac offering	Section 4.1.3, “Installation” on page 79.

Table 3 (Page 2 of 2). Upgrade Activities		
	Task	**See**
4.	Test the new system	Chapter 5, “Testing” on page 123.
5.	Put the new system into production	Chapter 6, “Production and Implementation” on page 139.

2.6 Implementation Strategy

As well as looking at the philosophy behind the decision to maintain software, it is also important to look at how the maintained software will eventually be implemented into the production data sets.

The traditional strategy for software implementation is that of a *physical software platform* which is contained on a single SYSRES volume with all the data sets on the SYSRES accessed via indirect cataloging *VOLUME(******) and DEVT(0000)* in the catalog entry which points to the OS/390 target libraries on the current IPL volume.

The SYSRES volume is then updated using a “cloning” process, which is described in 3.7, “Cloning an Existing System” on page 71. This technique is one of the basic building blocks of software management.

There are a number of factors that may affect this strategy, making it necessary to manage some software differently, either by choice or by necessity. Let′s look at some of these factors:

SYSRES Capacity

A physical platform is achievable as long as there is enough capacity on the SYSRES volume for all the OS/390 target libraries. As soon as the capacity of the SYSRES is exceeded and the SYSRES overflows, it is necessary to find another technique for managing the software that will no longer fit on the SYSRES.

The total space required for all the target data sets listed in the OS/390 Release 1 program directory is 2,466 cylinders on a 3380 (93% of a 3380 Model K) and 2,157 cylinders on a 3390 (65% of a 3390 Model 3). This does not leave much space for any other software to be placed on the SYSRES.

This makes the technique of indirect cataloging unavailable for all the additional SYSRES volumes. The options for managing a logical platform are investigated further in 3.3.1, “Considerations” on page 45.

Number of Implementation Cycles

It is unusual for an enterprise to implement the whole software platform at one time or in one implementation cycle. The number of separate implementation cycles will have an influence on the implementation strategy that is adopted.

How many different implementation cycles will there be? We can look at some possible scenarios:

- One for the OS/390 and associated ISVs
- One for JES
- One for the DB2 and associated ISVs

- One for the IMS and associated ISVs
- One for the CICS and associated ISVs
- One for the NCP and associated ISVs

Other combinations may include separate implementations for some of the products under the OS/390, or implementation of JES at the same time as the other OS/390 products. The permutations and combinations are almost limitless.

Having a specific physical volume for each system/subsystem will simplify the implementation of a single scenario.

Variation in Software Portfolio

If an enterprise is comprised of more than one OS/390 image, there will almost certainly be variations in the software that is in use on each of these images. Depending on the licensing agreements that are in place, it may or may not be possible to install all of the software portfolio on a single SYSRES, which is common across all the enterprises OS/390 images.

Organizational Boundaries

If different components of the software portfolio are managed by different groups within an organization, this can often result in differing time tables for the maintenance of these components making it more difficult to contain the whole software platform on a single SYSRES.

Multiple Vendors

If different components of the software portfolio are supplied by different vendors, this can also result in differing time tables for the maintenance of these components making it more difficult to contain the whole software platform on a single SYSRES.

Non-Disruptive Change

Although it is possible to provide continuous operation through non-disruptive changes to OS/390 images sharing a SYSRES volume, it can be a single point of failure.

If due to these considerations, or any others that we have not included, it is not possible or practical to adopt the strategy of a physical software platform, the alternative is referred to in this book as a *logical software platform*.

A logical software platform is defined as an environment in which the target libraries for some software components are contained on a single SYSRES volume, while others reside on additional SYSRES volumes. For example:

- On SYSRES
  - The OS/390 target libraries and associated ISVs
  - JES
- Off SYSRES
  - Elements libraries from OS/390 non activated in this configuration
  - All other ISVs
  - The DB2 SREL
  - The IMS SREL
  - The CICS SREL
  - The NCP SREL

Table 4 summarizes advantages and disadvantages of a physical platform over a logical one.

	Physical Platform	Logical Platform
SYSRES capacity	Free space	Overflowing
Number of components	Few	Many
Variation in software portfolio	Small	Large
Organizational boundaries	Few	Many
Multiple vendors	Few	Many
Non-disruptive change	A consideration	Critical

Table 4. What Sort of Platform?

2.7 Concurrent Maintenance

Concurrent maintenance is the term used to describe the process of SMP/E applying maintenance to specific elements of the system, while that system is currently active. Activation of the change will depend on where the elements affected reside, as follows:

- Updates to elements in virtual storage: This includes modules in LPALIB and the NUCLEUS. These are activated only at the next IPL.
- Updates to elements for which in-storage pointers exist: This includes changes to elements that reside in LNKLST libraries that are activated by an LLA refresh, and libraries specified in the CSVLLAxx member of PARMLIB. See the *OS/390 Initialization And Tuning Reference* for more information.
- Updates to libraries for elements that are not kept in virtual storage. Since in-storage pointers do not exist for elements such as panels and non-LNKLST program libraries, the change becomes effective the next time the program is loaded. This does not include those libraries specified in the CSVLLAxx member of PARMLIB.

Note: Before attempting any concurrent maintenance on your system, you must have a thorough understanding of the OS/390 operating system components. The relationships between the various products installed on your system should be understood completely. This includes full knowledge of where all the elements involved in the action you wish to perform reside. There is a high risk of impacting system availability when performing concurrent maintenance. Therefore, we recommend considering concurrent maintenance *only* in critical or emergency situations.

Please, use all the available tools to get an idea of what is going to be changed and which actions are required to support the changes, as for example the reports from the SMP/E Apply Check function and the ++HOLD actions documented through SMP/E.

2.7.1 Why Concurrent Maintenance?

Ensuring that maintenance actions are taken on a timely basis can help maintain performance and availability at the high levels that customer operations demand. However, because of the availability requirements of today′s online systems, customers are most reluctant to schedule disruptive activities during prime access hours. Additionally, free hours that were traditionally used for maintenance in data processing installations have virtually disappeared with the increasing requirements for 24-hour, 365-day a year availability.

Concurrent maintenance allows you to perform maintenance in parallel with normal operations, with continuous data access, but involves a reasonable amount of risk.

2.7.2 When to Consider Using Concurrent Maintenance

Questions and factors are presented below to help you determine whether a business decision should be made to take a risk and introduce concurrent maintenance.

Note: To ensure mission stability and continuity of operation, a backup of your intended concurrent maintenance target should be made before processing updates. You may encounter problems such as running out of space or going into multiple extents when updating data sets.

- Is the component vital to running the business? If the component that will be affected by the change is not vital to your business, you must consider what effect this change may have on the critical business applications.
- How severe is the problem? Unless the problem is severely affecting your critical applications, concurrent maintenance should not be your first choice.
- Not everything requires an IPL; is the situation one that could be resolved by stopping and starting an address space? For example, if the change affects only elements contained within LNKLST libraries, the change can be implemented by an LLA refresh. Or if the change lends itself to a STEPLIB, link to the change and, if it works, then process an LLA refresh.
- How disruptive will an IPL be? If an IPL can not be tolerated in this situation, implementing the change on an alternate SYSRES could be a better option.
- Will the fix hurt other users if it fails? You must consider the impact of both success and failure of the change to the other users of the system.
- Can the maintenance be backed off with minimal impact if it does not work?
- Is your environment a Parallel Sysplex? Since all cooperating images in a Parallel Sysplex are sharing the SYSRES volume, concurrent maintenance may have a critical impact from an operational point of view. For example, it might be required to refresh in-storage information on all the images and in case of failure you have to recover all the images.

  So, in a Parallel environment, concurrent maintenance is not recommended. The best way to upgrade the system software is to ripple IPL through all the images. This will allow you to upgrade the software in a short period of time without any risk for the application and a total system outage. In case of failure you can recover the failing image. Meanwhile, the applications are still running on the rest of the Parallel Sysplex.

2.7.3 Evaluate Risk Associated with Concurrent Maintenance

To help evaluate the risks associated with concurrent maintenance consider the following:

- *Never* specify COMPRESS(ALL) on the SMP/E APPLY command for concurrent maintenance fixes. For example, if modules needed by IEBCOPY are moved during a compress operation, it may not complete successfully, and the system could be left in an unusable condition.
- You cannot make simultaneous updates to modules kept in virtual storage and modules not kept in virtual storage; for example, make no concurrent updates to the NUCLEUS and LINKLIB for the same fix.
- Do not manually update in-storage modules.
- Never use GROUP or GROUPEXTEND on your SMP/E APPLYs, unless you have evaluated the fix using SMP/E APPLYCHECK. Always check the processing to verify what is actually being updated. Consider prerequisites, co-requisites, and so on.

Chapter 3. Maintenance Environment

This chapter discusses the environment you should set up to best enable you to maintain your software. Its position relative to the software management cycle is illustrated in Figure 12. The following topics will be described in more detail:

- Terms definitions
- Naming standards
- Logical design
- Shared SYSRES
- Catalog environment
- Physical design
- Cloning an existing system

Since you have to be able to make many changes and upgrades during the life of your systems, it is important that the design of your environment does not restrict your flexibility when making these changes.

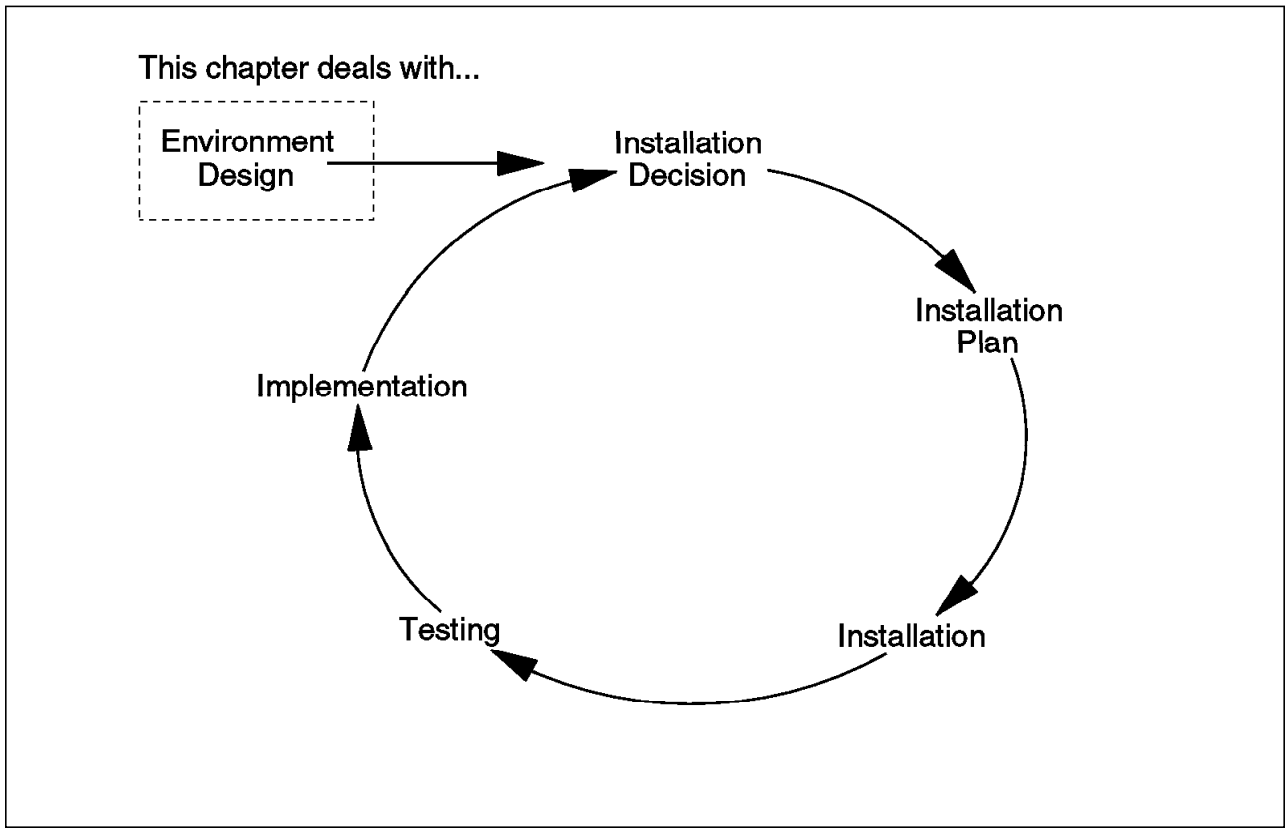


Figure 12. Maintenance Environment

3.1 Term Definitions

In this redbook, the following terms are used to describe various components of an environment. The following list is a summary of the terms that are used.

LIBRARIES References are made to four categories of libraries:

Target Libraries
All libraries containing product elements needed for product execution and managed by an SMP/E target zone.

Distribution Libraries
All libraries used by SMP/E to generate a set of target libraries and managed by an SMP/E distribution zone.

System Libraries
All libraries required by an OS/390 image that are not maintained by SMP/E. They include:

- Non-shared data sets
  - PAGE data sets
  - SMF data sets
  - SMS control data sets (in a SMS environment)
  - SVC dump data sets
  - VIO data set
  - STGINDEX data set
- Shared data sets
  - Master catalog
  - RACF primary and secondary data sets
  - JES managed data sets (SPOOL, checkpoint data sets)
  - LOGREC data sets
  - SYS1.BRODCAST
  - SYS1.PARMLIB
  - SYS1.PROCLIB
  - SYS1.UADS
  - SYSn.IPLPARM
  - HLQ.IODFxx
  - LOGREC logstream data set (in a sysplex environment)
  - Couple data sets (in a sysplex environment)

You may question the inclusion of SYS1.PARMLIB and SYS1.PROCLIB in this category. Each set of target libraries includes a version of SYS1.PARMLIB and SYS1.PROCLIB, which can be maintained by SMP/E. However, each OS/390 image should have its own active version of these libraries that will be specific to that MVS image, and should not be maintained by SMP/E.

You can continue to maintain individual LOGREC data sets for each system even if you are using the LOGREC logstream to provide a sysplex view of LOGREC data sets.

Operating System
A collection of target libraries and operational data sets.

PRODUCT SET An SMP/E target zone, distribution zone, and their associated target libraries and distribution libraries, containing products for one SREL.

OS/390 IMAGE An OS/390 image consists of the OS/390 product set and product sets for the subsystems used.

A distinction is made between an active and an inactive image.

An active image is a running OS/390 operating system consisting of OS/390, optional features and other program products. An OS/390 image provides service, and runs in one of the following:

- A dedicated processor
- A physical partition of a multiprocessor system
- A logical partition under PR/SM
- A guest under VM/ESA

An inactive image is the collection of all necessary libraries that form an OS/390 operating system, including all program products, and subsystems. The libraries reside on DASD. The inactive image can be IPLed to become active.

SUBSYSTEM IMAGE A subsystem product set and its components as an individual executable entity.

ENVIRONMENT A collection of active and inactive images. There are two types of environments:

- Operational
  The term “operational environment” will be used generically for all the OS/390 images used for production, development, and testing of applications. The operating system, including all subsystems and program products, is assumed not to be changed in this environment - we will deal with the exceptions later.
- Maintenance
  The term “maintenance environment” will be used generically to describe all the MVS images used to install and test preventive maintenance, corrective maintenance, and new program products.

SINGLE SYSTEM A single system consists of only one OS/390 image.

MULTISYSTEM A multisystem consists of more than one OS/390 image that resides in one or more processors, and programs on one image can communicate with programs on the other images.

SYSPLEX A sysplex (**Sys**tems Com**plex**) is a set of OS/390 images communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. A sysplex implementation without a coupling facility is referred to as a

base sysplex. When the implementation includes a coupling facility, it is a called a *Parallel Sysplex*.

COUPLING FACILITY A coupling facility is a special logical partition (LPAR) defined on a ES/9000 711-based machine, on a S/390 9672, on a S/390 9674, or on a ES/9000 511-based machine. Enhancements to PR/SM LPAR allow this special logical partition that runs the coupling facility control code (CFCC) to be defined. This control is IBM Licensed Internal Code (LIC). IBM's coupling technology makes high performance data sharing possible through a combination of hardware facilities and software services (starting with MVS/ESA Version 5).

ENTERPRISE The collection of all your environments.

LEVEL In this chapter, we do not use the term "level" to describe the service level of a single component in terms of "PUT level" or "IBM recommended service level (ISL)." We define it as a collective term to describe the maintenance level of all the operating system components in an image. It should help to distinguish between a higher and lower maintenance level and point to the respective environment.

In particular:

Production Operational Level
The operating system level currently used for production.

Backup Operational Level
An inactive OS/390 image. It is the operating system level that was used prior to the one that is currently used for production.

Production Maintenance Level
A copy of the production operational level used to apply and test corrective service for errors detected during production before this service is propagated to the images on the production operational level.

Upgraded Maintenance Level
The image where preventive maintenance, new products, and new releases or versions of operating system components are installed and tested. This image can have more than one set of target libraries. There could also be more than one image used for this purpose.

3.2 Naming Standards

When you first start to design your environment, it is important to define a naming standard for all its components, such as VOLSERs, data set names, member names, and aliases. A naming standard should achieve the following goals, and be adhered to rigidly:

- The name should reflect the sysplex name the component is a part of.
- The name should reflect the system name the component is a part of.
- The name should reflect what the component is.

- The names should be as simple as possible to use.

An exhaustive analysis of naming conventions is outside the scope of this document, but the following common practices provide examples of naming standards commonly used:

System Data Set Names: Some data set names must have a high-level-qualifier of SYS1. Many installations use SYS1 as a high level qualifier on all system data set names. This allows them to group the data sets for browsing, analysis, and so forth. The middle or low-level qualifier then describes the data set more specifically for example, SYS1.PAGE.COMMON.

System Data Set Names in a Shared DASD or Coupled Environment: In a shared DASD or coupled system environment, it is a good idea to use a static system symbol as a qualifier or part of a qualifier for system-unique data sets, such as page and swap, LOGREC, STGINDEX and SMF data sets. When naming data sets that must be unique to each system, for example, page data sets in a shared environment might be SYS1.&SYSNAME..PAGE.COMMON. You can also use the variables &SYSPLEX and &SYSCLONE, which represent the sysplex name and one or two character abbreviation for the system name respectively.

System and System Resource Names: Try to use the same name for any elements that have a one-to-one relationship with a system. For example, the SYSNAME of an MVS operating system should be the same as the SMF and JES name. To be consistent with SMF, the name must be four characters or less. Also, in a sysplex environment, the operator may need to type the name when routing commands to a specific system.

Volume Serial Numbers: It is a common practice to have a VOLSER scheme that uses the first three characters to define a common group, and the last three characters as a sequence number, for example, TSOnnn.

Job Names and Proc Names for System Products: Some products and components have required Job and Procedure names. Others allow the installation to tailor the names. In cases where you can tailor names, using the product′s default name often makes maintenance and cross-reference definition much easier. You can use the symbolics in JCL for started tasks and TSO/E logon procedures to perform symbolic substitution. With OS/390, you can assign job names to started tasks by using a job as the source for the started task rather than a procedure. This allows you to track multiple instances of the same started task more simply and to assign job level characteristics to control output, specifying accounting information and using SYSIN data.

3.3 Logical Design

This section describes the logical parts of an installed OS/390 image, and outlines possible designs of maintenance and operational environments.

Before installing an OS/390 image for the first time, you must think about how you will maintain your OS/390 images once they are built, and devise a design that best aids you in achieving this maintenance strategy. The design that best suits an installation greatly depends on a number of factors, such as:

- The number of OS/390 images to be supported
- The available disk space
- The number of machines available
- The method of maintenance chosen
- The frequency of planned updates
- The degree of system coupling

Depending on the particular situation, there are several other questions that must be answered before installing OS/390, including:

- Will a dedicated machine, LPAR partition, or VM machine be available for the maintenance environment?
- Will the operational environment be on a sysplex environment?
- How many levels of the operating system should be supported?
- How many catalogs do I need, and which libraries should they reference?
- How many SYSRES volumes do I need?
- How many SMP/E global zones do I want, and which target zones and distribution zones should they relate to?

When discussing system design, we distinguish between two totally separate types of environments:

- Maintenance environment
- Operational environment

There can be several types of operational environments, and they may include:

- Production operational environment
- Development operational environment
- Test operational environment

Which environments and how many of these environments are present is solely dependent on the particular needs of an installation. As mentioned before, we will not apply any service in an operational environment unless an error is encountered. This type of service is called “corrective service.”

The maintenance environment has a twofold purpose:

1. To be the central “master” copy of all the operational environments: the “production maintenance level” where corrective maintenance is installed and tested.
2. To be the place where new system levels, the upgraded maintenance levels, are developed.

   The service applied in this environment does not correct an already encountered error, but should prevent errors from occurring. Therefore, we call this type of service “preventive service.”

When the upgraded maintenance level is ready to be taken into production, it is propagated to the operational image that is to be upgraded. The former production operational level becomes the backup and the image that was used as the production maintenance level becomes available for new upgrades. For more detail on propagation, see section 6.6, “Propagating a System into Production” on page 145.

The key advantage of the OS/390 system replace vehicle, ServerPac for OS/390, is that the service is integrated for you and the target and distribution libraries and reflected in the SMP/E CSI. Preventive corrective servicing will be business as usual once you have installed OS/390. Upgrades to service levels of OS/390 or any of its program products as well as new products should be thoroughly tested in the maintenance environment. Testing efforts cannot eliminate the possibility of errors in an operational environment. Evaluate a problem in an operational environment for severity before requesting corrective service to fix it. When the situation does not demand that the service be applied to the operational environment directly, (as discussed on section 2.7, “Concurrent Maintenance” on page 35) a better alternative is to apply the service to a copy of the system in the maintenance environment, on the production maintenance level, where it can be tested and subsequently propagated to the operational environment that requires the service. Then, if any other image in the operational environment is likely to encounter the same problem, the correction can be put as preventive service.

Section 2.7, “Concurrent Maintenance” on page 35 discusses concurrent maintenance, its use, and implications.

3.3.1 Considerations

As mentioned previously, under normal circumstances, all upgrades are installed and tested in the maintenance environment prior to propagation to any operational environment. However, it is good policy to choose a design that allows the application of corrective maintenance to an operational environment, should a critical situation demand it.

To make the application of corrective maintenance to an operational environment as easy as possible, use the following list as a guideline when designing your maintenance environment:

- An SMP/E distribution zone should exist for each set of distribution libraries.
- An SMP/E target zone should exist for each set of target libraries.
- Each target zone should have an associated distribution zone. Several target zones could share the same distribution zone. Those target zones could be located on different processors, and be in different locations as long as a path exists from the processor to the DASD volumes where the target zone and the distribution zone reside. For example, a target zone in an operational environment can be associated with a distribution zone in the maintenance environment.
- The OS/390 target libraries may not fit on one SYSRES volume. Then you should consider allocating the target zone and target libraries on different volumes. The other volumes that contain the target libraries will be called SPILL volumes.
- The question of how many global zones you should have in your environment is not easily answered:
  - Having one global zone for all the images you have in your environment allows you to find out dependencies between different target zones or distribution zones, such as:
    - Dependencies between target zones of different SRELs within an image, for instance between your DB2 and CICS zones.

- Dependencies between zones of the same SREL, but in different images. A case where you would find it especially helpful would be when you are planning to run different versions of DFSMSdfp in a environment sharing (for instance) catalogs. An SMP/E REPORT CROSSZONE will tell you if you have to install toleration PTFs and if so, which ones.
- Have only one set of data sets related to the global zone (SMPPTS, logs).
- One disadvantage of this design is that if you want to propagate one of the OS/390 images, including its SMP/E environment, you will have to manually maintain the SMPPTS, making sure you do not purge any PTFs you accepted in one image but not in another one. The SMPPTS in this case holds PTFs for all the different levels that were serviced.

– Having one global zone for each level in one environment, (one for the operational maintenance level and another one for the upgraded maintenance level) allows you to cleanly separate the two images and makes maintaining your SMPPTS a lot easier. The disadvantage of this approach is that crosszone reporting is restricted to that one image.

  There is a way around this dilemma. You could have one extra global zone without its own SMPPTS and even without SMPLOG data sets. We will call this a “dummy global zone.” This dummy global zone will contain only ZONESET entries with all the respective target zones and distribution zones you would like to include for crosszone checking. You would just simply run your REPORT CROSSZONE with the boundary set to this dummy global zone and the ZONESET name that you created.

3.3.2 Maintenance Environment

The design of a maintenance environment depends on your particular needs, such as the number of levels of OS/390 you want to support, and the available resources (processors, DASD volumes).

It should be your goal to make the design as simple as possible and consequently reduce the possibility of errors. It might also prove beneficial to design the maintenance environment similar to the planned or existing operational environment, thus making propagation easier.

The simplest design can be achieved when only one level of OS/390 is used in all the operational environments, and maintenance and testing is done in a completely separate environment. It is of very little significance whether this means a separate machine, a physical partition, an LPAR partition, or a guest system under VM. For the purposes of this book, no distinction is made between the possible solutions. There are only differences when performance and the I/O configuration are considered.

Figure 13 on page 48 depicts a sample logical maintenance environment.

The design of this maintenance environment is based on the following assumptions:

- All the operational environments are at the same maintenance level, the production maintenance level.

- A master copy of the production maintenance level is kept spinning on other DASD. Corrective maintenance of the production systems can be applied and tested in the maintenance environment before it is installed on the production systems. This is assuming that the problem is not so urgent that you would attempt to solve it by applying the maintenance directly against your live production system. This is referred to as concurrent maintenance and is discussed in more detail in Section 2.7, “Concurrent Maintenance” on page 35.

  This is the image called the production maintenance level in Figure 13.

- The OS/390 image that is referred to as the Upgraded Maintenance Level is the one where all the changes are applied and tested. Note that Figure 13 is called a “sample” design, which means that what is depicted here can be much more sophisticated should your particular situation require it.

- Either of the two OS/390 images can be IPLed at the same time if dedicated logical partitions are available.

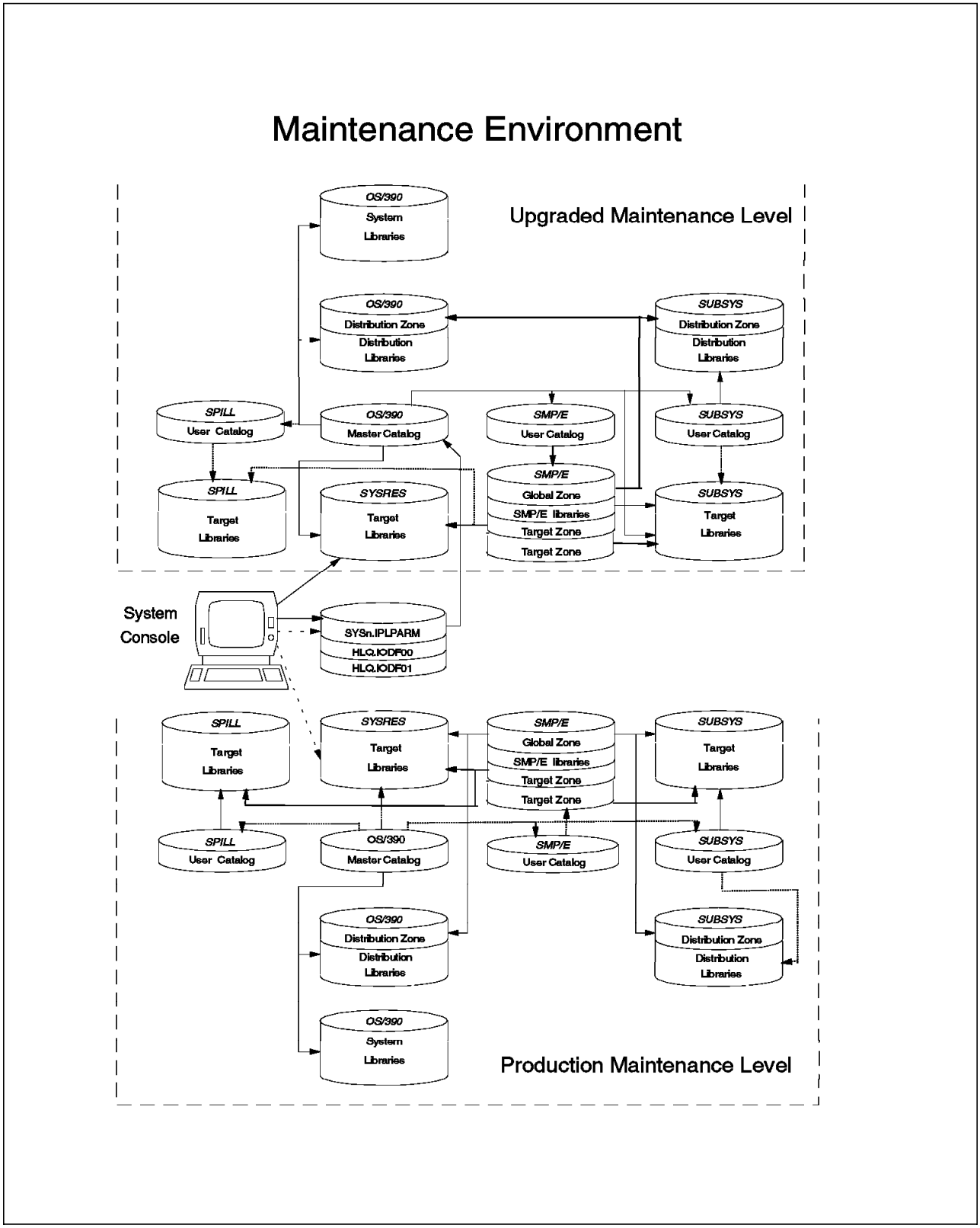


Figure 13. Sample Maintenance Environment Design

The elements in this maintenance environment are:

System Levels

The maintenance environment consists of two maintenance levels:

- The production maintenance level
- The upgraded maintenance level

These two maintenance levels are usually at different product levels, or service levels, or both. Each contains a full set of SRELs (for example OS/390, CICS, NCP, and DBS), each with its own SMP/E product set.

At any given time, one of these maintenance levels is the production copy of the system level currently running on the operational environments. It is desirable to keep this maintenance level so that corrective maintenance can be applied if necessary. At the same time, you have to be able to develop a new maintenance level at a higher service level, or product level, or both.

Having two maintenance levels allows you to fulfill both requirements by alternating between them; for example:

1. Maintenance level "A" is the current production maintenance level. "B" is the upgraded maintenance level containing product upgrades and service.
2. Upgrade all the operational environments with maintenance level "B."
3. Maintenance level "B" is now the current production maintenance level. Maintenance level "A" is now available to be used to develop the next upgrade.

The situation in a particular installation may dictate having more than one level of the target libraries on one system. One difficulty of such a configuration is ensuring that the correct library is accessed.

All the means to guarantee correct access to the libraries in such an environment entail a certain administrative effort. None of the possible solutions can be considered ideal, and it is entirely the user′s responsibility to weigh the restrictions against the effort to implement.

The most common ways of maintaining different levels of target libraries on one system are through the use of:

- Hardcoding VOLSERs and units in the target zone DDDEFs.
- DDCARDS in the SMP/E procedure.
- System-Specific Aliases (SSA). See 3.5.2, "System-Specific Aliases" on page 63.
- Indirect catalog function. See 3.5.1, "Indirect Catalog Function" on page 61.

SYSRES

Place in SYSRES only those data sets that must be there; in other words, SYSRES should contain only system software and related SMP/E-installed code. Do not use the system residence volumes for data sets that might be destroyed such as SYS1.PARMLIB, SYS1.PROCLIB, and SYS1.VTAMLST, which are tailored by each installation. There is a possibility that all target libraries cannot fit in a single volume. Then you should consider having an user catalog for the additional SYSRES to make the propagation of this volume easier.

SMP/E Global Zones

There is one global zone for each of the two maintenance levels. The reasons for this approach are:

- As mentioned before, the production maintenance level is the image that was propagated into the operational environment for production. It contains all subsystems and program products that are used on any of the images in the operational environment. It has its own global zone, which reflects the service levels of the different products used for production.

  The other image, in our maintenance environment, is the one where all the changes such as PTFs and new products will be installed. The design of this image will greatly depend on your needs.

  The design shown in Figure 13, has just one set of target libraries for each level. Service is applied directly to those data sets. This method might appear to involve a certain amount of risk. While it is basically concurrent maintenance, it is on a system reserved for this purpose. As long as there are no other activities on the system while changes are applied, it is a safe method.

  If you decide to have more than one set of target libraries, you will have to find a way of dealing with duplicate data set names. Suggestions on how to handle duplicates is described later. (See section 3.5, “Catalog Environment” on page 61 and section 3.5.2, “System-Specific Aliases” on page 63).

- When an upgrade has been installed and tested, it will be propagated into production. The upgrade now serves as the master copy of the production images. The former production maintenance level is now obsolete. It is upgraded to the new level that has been propagated into production and serves as a vehicle for the installation of new upgrades. Before new upgrades are installed, it might prove useful to accept all the applied changes and purge the SMPPTS of all the accepted PTFs. To ensure that you can remove changes from your production maintenance level, you would not accept the latest PTFs in this image. Having a separate global zone for each of the two images will probably make it easier to maintain.

- Keep your ISV (Independent Software Vendor) products in a separate SMP/E Global, Target and DLIB zones. If possible, place the products in their own sets of libraries. OS/390 R2 provides an SMP/E enhancement that will automate the process of copying SMP/E supported ISV products FMIDs from your old system (zone libraries including services) to the new system. However, those ISV products that are not SMP/E-installable and are imbedded in system libraries will have to be re-installed manually. This enhancement will be discussed in more detail in Appendix C, “BUILDMCS Command” on page 199.

- The global zone is accessed through a user catalog. Designing the maintenance environment in this way will make propagation easier.

- The two maintenance levels kept in the maintenance environment are completely different in the service applied and possibly also differ in the installed products. Therefore, it seems reasonable to separate their SMP/E environments.

Master Catalogs

Each of the two maintenance environments has its own master catalog.

3.3.3 Operational Environment

Figure 14 outlines a logical operational environment that consists of the following elements:

- A production operational level running in a sysplex (System A, B and C)
- System non-shared libraries
- Shared SYSRES
- SPILL volume
- Shared master catalog
- A backup operational level including the corresponding SMP/E environment
- SMP/E global zone and distribution zone for OS/390 and the subsystems

As long as the considerations will be made for a sysplex, the same can be done to a single system environment.

An operational environment differs from a maintenance environment in many ways. In a maintenance environment, one of the key points might be to use DASD space sparingly, while performance and availability do not seem to be of primary importance. The situation in an operational environment is just the opposite. Availability and performance are certainly at the top of the list. Therefore, it should be stressed once more: Concurrent maintenance in an operational environment must be done with due care.

The backup operational level is kept to provide the possibility of returning to a “proven” service level in case of grave difficulties. No SMP/E activities are to be expected for the backup level. Still, if the available resources allow it, keep the SMP/E environment for this level.

The main difference between the operational environment and the maintenance environment is that the images do not need their own global zone. There is one more image that is on the same level as your production systems; namely the production maintenance level in your maintenance environment. This means that you could effectively share the global zone, distribution zone and distribution libraries between the image representing your production maintenance level and all your production images. However you need to have a target zone for each environment to apply corrective service in a emergency situation.

In a multisystem environment, try to share SYSRES, master catalog, and system-type libraries (such as IODF, SYS1.DAE, and RACF data bases). Use symbolic substitution to reduce the number of PARMLIB and PROCLIB members that are unique to specific systems. For more information see *OS/390 Implementation Guide*.

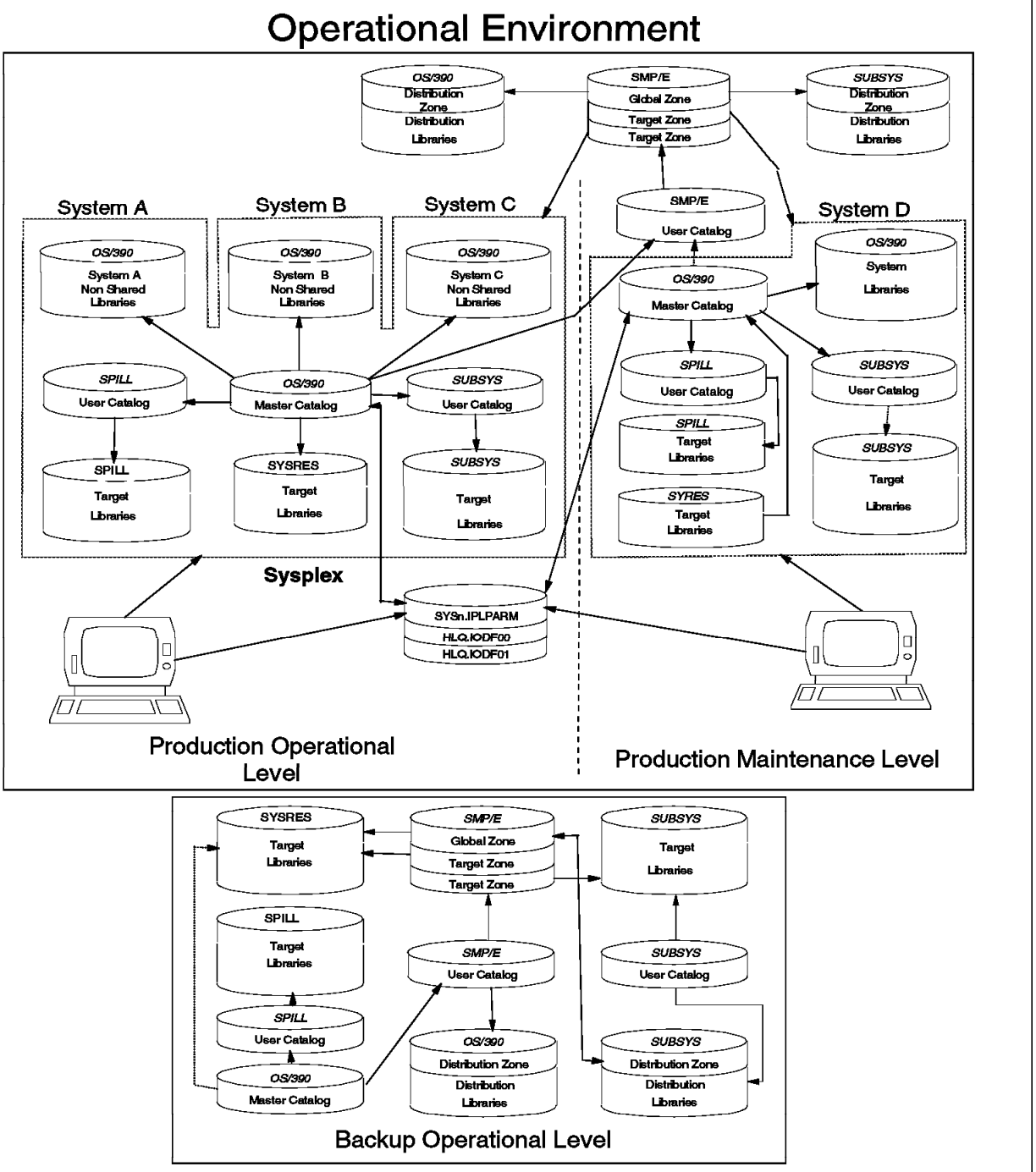


Figure 14. Operational Environment Design

The elements in this production operational environment are:

System Levels

The operational environment consists of two operational levels:

- The production operational level
- The backup operational level

Both these operational levels contain a full set of target libraries propagated from the maintenance environment (usually more than one physical DASD volume, the system residence volumes), the associated target zone, and a user catalog containing entries for some of the target libraries. As suggested in Figure 14, the backup operational level keeps a set of distribution zone and distribution libraries, while the production operational level shares distribution zone and distribution libraries with the production maintenance level in the maintenance environment. These two operational levels can contain different product levels, service levels, or both.

Environmental changes, new service levels, and new product levels will be introduced by propagation from the maintenance environment, where those changes have been installed and tested. For propagation of new system levels, you can think about *rippling* the changes through the sysplex. For more details see 6.6, “Propagating a System into Production” on page 145.

Most of the time the operational environment is stable and unchanging. It is sensible to have the backup operational level and the production operational level the same, thus providing an online backup of the current system level in case of problems.

System Libraries

System libraries are made up of a complete set of operational data sets. As there is only one operational level active at a given time, some of the operational data sets such as SYS1.DUMPxx can be used by either operational level; others, such as page data sets and VSAM clusters, have to be allocated separately.

See 3.1, “Term Definitions” on page 40 for a list of operational data sets.

3.4 Sharing System Residence Volumes

Today′s DASD is typified by high performance and outstanding reliability. This, coupled with the ability to use more than one I/O definition with any particular system residence (SYSRES) volume, has made sharing SYSRES volumes a valuable software management tool.

3.4.1 What Is a Shared SYSRES?

A shared SYSRES is one from which more than one system image is IPLed at a time. It contains the target libraries for OS/390 and its related products. (Although it will be discussed here as though it were a single volume, more than one volume might be needed to contain your target libraries as considered in Figure 14 on page 52.)

The number of images that can share a SYSRES is limited predominately by the number of paths to the DASD device on which it resides. Performance is of ever-decreasing concern, because I/O to the SYSRES has been significantly reduced by enhancements made to MVS, starting with MVS/XA Version 2. If I/O

to the SYSRES rises to the point where device response time affects the performance of the systems sharing it, you can manage the situation by placing the SYSRES behind cache, by reevaluating the mix and placement of data sets on the SYSRES, or (as a last resort) reducing the number of systems sharing the SYSRES.

Experience with 4-way, large-system complexes suggests that a SYSRES would have to be shared by a much larger number of systems for the SYSRES device response time to become a significant factor in system performance. However, depending on data set placement and whether or not the SYSRES resides behind a cache controller, you might experience some contention during a simultaneous IPL of more than one image.

3.4.2 Availability Considerations for Shared-SYSRES

Before choosing to implement a shared-SYSRES in your environment, you should consider the implications for availability. When you implement a shared SYSRES, it becomes a single critical resource for all the systems that share it. In this way, it becomes a resource analogous to other single critical resources in your installation, such as the RACF database.

There are times when you will want to avoid sharing the SYSRES. For instance, you might need to limit the number of single critical resources for two or more systems.

In this and similar cases, the more important consideration is the probability of both systems failing at the same time. This probability is increased in a shared-SYSRES environment.

You might also want to avoid a shared-SYSRES environment between images that require frequent maintenance and those that do not. (Note: See 2.7, “Concurrent Maintenance” on page 35 for some things to consider when applying maintenance for one image to a shared SYSRES.)

3.4.3 Shared-SYSRES Design

The benefits you realize from a shared-SYSRES environment depend on how you choose to design your operational environments and on the applications you use. Generally, the major benefits are:

- Higher complex-wide availability
- Reduced propagation workload
- Reduced maintenance workload
- Reduced DASD usage

Some factors to consider during the design include:

- If it is not system software, put it somewhere other than on the SYSRES. This general rule alone is almost enough to get you to a functional shared SYSRES environment. PARMLIB, PROCLIB, and so forth, are not system software. Less obvious are things like VTAM mode tables used on one system but not another. Put these somewhere else and concatenate them to VTAMLIB. If IBM doesn′t supply it, put it somewhere else (preferable) or make it common (for example, IMAGELIB).
- Using indirect cataloging (DEVT(0000) and VOLUMES(******)) will simplify things considerably for backup, recovery, and migration. Also, it will help

enforce a good shared SYSRES design. See 3.5.1, “Indirect Catalog Function” on page 61 for more detailed information.

- If you need a SPILL volume, you should consider having a user catalog for the target libraries in these volumes instead of cataloging them directly in the master catalog. In this way, if you need to migrate these volumes, you only have to alter one entry in the master catalog, using EXPORT DISCONNECT/IMPORT/CONNECT (with alias option) commands.
- DASD contention is generally not a problem, but if it becomes a problem, you should cache the SYSRES. Note that one cannot add the RMF device busy numbers together; DASD busy calculations are beyond the scope of this book, but the effects of sharing a SYSRES can be calculated using various published formulas.

  Clustering LNKLST data sets around the VTOC in the center of the volume is cheap and will help if things are marginal. In this case, try to ensure that a normally-running system does not need other things on the volume too often (for example, put your SMF data sets somewhere other than on the SYSRES).
- Thought should be given early to importing a new SYSRES into the operational environment. All system software should be SMP/E-managed, but not all of it needs to be updatable at any given time. For instance, the currently IPLed volume could be renamed although it could contain VSAM CSIs. They won′t be usable until another rename because of VSAM data sets, but they will always be an accurate reflection of what is on the volume so it can be maintained later. In this case, the backup SYSRES should be in a condition SMP/E can update. If possible, do not place VSAM data sets on SYSRES volumes.

  Note: All this presumes a high-availability environment in which one system cannot be made dependent on another in case a failure occurs. If such dependencies are acceptable, you may not need an online backup for the SYSRES at all.
- Things that vary by system should be named according to a predictable convention, especially when this difference will be visible to the operators or end users.
- How one plans to implement changes will dictate how many SYSRES volumes are needed. If all images will be IPLed within a short period of time, one plus a backup are enough. If changes will be staged over time, three or more might be needed.
- Data sets that cannot be shared are identified in *OS/390 MVS System Data Set Definition*. Each system will require its own set.
- The use of GRS to eliminate RESERVEs for data sets on the SYSRES is recommended. However, since data sets on the SYSRES are generally updated only by system programmers, this risk can be managed and some installations might choose to accept it. Note that DFSMSdss will issue a RESERVE when dumping the production SYSRES.
- There should be *no* free space on the SYSRES volume. Allocate dummy data sets to make it so. This will prevent users and batch jobs from allocating temporary or permanent data sets on the SYSRES when it is the IPL volume, and eliminates one potential source of high device busy and RESERVE.
- All data sets on the SYSRES should have UACC(READ) or *lower* to prevent people other than system programmers from updating the SYSRES volume.

Figure 15 shows one logical design for a shared SYSRES and master catalog environment. In this design:

- The I/O configurations must either be common to all images, or must be selected at IPL time.
- The master catalog must be pointed to by the SYSCAT statement in the LOADxx member of PARMLIB or the SYSn.IPLPARM data set, or must be placed in the NUCLEUS data set in SYSCATxx member (the former is preferred).

The SYSRES should contain only system software. All other operational data sets, such as BRODCAST, UADS, RACF, PROCLIB, LOGREC, PARMLIB, the master catalog, and non-target SMP/E data sets, should be placed on other volumes. Data set placements and benefits relative to this environment depend strictly on how your data processing center utilizes the operating systems. Beside the benefits of availability, ease of propagating multiple system images is also a valuable gain. Figure 15 shows the logical design of a single shared SYSRES environment.

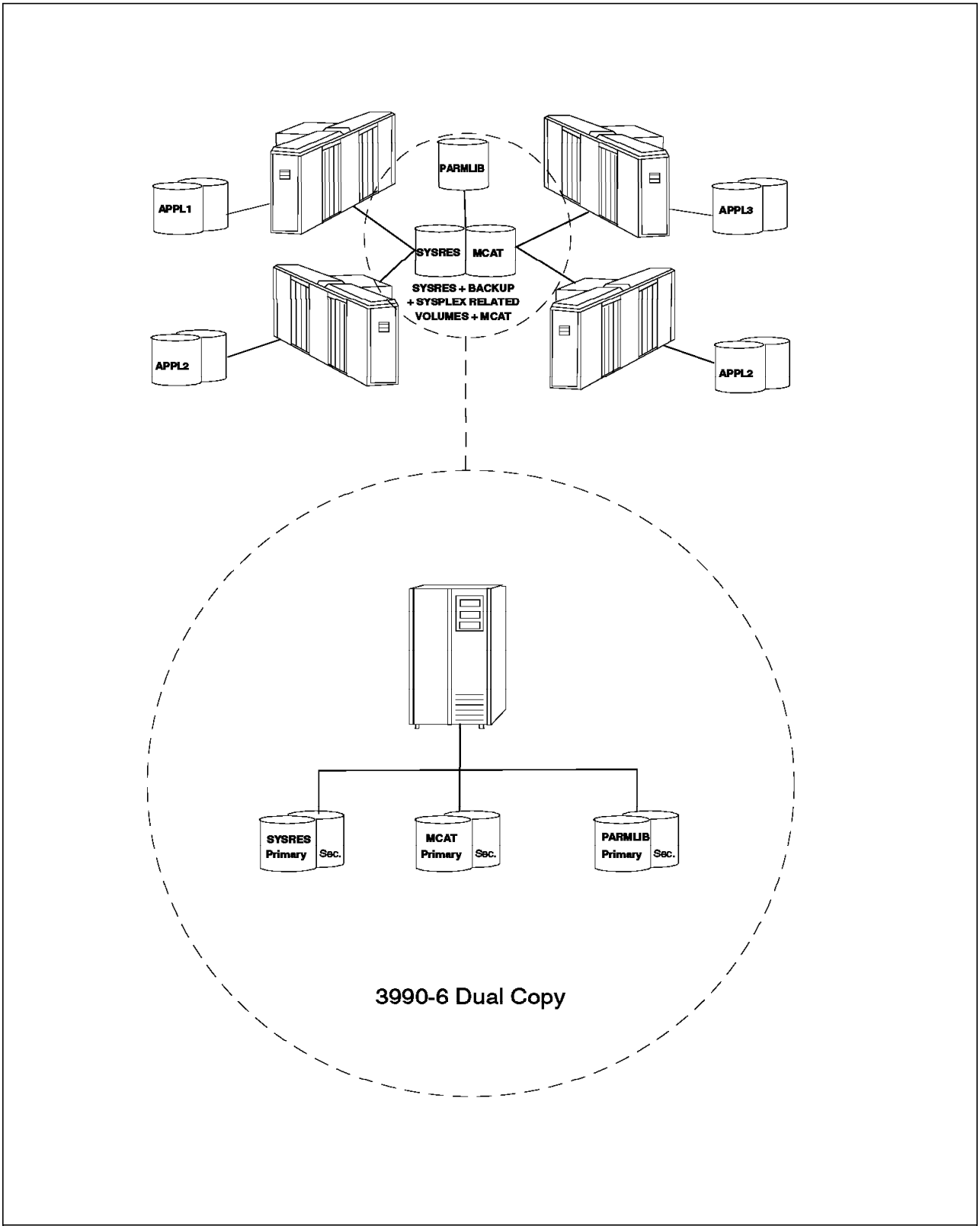


Figure 15. Shared SYSRES Image, Logical View

To implement a new SYSRES, the new or updated system is propagated onto the backup SYSRES, and when completely tested, the workload of the first processor is given to another processor (not necessarily sharing the same SYSRES). The first processor is then IPLed using the new SYSRES, and its workload is returned to it. The same then happens to the second processor within the complex, and so forth. In this way, all the processors sharing the SYSRES can be successively moved to a new SYSRES without disrupting the processing that is being performed. This will reduce the risks of concurrent maintenance discussed in Section 2.7, “Concurrent Maintenance” on page 35.

Note: In the above discussion, a new system could be a system that has had anything from an emergency PTF applied, to a system installed using a ServerPac or CBPDO, including further customization.

By taking advantage of multiple paths to devices, and using current hardware features such as dual-copy and RAID-5 technology, the potential for failure is further minimized.

3.4.4 Sharing System Parameters

The benefits provided by a shared SYSRES also apply to a shared SYS1.PARMLIB. Reduction in propagation and maintenance workload is proportional to the degree to which the environments sharing the PARMLIB are similar. If you share both SYSRES and SYS1.PARMLIB, maintenance will be easier.

PARMLIB structure allows you to maintain sets of members that apply to all systems in a shared environment, and other members that may need to be unique to a particular system. You should try to keep as many members common to all systems as possible. This reduces the need for complex sets of pointers from one member to another, and also facilitates maintenance because a change needs to be made only once for all the sharing systems. Even in cases where unique values are required by specific systems, static system symbols (defined in PARMLIB member IEASYMxx) can be used to allow a single member to be used by all systems in a sysplex. The authors believe that static system symbols should be used whenever possible to allow a single PARMLIB member to be used instead of many PARMLIB members. In cases where you must create members with different parameters for each system, you should:

1. Name the system-specific members with the same suffix (for example, all system-specific members for SYS02 should have the suffix 02).
2. Concatenate the system-specific parameters to a member containing common parameters if concatenation is allowed for that member (for example IEASYSxx and COMMNDxx).

Members that may need to be re-created or tailored using symbols for each system, are usually:

1. Members that assign names to specific resources, especially if those resource names need to be unique within a larger environment (such as a sysplex or a network).
2. Members that contain resource-allocation parameters, (for example virtual and real storage sizes, or numbers of regions). These may be different if the systems sharing PARMLIB have significantly different capacities, or are supporting different operating environments.

3. Members that contain pointers to members in the above categories need to be repeated or tailored using symbols to complete the structure.

Specific PARMLIB members that you may create for each system, or tailor using symbols, sharing PARMLIB for resource identification are:

LOADxx is used to specify the master catalog, select the I/O configuration, an alternate nucleus, a NUCLST member (NUCLSTxx), a symbol definition member (IEASYMxx), the sysplex name, and sysparm members (IEASYSxx). It is recommended to place this member in SYSn.IPLPARM rather than in SYS1.PARMLIB. It seems preferable to specify the sysparm in the IEASYMxx member. If you have the SYSPARM parameter coded, this will be concatenated to any specified in IEASYMxx. OS/390 Release 2 introduces some enhancements to LOADxx.

- The SYSPARM statement can be used to specify the logical PARMLIB concatenation (there can be 10 data sets, in addition to SYS1.PARMLIB that make up the concatenation).
- Statements in LOADxx can be associated with a specific platform (basic mode machine, LPAR, or VM virtual machine). This allows a single LOADxx to be easily shared by many systems. For example, each system′s master catalog name can be specified on a SYSCAT statement for each platform.

IEASYMxx This member is used to specify the symbols in use as well as the different IEASYSxx members to use for each image, and the system name for each image.

IEASYSxx It points to all other system-specific members. It also should contain a unique system name. If in a sysplex, it *must* contain a unique system name. The system name is probably better specified in IEASYMxx. Page and swap data sets, LOGREC data sets, and VIO journal data sets (STGINDX) can be defined using &SYSNAME.. This allows you to specify one set of parameters for a data set name, which can be shared by all systems in the sysplex and resolves to a unique name for each system.

COMMNDxx will usually need to be repeated for each system so that system-specific tasks can be started, or system-specific initialization parameters can be identified on START commands (the latter may be done using symbols).

SMFPRMxx will need to be unique for SID definition, unless you use symbols to set the SID, or use the form of definition that binds the SMF ID to a CPU serial number. The SID parameter supports the &SYSNAME. system symbol. This allows one SMFPRMxx definition to be used throughout the sysplex. SMF data set names are no longer required to be SYS1.MANx. You can now use &SYSPLEX., &SYSNAME., or &SID. (SMF ID) as system symbols for SMF data set names.

IEASLPxx SLIP supports the use of wild cards in job names, data space names and user IDs. This allows a single command to more easily set a trap on replicated jobs.

APPCPMxx identifies the APPC LU name, which should be unique within a network (again a symbolic value may eliminate the need for separate members).

IEFSSNxx may need to be unique if dealing with subsystems that need to be uniquely identified across systems, for example IRLM. It is now possible to use dynamic SSI to modify the subsystems to a system without requiring an IPL by using the SETSSI command.

COUPLExx if defining paths between systems in a sysplex, you need to uniquely identify paths between systems. Choosing a good scheme for defining XCF paths (as described in the redbook *MVS/ESA HCD and Dynamic I/O Reconfiguration*) will eliminate the need for separate members).

The following members may have different values due to differences in available capacity of the sharing systems. The extent to which you would have to redefine them depends on the configurations sharing PARMLIB, and the extent of use of symbols.

ASCHPMxx APPC/MVS scheduling information.

CONFIGxx If used for different configurations.

CONSOLxx Varied console configurations (We recommend that CONSOLxx members in a sysplex should be shared.)

CSVLLAxx Varied product configurations.

IXSPATxx Excessive Spin options.

GRSRNLxx Varied product configurations (NOTE: GRSRNLxx members in a sysplex must be the same).

IEAFIXxx Fixed LPA list.

IEAICSxx, IEAIPSxx, IEAOPTxx Performance parameters if you are not using WLM (workload manager).

IEALPAxx Modified LPA list.

IEAPAKxx LPA Pack list.

IECIOSxx Missing interrupt handler and hot I/O parameters.

IKJPRMxx TIOC parms.

LNKLSTxx Link library list.

LPALSTxx LPA library list.

TSOKEYxx TIOC parameters.

The principles noted above can apply to definitions that affect other system resources, such as procedure libraries, VTAMLST, NetView DSIPARM, and so forth. In some cases, such as VTAMLST, a collection of pointers in a startup member points to common and system-specific parameters. In others, such as NetView DSIPARMS, the system-specific parameters can be in data sets that are concatenated to a data set containing common parameters.

Product procedures may be shared by setting the procedure up with symbolic substitution values for specific instances, and resolving the substitution via parameters on the START command. If using override parameters, you should make the corresponding defaults inoperable, if possible, to prevent accidental initialization with unwanted values.

In determining what should be the same and what should vary, use the same principles as in PARMLIB, that is, look for unique resource identifiers, and

configuration variations, as well as pointers to the unique definitions to determine what parameters will vary.

3.5 Catalog Environment

This section discusses the catalog setup for the operational and maintenance environment.

Depending on the maintenance and operational environment design you chose to best fulfill your needs, an adequate catalog structure has to be built.

In the preceding figures, the maintenance environment and operational environment were designed so that the master catalog pointed to only one copy of any system data set. The selection of the OS/390 image to be IPLed is made by the setting of the IPL parameters at the system console (Figure 13 and Figure 14). Assuming that access to the system libraries at the lower maintenance level, which would serve as a backup, is practically unnecessary, the design of your system can be straightforward.

The available hardware configuration and the number of service levels you have to support may require that more than one copy of a data set has to be accessed from one OS/390 image. With the restriction that only one catalog entry can exist for each unique data set name, we will have to discuss other ways of accessing these data sets.

This book will consider the facilities provided by the catalog architecture, in particular, the indirect catalog function and system specific aliases.

Both these methods entail a certain administrative effort. Neither is ideal, and it must be left to the users to find the one that best satisfies their needs. Other means to achieve the same goal, such as including the VOLSER and device type in DDDEFs for the target and DLIB zones or use of JOB/STEPCATs, exist and are frequently used, but they will not be discussed in this publication.

Note: JOBCATs and STEPCATs cannot be used for SMS-managed data sets.

3.5.1 Indirect Catalog Function

A catalog entry for a data set contains, among other information, the volser and device type of the volume on which the data set resides. Cataloging a data set using the indirect catalog function results in a catalog entry that does not have the volume information. Those libraries, usually MVS target libraries, are cataloged in the master catalog with volume information of "******" and device type of "0000."

Note: The indirect catalog function can be used only for Non-VSAM data sets.

Libraries that are cataloged using the indirect catalog function must reside on the system residence volume, or a data-set-not-found condition arises. The indirect catalog function works for any library, whatever high-level qualifier or name it may have.

An approach to designing a system exploiting this function might be:

- In the operational environment, the indirect catalog function is used to reference system libraries located on the system residence volume. If you have a SPILL volume, you can only use this facility for the IPL volume.

Another technique must have to be used for this other one as mentioned before.

- All other system libraries (for example JES data sets, PARMLIB, and PROCLIB) are cataloged in the master catalog with specific volume information.

Such a system design would enable you to use different levels of target libraries independent of the IPL device you choose, and the ability to utilize the same master catalog.

Figure 16 shows an example of how the facility works when referencing SYS1.LINKLIB cataloged using the indirect catalog function (volume=******). SYS1.LINKLIB is located on the IPL volume. The active operational level of SYS1.LINKLIB is volume SYSRS1. Note that for the operational libraries, SYS1.PAGE01 and SYS1.HASPCKPT, the catalog data set entry has a specific volume pointer.

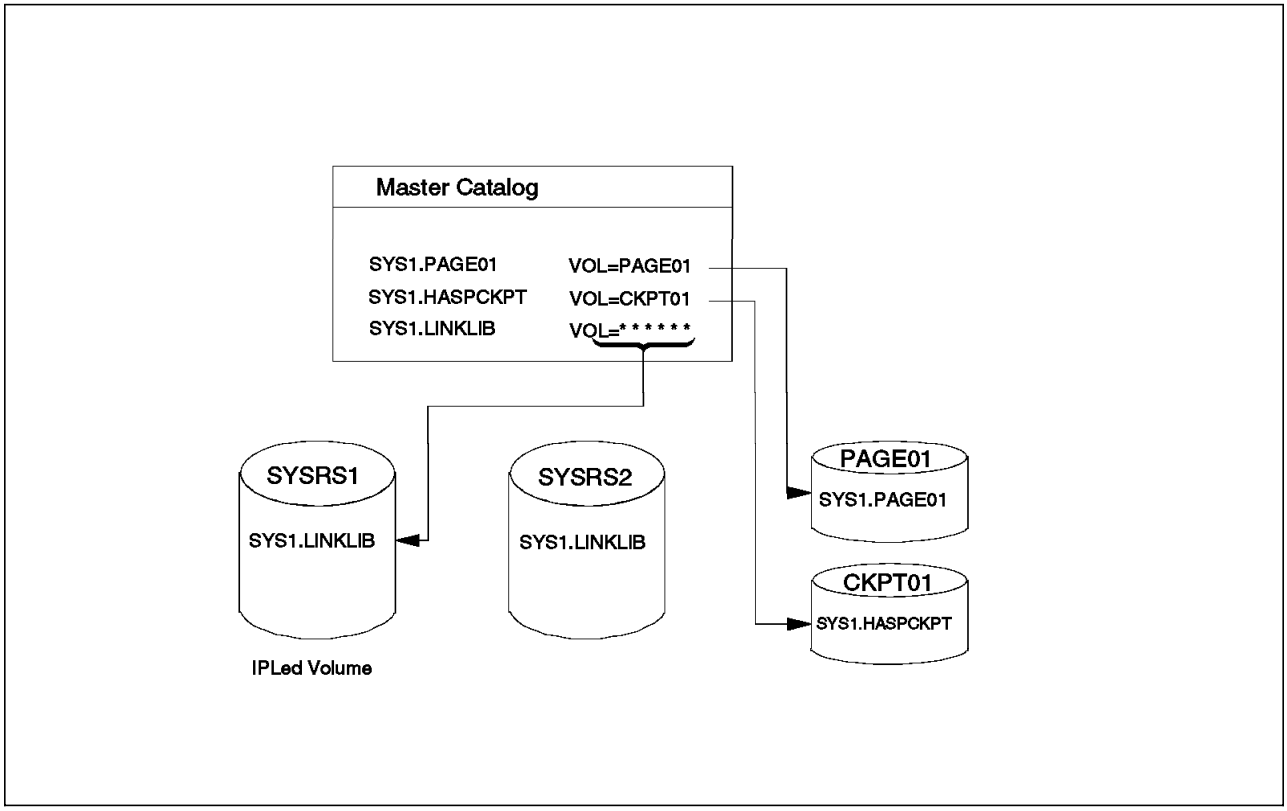


Figure 16. Indirect Catalog Function with SYSRS1

Figure 17 shows the indirect catalog function in the operational environment for SYS1.LINKLIB, where the active operational level is volume SYSRS2, the IPL volume. Note that the catalog pointers to system libraries SYS1.PAGE01 and SYS1.HASPCKPT are unaffected by the switch in IPL volumes. Therefore, it is possible to continue to use existing system libraries after a system upgrade.

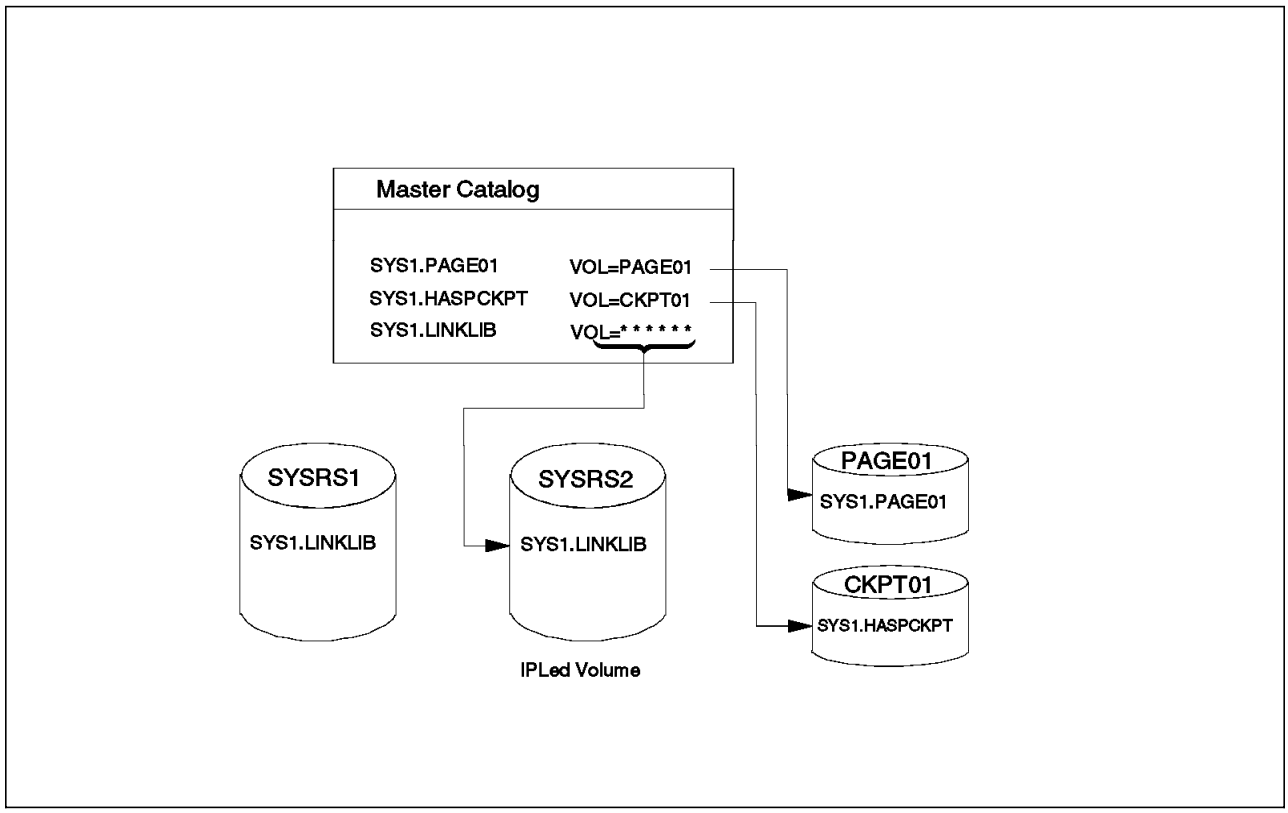


Figure 17. Indirect Catalog Function with SYSRS2

The indirect catalog function by itself does not provide a means to maintain multiple copies of libraries with SMP/E. It has to be used with either system-specific aliases, VOLSER and device type in the DDDEFs, or hard coded DDCARDS in the SMP/E procedure.

3.5.2 System-Specific Aliases

CBIPO and CustomPac offerings have introduced a function to deal with cataloging multiple data sets with the same name; namely, the concept of system-specific aliases (SSAs).

SSAs are used to locate data sets across systems, even if they have the same names, without using a JOBCAT, STEPCAT, or specifying the volume the data set resides on (VOL=SER).

Users may not find SSAs to be ideally suited for their installation. Nevertheless, it is a useful approach for managing duplicate data sets within one OS/390 image. A suitable structure and naming convention can always be chosen to conform to existing installation standards.

3.5.2.1 ServerPac SSA Implementation

SSA implementation by ServerPac provides for alias pointers to the SYSRES user catalog and the DLIB user catalog. DLIB data sets can now be accessed without a volume specification. An example of SSA implementation is shown in Figure 18.

The figure shows the access from System A to SYS1.LINKLIB and DCF.DCFLOAD on System B. It is assumed that System A is used as a driver system for the maintenance of System B. All System B′s catalogs are connected to System A′s

master catalog as user catalogs. A simple naming convention has been chosen whereby all System B′s system data sets will be accessed from System A with a high-level qualifier of TARG; the data sets belonging to program products use a high-level qualifier of PP. SSAs are shown only for the libraries on System B. It is certainly possible to define SSAs in such a way to allow both systems to be either the driver or target system.

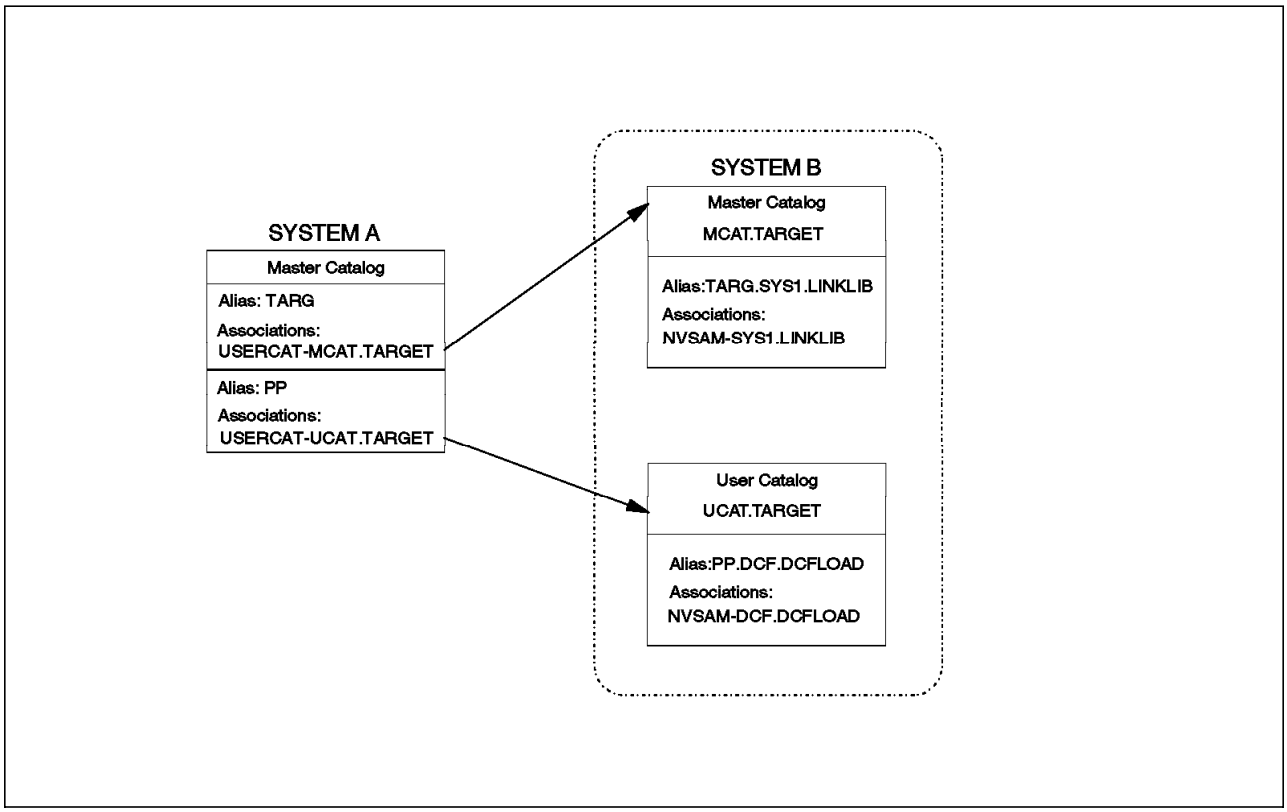


Figure 18. ServerPac SSA Implementation

As the figure illustrates, every data set can be managed with SSAs, regardless of whether it is a SYS1 data set, a library for a program product, or user data.

When allocating a new data set, the name that a data set is allocated with is the name that is placed in the VTOC. This name is also displayed when using ISPF option 3.4. When searching the master catalog, the SSA name has to be used as the high-level qualifier.

There are implications when security software products are used. For instance, installations running RACF will have to authorize both the SSA and the high-level qualifier of the actual data set name (as seen in an ISPF 3.4 display) to RACF.

SSAs have been incorporated in ServerPacs in the OS/390 DBS, NCP, and CICS environments. Products installed individually or installed using CBPDO are not delivered with SSAs, but can certainly be defined this way during installation.

A unique system-specific alias is defined for each catalog. All data sets defined by the ServerPac feature will have an alias defined.

With OS/390, you must specify which system-specific alias (SSA) should be used to gain access to a catalog from the driving system in the ServerPac. The same is true for the ServerPac subsystem.

For more information, refer to *IBM ServerPac for OS/390 Introduction to CustomPac Installation Dialogs* that is shipped as part of the ServerPac offering.

3.6 Physical Design

This section discusses the physical components of an installed OS/390 image, and gives some guidelines for the physical design.

Before installing an OS/390 image for the first time, you must think about how you will maintain your OS/390 images once they are built, and devise a design that will best aid you in achieving this maintenance strategy. Some of the questions you must ask yourself are:

- Will the maintenance environment be run on a dedicated machine, LPAR partition, or VM machine?
- How many DASD volumes are required?
- How many volumes are available?
- What are the volumes used for?
- Where do I place the different libraries?

ServerPac installation dialogs provide a default layout, and also allow you to choose the placement of the system data sets. The installation dialog is described in more detail in section 4.1.3, “Installation” on page 79.

3.6.1 Maintenance Environment

The maintenance environment consists of two separate components; namely:

- The production maintenance level
- The upgraded maintenance level

  Both contains the following data sets:

  – Master catalog
  – Target libraries
  – User catalogs
  – SMP/E global zone, SMPPTS, SMPMTS, SMPSCDS, SMPSTS and logs
  – SMP/E target zone
  – SMP/E distribution zone
  – Distribution libraries

  Only the target libraries and the catalogs are needed for testing; all the others are only needed for maintenance.

If DASD space is a real concern, it would be possible to share the SMP/E global zone and all its associated data sets between the production operational and the maintenance level. The advantages and disadvantages of different global zone configurations are discussed in section 3.3.1, “Considerations” on page 45. Having separate SMP/E environments would also allow easy propagation of an OS/390 image into production where the system and the SMP/E environment match exactly.

As the upgraded maintenance level is the one that is periodically updated and tested to be later propagated into production, the production maintenance level for most of the time will only be a backup copy of the operational OS/390 image. Taking this usage into account, a physical design where the maintenance

environment is running on a dedicated processor, LPAR partition, or VM machine appears to fulfill all the requirements. The recent generations of processors allow the use of load parameters. Load parameters allow the user to effectively select the I/O configuration, master catalog, NUCLEUS, and the SYSPARM suffix when IPLing the system. Making use of this hardware feature allows a very high degree of flexibility in activating and testing changes.

The suggested design places each of these components in its own DASD pool. The size of each pool is dictated by the number of products installed.

3.6.2 Operational Environment

When developing an operational environment design, the most important factors to take into account are:

- Performance
- Contention
- Availability
- DASD resources

Of these, DASD resources may be the limiting factor. If an operational environment has only one DASD volume at its disposal, then all the system libraries must be placed on that one volume. This operational environment works, but there are no contingency measures in case of a hardware failure, and performance may be limited.

The volume containing all libraries propagated from the maintenance environment (system residence volume) is the most critical component in any operational environment; without it, an OS/390 image cannot IPL or function.

Additional operational data sets that are critical when IPLing an OS/390 image include:

- Master catalog
- SYS1.PARMLIB
- SYS1.PROCLIB
- LOGREC data sets
- SYS1.BRODCAST
- SYS1.UADS
- PLPA page data set
- Common page data set
- Local page data set
- IODF files
- Couple data sets (if in sysplex)

Some other data sets are also critical. They will not prevent the system from IPLing, but would make operation very difficult or impossible. These data sets are:

- JES-managed data sets (SPOOL, CHKPT)
- RACF primary data sets and RACF backup data sets

As long as performance considerations do not require a different approach, it makes sense to have these libraries contained on as few volumes as possible, even just one. This volume is referred to as the “master catalog volume,” and it contains most, if not all, the critical libraries listed above that do not reside on the system residence volume.

By carefully placing the data sets, we can be sure that in the event of a hardware or non-system software error, provided that we have access to two volumes (the system residence volume and the master catalog volume), we are able to IPL OS/390. In essence, what we have is a “2-volume system.”

This is fine from an availability standpoint. However, it is still necessary to consider performance and contention. Data sets that are highly used or have to be accessed quickly include:

- PLPA page data set
- Common page data set
- Local page data sets
- JES-managed data sets
- RACF primary data sets
- RACF back-up data sets
- Couple data sets (if in sysplex)

Depending on the products installed in your environment, this list could grow. For example, if you have DFSMShsm installed, its control data sets must be included in the list. However, the concepts and considerations are the same.

A compromise has to be reached, and to a large extent the factors affecting this compromise (performance, contention, DASD resource, and availability) are different for each operational environment. The key factor is how much DASD can be used for your operational environment.

3.6.2.1 Operational Environment Example

In the example of an operational environment in Figure 19, an attempt has been made to balance the relevant factors. This system is an OS/390 system running JES2 and RACF, providing both batch and TSO services.

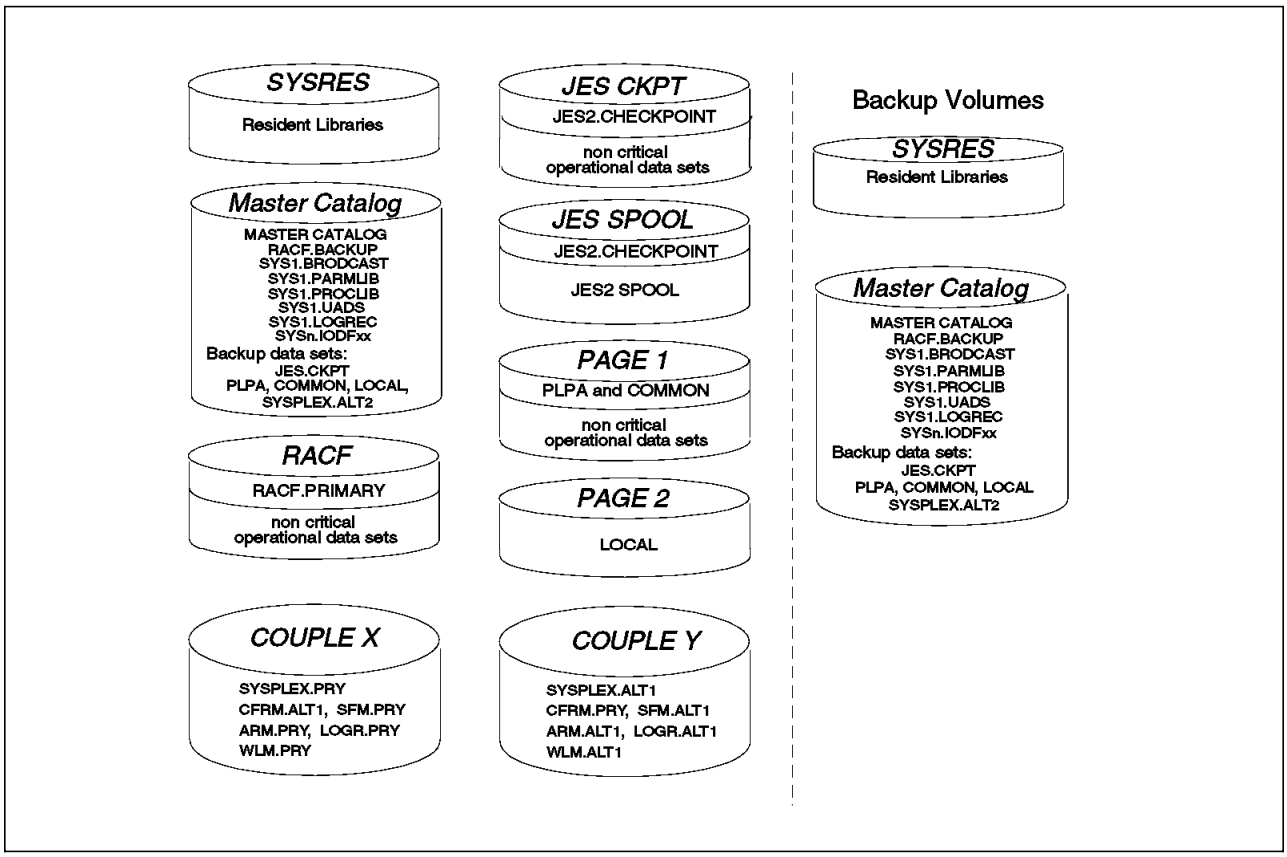


Figure 19. Operational Environment Physical Design

The design criteria for this operational environment are:

- The JES checkpoint data set is highly used, and is critical to the system. To increase availability, you take advantage of checkpoint duplexing. To maximize performance and reduce contention, place the primary JES checkpoint data set on its own dedicated volume. The duplex checkpoint data set is placed on the same volume as the JES spool. This will utilize the internal mechanism of JES that checks during every write to the JES spool to determine whether a checkpoint data set resides on the volume. This reduces I/O and improves performance.

  OS/390 JES supports the option of placing the checkpoint data set on the coupling facility (CF) when you are in a Parallel Sysplex environment, to improve the access time. This option is compatible with the checkpoint data set on DASD. The use of the coupling facility for the checkpoint is completely optional. The support includes the use of DASD or the coupling facility for the primary and/or secondary checkpoint data sets in all possible combinations.

  Back-up JES spool and checkpoint data sets are allocated on the master catalog volume to maintain the ability to IPL the OS/390 image from just the system residence volume and the master catalog volume by using an alternate load parameter.

- Page data sets are also very active, and are critical to the system. To increase performance, place the PLPA page data set and common page data set on the same volume. Allocate a PLPA page data set of one cylinder and a larger common page data set. At IPL, the PLPA spills into the common

page data set and, for I/O purposes, they act as one data set, increasing performance.

To improve availability, duplex the PLPA and common page data sets.

Each local page data set is allocated on a separate dedicated volume with a size based on the device type and the system requirements.

Backup PLPA, common, and local page data sets are allocated on the master catalog volume to maintain the ability to IPL the OS/390 image from just the system residence volume and the master catalog volume, selected from an alternate IEASYSxx member.

- The RACF primary and backup data sets are highly used, and are critical to the system. To increase performance and reduce contention, the primary data set should *not* be placed on the same volume as any of the data sets already mentioned.

  A reduction in I/O operations and a consequent performance improvement can be realized by using in-storage profiles. *System Programming Library: RACF* devotes a chapter to RACF performance considerations.

  The RACF backup data set is placed on the master catalog volume to maintain the ability to IPL the OS/390 image from just the system residence volume and the master catalog volume.

  When using RACF in a sysplex, take advantage of RACF′s *data sharing mode*, the mode in which RACF uses the coupling facility for data sharing. With the use of global resource serialization (GRS) and with caching of RACF data base on the coupling facility, you have the potential to improve performance of access to RACF data in a sysplex. For performance purposes, place the primary and the alternate structures in a separate coupling facility. Spread the primary structures across the available coupling facility as their space requirements and access requirements allow, trying to keep the alternate database structures separated from the primary database structures.

- The system residence volume and master catalog volume are the most critical volumes in the operational environment. During periods when the system is not being upgraded, the system residence volume has an online backup in the form of the second system residence volume. It is desirable to also have an online backup of the master catalog volume to maintain maximum availability.

  Depending on your requirements and available resources, there could be other backup variations. For instance, if you require a very short restart time, you may consider placing a copy of the system residence volume on a temporary volume while your normal backup copy is upgraded.

  On the other hand, when DASD space is a real concern, you may consider having no online backup, but a backup copy on tape.

- If you are in a Parallel Sysplex environment, the couple data sets should not be placed on volumes that have high I/O activity, subject to “reserves,” have page data sets or may have an SVC dummy data set allocated and written to. These same considerations are valid when you are selecting a volume for an alternate couple data set. For maximum performance and availability, each couple data set should be on its own volume. However, this is an expensive approach. A reasonable approach is to have one volume for all the primary couple data sets, excluding the primary CFRM (coupling facility

resource management) couple data set. This should be placed on other volumes with all the other alternate couple data sets.

- The remaining low-activity critical data sets are allocated on the master catalog volume. They include:
  - SYS1.BRODCAST
  - SYS1.LOGREC
  - SYS1.PARMLIB
  - SYS1.PROCLIB
  - SYS1.UADS
  - HLQ.IODFxx

  This design does not use DASD space very efficiently. In particular, the JES checkpoint, the RACF volumes, and paging volumes will have a large amount of free space. This free space could be used for very low activity libraries, such as JES spool off-load libraries, DLIBs, and the remaining non-critical system data sets.

  While the following data sets can be classified as being non-critical to the operation of the system, they may be critical to your business. Care should be taken with their placement to avoid contention:

  - SMF data sets
  - SVC dump data sets
  - IPCS data sets

Note that more than one JES spool volume, or local page volume, or both, may be required, depending on the size of your system.

If sufficient DASD for this design is not available, you might consider running without the JES checkpoint volume and using one checkpoint on the spool volume, or running with the exposure of having no online backup of the master catalog volume.

By adopting the same approach to your physical operational environment, you should have similar layouts on all your operational environments. This has several advantages, including:

- All systems look the same to the system operators, and as a result are easier to operate.
- All systems look the same to the systems programmers, and as a result are easier to maintain.

The SMP/E global zone and all its associated libraries can be shared among all the operational environments.

3.6.2.2 Two-Volume System

From a performance view point, the active critical data sets have been spread as much as possible. At the same time, by allocating inactive backup versions of most of these critical data sets, a 2-volume system has been maintained.

Having a 2-volume system has the following advantages and uses:

- It can save time and effort when building a new operational environment.
- During any complex failure, if the 2-volume system is available, it is possible to bring up some form of operational environment, possibly preventing the need for stand-alone restores.

- In the event of an inability to IPL an OS/390 image, you need only to stand-alone restore the 2-volume system. The other operational environment volumes can be restored online, which saves time.
- With this 2-volume system in every operational environment, there is no need for a separate starter system. This saves DASD, and in a multi-location enterprise, every location has the capacity for rapid recovery.
- The 2-volume system is contained within every operational environment. Hence, little additional effort is required to keep it current. This is not the case if you maintain a separate starter system, which would have to be upgraded and maintained separately.

It is especially important in a single-system image environment, without shared DASD with other images, to have a 2-volume system that can be IPLed in an emergency.

3.7 Cloning an Existing System

The process of cloning involves making an exact copy of a system level in either the maintenance environment or the operational environment. A system is cloned to propagate a certain system level into production or to create an additional maintenance environment. You may, for instance, want to clone your existing production maintenance level to create a new upgraded maintenance level prior to installing a CustomPac.

In this chapter we will not consider propagation; it is dealt with in section 6.6, “Propagating a System into Production” on page 145. We will concentrate on creating a new maintenance environment. The copies of target and distribution data sets on a propagated system are maintained for a certain time on an existing driving system. Once maintenance has been applied and the desired service level has been reached, the libraries can be used to form a separate OS/390 image. But, while still maintained by a driving system, a way must be found to deal with duplicate data set names.

Before you can start copying any libraries, you have to define how you will handle duplicate data sets. There is no “ideal” way of dealing with this problem. If you have developed a method in the past that suits your needs, there is no reason to change. If for some reason you have to reevaluate your procedures or have to develop new methods, you may consider using System Specific Aliases (SSAs).

You will have to perform the activities listed below to create a new OS/390 image. The final product, after executing the whole list, is an IPLable system based on the system it was copied from.

The necessary activities can be summarized as follows:

- Initialize the operational environment volumes.
- Allocate a new master catalog and, if applicable, user catalogs.

  The catalog structure is quite critical, not in the sense that you cannot get your new system to run, but in terms of saving work when you have to move your newly created system level into production. See section 3.5, “Catalog Environment” on page 61 for more information.

- Copy the target libraries, and add the respective entries in the new master catalog and user catalogs.

  This is the step where the chosen method of cloning has to be taken care of. If you chose to use SSAs, now is the time to catalog the data sets and define their aliases. If you are using the indirect catalog function, it is now that you have to catalog your data sets accordingly. For details on how to catalog data sets using the indirect catalog function, see *OS/390 DFSMS Access Method Services for the Integrated Catalog Facility*.

- The following system data sets will not be copied, but have to be allocated and cataloged in your newly created system:

  - JES managed data sets
  - ISF.HASPINDX (if using SDSF)
  - PAGE data sets
  - SMF data sets
  - SVC dump data sets
  - SYS1.DAE
  - LOGREC data set
  - Couple data sets (if in sysplex)

- Copy the following operational data sets. Most contain members that were tailored to fit your old environment. Even though they may not exactly fit what you are planning, it will probably be easier to change the already customized members than to start with the defaults:

  - SYS1.PARMLIB
  - SYS1.PROCLIB
  - SYS1.BRODCAST
  - SYS1.UADS
  - SYS1.VTAMLST

  This list is by no means exhaustive; you may have user libraries that will have to be copied. It all depends on how your system was previously built.

- Tailor PARMLIB and PROCLIB. Some of the PARMLIB members can be customized on the driving system using a different suffix.

- Allocate and initialize the RACF primary and backup data sets. Refer to *System Programming Library: RACF* for more details.

- Tailor the RACF data set to reflect the new environment.

- Set up and tailor VTAM definitions to reflect the new system configuration.

- Set up the SMP/E operational environment (SMP PTS, and SMP logs) and copy the global zone from the driving system.

- Copy the SMP/E target zone.

- Change the DDDEFs so they match your data sets. If you have decided to use SSAs, they will contain the alias names. Other methods may require different alterations, such as adding device type and volser, all depending on the particular method chosen.

- Create an IODF configuration for the new system if you need.

- Write the IPL records on your new SYSRES volume. Refer to the REFORMAT command in *Device Support Facilities, User's Guide and Reference*.

- Update the master catalog pointer in SYS1.NUCLEUS or update the SYSCAT statement in the LOADxx member of SYSn.IPLPARM or SYS1.PARMLIB.

- Finish preparing the LOADxx member in SYSn.IPLPARM or SYS1.PARMLIB. See the *OS/390 MVS Initialization and Tuning Reference*.

Chapter 4. Installation

This chapter deals with the installation part of software management as shown in Figure 20. The following will be discussed:

- System replacement
- System upgrade
- OS/390 upgrade path
- Environment setup for OpenEdition and DCE

IBM provides a variety of packages to help you replace or upgrade the system and these are described in 2.5.1, “Replacement Methods” on page 23 and 2.5.3, “Upgrade Methods” on page 26.

The direction of OS/390 is that the *replacement* method should be used as the preferred method of upgrade. We will therefore examine the specific tasks required to install a ServerPac and also look at the upgrade methods that will allow you to position yourself for system replacement in the future.

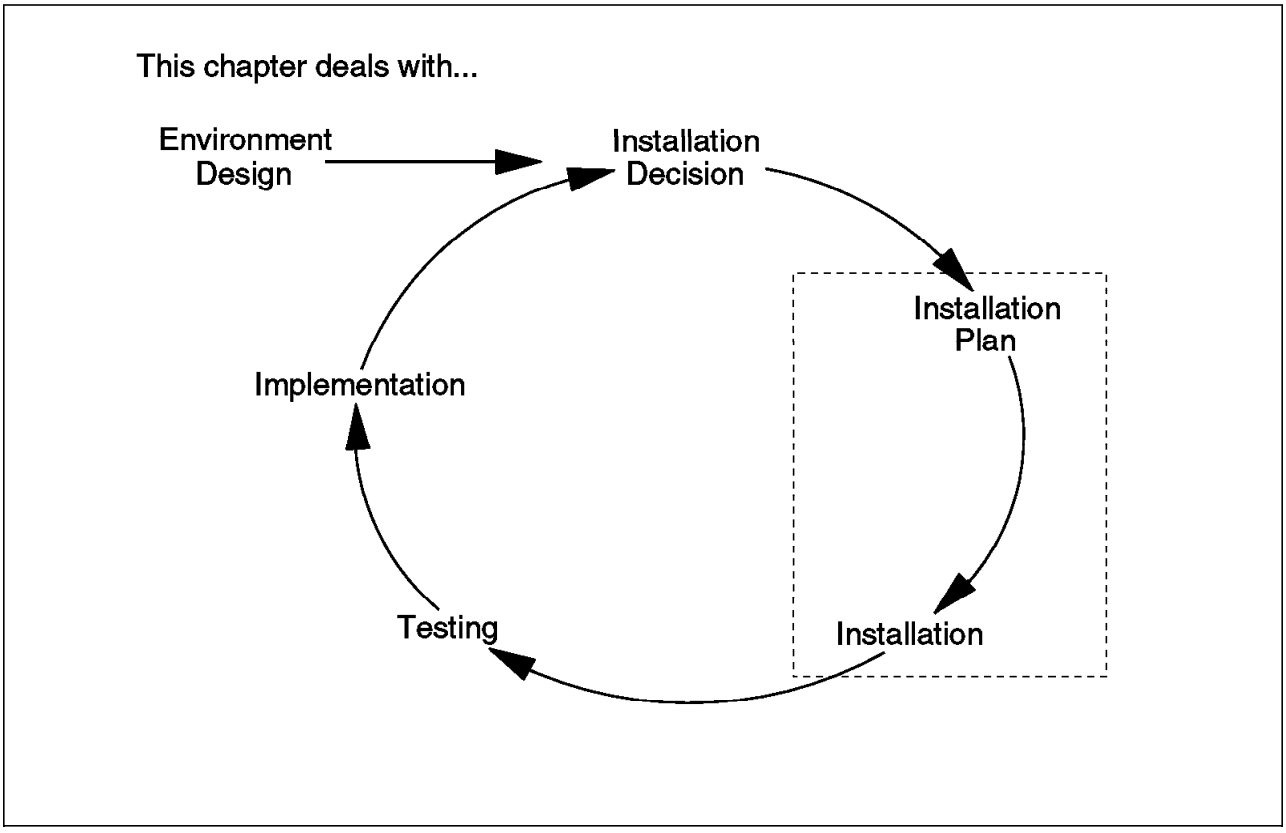


Figure 20. Installation

4.1 System Replacement

This section discusses the system replacement option based on the environment that was described in Chapter 3, “Maintenance Environment” on page 39. To replace a system level, the following steps are necessary:

1. Plan the system replacement
2. Install the new system level
3. Customize the new system level
4. Test the new system level

There may be a number of offerings available to you for system replacement. See section 2.5.1, “Replacement Methods” on page 23 for a brief description of some of the options. For this discussion, the ServerPac offering is used to illustrate how to replace a system in the maintenance environment. Refer to Chapter 3, “Maintenance Environment” on page 39 for a discussion of how to design the environment.

4.1.1 Planning System Replacement

In section 2.3, “Approach for Keeping Current” on page 17, the activities required to update a system are discussed. When you decided to replace the system, factors that influenced your decision process included:

- How old is my current system?
- How many new products do I need?
- How many of the products have changed release or version level?
- How much time do I have to implement the new maintenance level in the operational environment?
- How many other code changes are required?
- What will the impact be on storage management?

Careful planning is a must for a smooth installation. The following sections describe an installation.

4.1.1.1 Installation Plan

The installation plan is a document describing all installation activities. It is created through an iterative process involving staff work and discussions at planning meetings. When completed, this plan provides the framework around which the installation of, and conversion to, a new system is to be performed. It outlines all task activities in sufficient detail to be used both as a guide and as a checklist. Included in the plan are:

- Objectives
- Project team organization and responsibilities
- Acceptance criteria for the new system
- Hardware (including DASD and processor storage), software, and personnel requirements
- Installation time frame and schedule
- Inventory of the system maintained software such as products (both IBM and non-IBM), exits, SVCs, and user modifications
- Task definitions and assignments
- Definition of periodic checkpoint criteria and milestone indicators
- Education and training plan
- Functional-test, application-test, and stress-test plans
- Problem and change management structure

- Implementation (production introduction) plan

4.1.1.2 Installation Tasks

An important part of the planning process is the identification of all tasks or work items that must be completed before the new system level is propagated to the operational environment. These tasks, sometimes referred to as implementation tasks, must be defined in sufficient detail to describe:

- Scope of effort
- Degree of complexity
- Required resources
- Completion criteria

Once the tasks are identified and defined, a time schedule for the installation can be developed. This schedule (called the installation schedule) lists the tasks in chronological order and includes estimated start and completion dates. The installation schedule, once developed, should be diagrammed pictorially and used as the basis for discussion and review at project meetings.

4.1.2 CustomPac Ordering and Delivery

When you order IBM program products for delivery by traditional means, you get the product software along with its installation, customization and operation documentation. Your system programmers must then read through much of the documentation and customize much of the installation JCL to suit your site, as well as possibly modifying other products′ installation configurations to take advantage of, or even be compatible with, the new products being installed. Installing new products in this fashion can be very time-consuming if the systems programmer has a number of products to install. CustomPac packages aim to reduce this time considerably, while also reducing the amount of computer resource needed to install IBM program products.

The CustomPac approach to product delivery brings IBM software in a form better suited to your requirements. You get a simple, standardized installation process. The installation itself is done using the CustomPac installation dialogs. The dialogs guide you through the whole installation process, and can be dynamically extended by defining your own variables and post-installation jobs. All changes and new definitions can be saved with the dialogs and used during the next installation of any CustomPac.

For most types of CustomPacs, the Software Service Manufacturing Center (SSMC) also needs a current SMP/E Consolidated Software Inventory (CSI), for the target system. This data is then used to create your CustomPac. When ordering a ProductPac or ServicePac, it is important that the system be frozen (no new products or maintenance added) after the CSI is taken so that maintenance included in the package will fit the target system on delivery. The CSI is not required when ordering a ServerPac.

An overview of the production process can be seen in Figure 21. Each CustomPac is produced in response to your specific order and your specific customization data. The products and service are tailored to support those products and your specified data. Jobs and related installation materials (RIMs) will be further customized to the products and service that were ordered. The SSMC uses a software manufacturing process to build and package your CustomPacs according to your specifications. To produce CustomPacs, the SSMC maintains a database that contains:

- IBM licensed programs supported by CustomPacs for OS/390 systems and subsystems
- PTF service for all products, including current error hold and HIPER information (updated on a daily basis)
- IBM developed materials, such as installation procedures, sample JCL, and job streams to help you install and customize your system

SMP/E is used to verify correct product and service installability. Therefore, you do not have to research PTF pre- and co-requisite chains, or investigate any PE corrections you need.

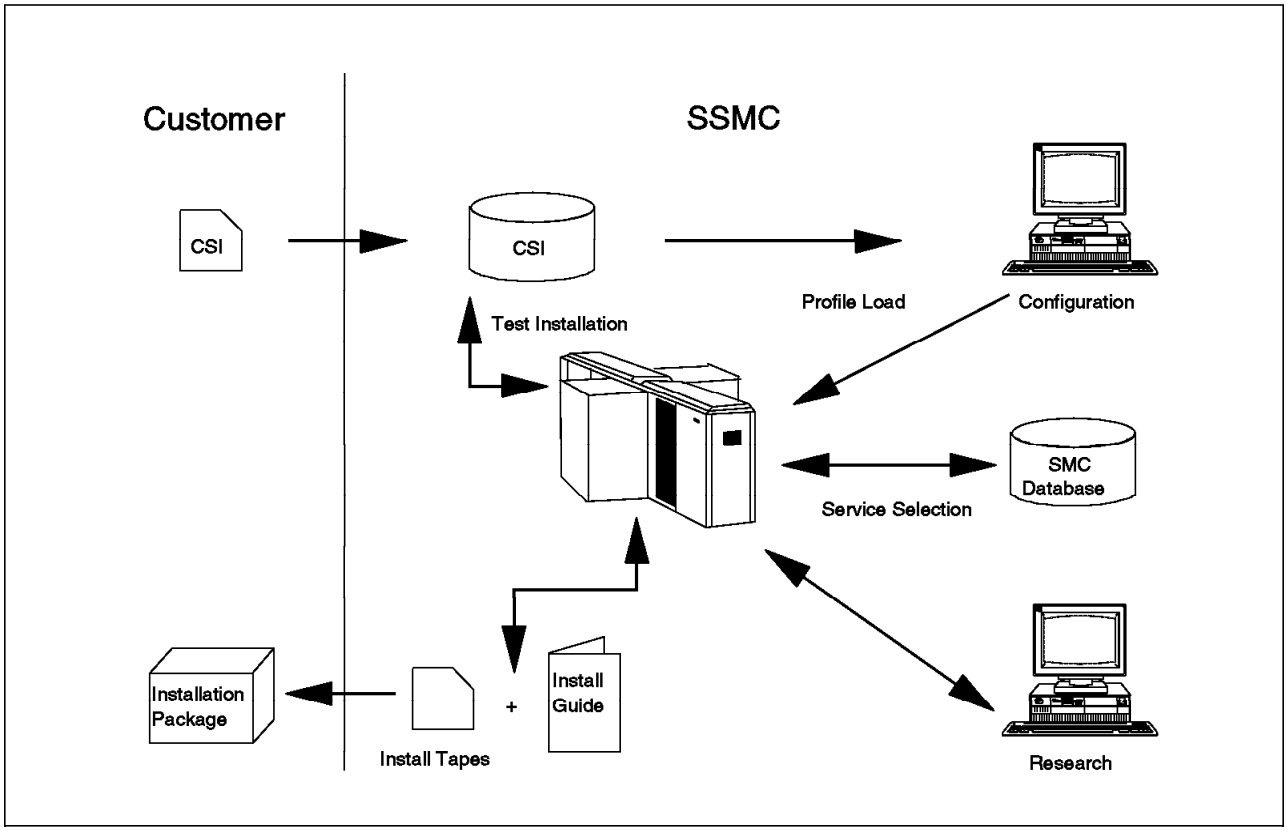


Figure 21. SSMC Process Overview

Using the CSI tape and your order, the local SSMC builds a package (of appropriate type) for you. The package includes all the documentation that you would normally get with the product tapes, as well as tailored documentation for installing the package, and tapes or cartridges containing:

- Products
- Service
- Related Installation Materials (RIMs) containing all information and jobs for installation

The installation dialogs are also shipped on the RIM tape.

Included with your order will be a hardcopy installation manual that is specific to your order. This is also included on the RIM tape along with the CustomPac Installation Dialog Reference Manual. Both will be required to assist with the installation.

There may be one or more Selective Follow on Services included in the purchase price of CustomPac; see section 1.5, “OS/390 Software Delivery Offerings” on page 5. A Selective Follow on Service is an added-value service upgrade whose contents are based on your maintenance history. A ServerPac does not include SFS.

4.1.3 Installation

This section discusses in more detail the actual steps required to install a ServerPac.

4.1.3.1 Getting Started

You will receive the following items with your ServerPac package:

- Hardcopy installation guide containing general information about the package, a description of machine-readable materials within this package, all installation activities to be performed, and an appendix with all reports created for this ServerPac. You will also find instructions on how to unload and start the dialog.
- System tapes containing all target and distribution libraries in dump by data set format.
- RIM and service tapes containing all information pertinent to the installation of this package, including the dialog.

The dialog guides you through the ServerPac installation. Extensive options are available for tailoring based on your specific installation requirements. You can change information about data set names, space allocation, volumes on which data sets reside, and catalogs where data sets are defined. The dialog saves your customized data for future CustomPac installations. The dialogs enable you to define your own variables interactively as well as your own installation jobs (for example, JCL to reinstall user modifications). Any changes or additions you made can be saved and used again when you install subsequent CustomPac packages.

Before beginning the install, you should review the documentation supplied on the RIM tape entitled *Additional Documentation in LIST3820 Format* (file 29). This contains the following:

DIAGREFM *Installation Guide* (provided hardcopy)

ORDERREF *CustomPac Installation Dialog Reference Manual*

DIAGMSGB *CustomPac Installation Dialog Message Book*

The *CustomPac Installation Dialog Reference Manual* is essential if this is the first CustomPac to be installed on your system. Before any CustomPac can be installed, the dialogs must be available on the driving system. Once these have been downloaded they will remain on the system and can be used for future CustomPac installations. As part of the CustomPac RECEIVE function, version checking and updating of the dialogs for the specific order will take place. Hence the dialog installation jobs in the *CustomPac Installation Dialog Reference Manual* are run once and *not* with every order.

4.1.3.2 RECEIVE the Order

The *CustomPac Installation Dialog Reference Manual* offers alternative methods of invoking the CustomPac Dialogs. Whichever method you use will cause the Primary Option Menu to be displayed (see Figure 22). You are now ready to RECEIVE the order. If the dialog environment needs to be refreshed, then that will take place at this time. The order inventory is also updated.

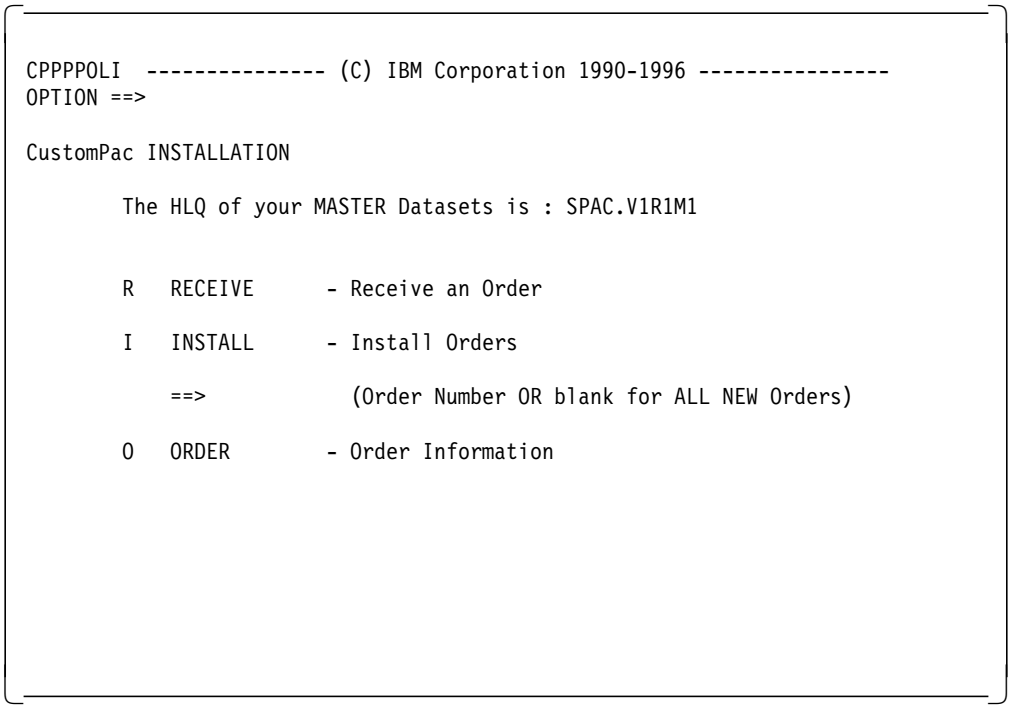


Figure 22. CustomPac Installation Dialog - Primary Option Menu

4.1.3.3 INSTALL the Order

You are now ready to download the actual software which you have ordered as part of your ServerPac. Select I from the Primary Option Menu for the Order Installation panel shown in Figure 23. Type an S against your ServerPac order and the Installation panel shown in Figure 24 is displayed. It is from here that all functions needed to install the ServerPac are called. As each step is completed, the following step becomes available for selection.

```
CPPP6071  -------------------- ORDER INSTALLATION ------------ Row 1 to 1 of 1
COMMAND ==>                                                    SCROLL ==> CSR

PRIM Cmds:(? SET L F N P REFresh STatus SORT VErbose VERsion)
LINE Cmds:<Copy Delete Edit Finalize Insert Products Report Select>

    ORDer   PROFile   SYStem         PAC       Prod        Change    Change
 S  Number  Number     Name   SREL   TYPE      DATE    St   USER      DATE
 - -------- -------- -------- ---- -------- ---------- -- -------- ----------
   DF031072                  Z038 EXP      01/01/1996 S  AYRESR   08/03/1996
****************************** Bottom of data *******************************
```

Figure 23. CustomPac Installation Dialog - Order Installation - Order Display

```
CPPPFLOW  --------------- (C) IBM Corporation 1990-1996 ----------------
OPTION ==>

Installation

Order ( DF031072 ) The Following Functions MUST be Executed in Sequence

   C    Configure        Select Configuration for Installation and Merge
   *    Variables        Define Installation Variables
   *    Modify           Modify System Layout
   *    Alias            Define Alias to Catalog Relationships
   *    SSA              Define SSA to Catalog Relationships
   *    Installation     Select and Submit Installation Jobs
   *    Save             Save Used Configuration
   *    Update           Update Order Inventory Status
   DI   Display          Display a List of Dataset Names
```

Figure 24. CustomPac Installation Dialog - Installation Panel

Configure: When an order is shipped to a customer it is installed using control information stored in tables. This Dialog verifies these tables and copies them to work versions. If this is the first CustomPac to be installed, then a CR command to create a new configuration is all that is required. For more

information on what a *Configuration* is, see Chapter 9 of the *CustomPac Installation Dialog Reference Manual*.

Variables: This takes you to the Variable Selection List (VSL) shown in Figure 25. Installation of the CustomPac involves the generation and submission of batch jobs. That work is carried out in the Installation Dialog. The Variables Dialog prepares for that step by allowing you to define user variables and alter data values for non-customized variables. Appendix A of the *Installation Guide* has a checklist that can be used as a planning aid for this.

```
CPPP6111  ----------- Installation Variables ( DFO31072 ) -- Row 1 to 14 of 77
COMMAND ==>                                                  SCROLL ==> CSR

Variable Selection List                                      SHOW( —C       )

PRIM Cmds:(? SET L F N P CANcel SAVE SHow VARname)
LINE Cmds:(Browse Delete Edit Insert Repeat Ship)

 S     Synonym            STA  Contents
 - --- -----------------  ---  ----+----0----+----0----+----0----+----0----
   ==> GEN SYSTEMPAC
       ASSEMBLER NAME       D  ASMA90
       VVDS SPACE           D  CYLINDERS(2 2)
       SMPTLIB PREFIX       D  SYS1.MVS
       NEW DLIB ZNAME       P  MVSDZN
       NEW TARGET ZNAME     P  MVSTZN
       SMPTLIB UNIT         D  3390
       SMPTLIB VOLSER       D  OS3RS2
       HFS VOLUME           D  OS3RS2
       HFS UNIT             D  3390

   ==> M 4.3.0
       ==> C COMM LIB       D  EDC.V2R1M0.SEDCBASE
       ==> C COMM LIB       D  PLI.V2R3M0.SIBMBASE
```

Figure 25. CustomPac - Variable Selection List

Modify: Shipped as part of your ServerPac is an ISPF table known as the Logical Volume Table (LVT). This table lists all the data sets that come as part of your order. The data sets for each product are assigned to a logical volume which in turn is mapped to a physical volume. Each product can be selected in turn (see Figure 26) and the data set profile for that product can be viewed (Figure 28 on page 85).

```
CPPP6051  ------------- Modify System Layout ( DF031072 ) - Row 14 to 27 of 40
COMMAND ==>                                                  SCROLL ==> CSR

SUMMARY Of Products

PRIM Cmds:(? SET F L N P SORT ALL CANcel SAVE SHIP DEVT SUML SUMP)
LINE Cmds:<Dslist Select>

        S  Product                                VSLname   Datasets
        -  ---------------------------------------- --------  --------
           IBM 3270-PC FTP FOR TSO REL 1.1          PCFT111        1
           ISPF              ENU 4.02.0             ISPF420       31
           LAN SERV FOR OS/390  1.1.0               OSLS110       31
           LANRES FOR OS/390    1.1.0               OSLR110       34
           LE FOR MVS            1.05.0             ADLE150       38
           MVS TCP/IP XA/ESA     3.01.0             TCPX310       63
        s  MVS/BDT VER 2 REL 1 BASE                 BDTB210       12
           NETVIEW CS            2.04.0             NCSE240       42
           NETVIEW PM            2.02.0             NPMB220       19
           OE DCE BASE SRVCS     5.01.0             OEDC510       23
           OE DCE DFS                  5.2.2        O2DF522       15
           OE DEBUGGER JES2      5.02.2             OEDB522        2
           OE SHELLS/UT JES2     5.02.2             OESU522        2
           OGL/370 ENU 1.01.0                       OGLE110        2
```

Figure 26. CustomPac - Summary of Products

The only required input within *Modify* is to specify the physical volumes in your system. This is done using the SUMP command and filling in the panel in Figure 27. ServerPac groups data set profiles using the following logical volume names:

- CATxxx Data sets that reside on the Catalog volume
- RESxxx Data sets that reside on the SYSRES volume
- DLBxxx Data sets that reside on a distribution library volume

As a minimum, three physical volumes are recommended for the installation of your ServerPac.

```
CPPP6055  ------------- Modify System Layout ( DF031072 ) ---- Row 1 to 4 of 4
COMMAND ==>                                                   SCROLL ==> CSR

SUMMARY Of Physical Volumes

PRIM Cmds:(? SET L F N P SORT DEVT)
LINE Cmds:<Assign Dslist>

      S  PVol     CUU   DEV name         CYLs    Largest         Datasets
      -  ------   ---   --------         ----    -------         --------
         OS3CAT   CF1   3390-3           1124       145              47
         OS3DL1   CF2   3390-3           2783       141             430
         OS3RS1   CF0   3390-3           2662       141             433
****************************** Bottom of data ******************************
```

Figure 27. CustomPac - Summary of Physical Volumes

However, if the supplied mapping does not fit your installation's DASD plan then you can also customize them here so that they do. Suppose you decide that you would like the element BDT to be moved to a different physical volume. We can see in Figure 28 that the logical volumes associated with BDT are DLB009 and RES009. If you now issue a SUML command you can see the mapping of all the logical volumes to the physical volumes on your system (see Figure 29). An Assign command at this point will allow the logical volume to be mapped to a physical volume of your choice in your configuration.

```
CPPP6054  ------------- Modify System Layout ( DF031072 ) ---- Row 1 to 2 of 2
COMMAND ==>                                                   SCROLL ==> CSR

Logical Volume By PRODUCT

PRIM Cmds:(? SET L F N P SORT)
LINE Cmds:<Assign Dslist>

PRODUCT  : MVS/BDT VER 2 REL 1 BASE

   S  LVol       CYLs     Largest     Datasets          PVol       DEV name
   -  ------     ----     -------     --------          ------     --------
      DLB009       11         4           6           OS3DL1     3390-3
      RES009        9         3           6           OS3RS1     3390-3
****************************** Bottom of data ******************************
```

Figure 28. CustomPac - Logical Volume by Product

```
CPPP6056  ------------- Modify System Layout ( DF031072 ) - Row 54 to 67 of 79
COMMAND ==>                                                   SCROLL ==> CSR

SUMMARY Of Logical Volumes

PRIM Cmds:(? SET L F N P SORT)
LINE Cmds:<Assign Dslist>

               S  LVol                PVol    CUU   DEV name
               -  ------              ------  ---   --------
               a  RES009              OS3RS1  CF0   3390-3
                  RES010              OS3RS1  CF0   3390-3
                  RES011              OS3RS1  CF0   3390-3
                  RES012              OS3RS1  CF0   3390-3
                  RES013              OS3RS1  CF0   3390-3
                  RES014              OS3RS1  CF0   3390-3
                  RES015              OS3RS1  CF0   3390-3
                  RES017              OS3RS1  CF0   3390-3
                  RES018              OS3RS1  CF0   3390-3
                  RES019              OS3RS2  FF0   3390-3
                  RES020              OS3RS1  CF0   3390-3
                  RES021              OS3RS1  CF0   3390-3
                  RES023              OS3RS1  CF0   3390-3
                  RES025              OS3RS1  CF0   3390-3
```

Figure 29. CustomPac - Summary of Logical Volumes

Other possibilities for modifying your system layout can be found in *CustomPac Installation Dialog Reference Manual*.

Alias: The next step is to specify the catalog data set name for each alias. The catalog name will initially be shown as '????????' because it is not yet defined. You can also define whether a catalog is the master catalog by using the Mcat or NoMcat line commands.

SSA: Data sets shipped with your order need to be installed by a driving system. Many data sets meant for the TARGET system will also exist in their own right on the driving system. This Dialog defines SSA's (system specific aliases), allowing the driving system to access the target system catalogs without danger of destroying data sets on the driving system. See section 3.5.2, "System-Specific Aliases" on page 63 for a description of system specific aliases.

Installation: The installation job stream shipped with your order consists of control information (in ISPF tables) and JCL that can be customized (in ISPF skeletons). This dialog allows you to sequentially tailor and submit each job (see Figure 30). Each job is built from predefined skeletons, taking into account all variables you have previously set. User-defined jobs can be set up for any recurrent activity. These jobs can be post installation tasks, additional customization, your own IVPs, migration jobs or any customer specific customization.

This is the most utilized panel during installation, because from here the "real" installation is performed. Note that the output for each job can be viewed from this panel and is stored into a permanent data set.

The actual data sets are downloaded by a RESTORE job using IEBCOPY and IEBGENER. This job may take some time to run. If caching and/or DASD FastWrite are available, then this time can be reduced significantly.

```
CPPP6121  -------------- Installation JOBs ( DF031072 ) --- Row 4 to 17 of 149
COMMAND ==>                                                  SCROLL ==> CSR

JOB Selection List                                           SS$( EXCLUDE )

PRIM Cmds:(? SET L F N P SUMmary SS$ VAREdit)
LINE Cmds:(Backup Delete Edit Insert Log Output Select SS-block Vars)

S      Description                                 STEP     MC STAtus      RC
-- --- ------------------------------------------  -------- -- ----------  ----
   ==> PACKAGE SPECIFIC INSTALL
   DOC PACKAGE INSTALLATION STEPS
   JOB OFFLINE INIT OF DASDS                      OFFLINIT 00 JOB10444    0000
   JOB DEFINE VVDS DATASETS                       DEFVVDS  00 JOB10446    0000
   JOB COPY SISBCIE TO AUTHORIZED LIBRARY         COPYSIS  04 JOB10447    0000
   JOB DEFINE CATALOGS                            DEFCAT   00 JOB10448    0000
   JOB DEFINE SYSTEM SPECIFIC ALIASES             DEFSSA   00 JOB10449    0000
   JOB ALLOCATE AND CATALOG DS                    ALLOCDS  00 JOB10450    0000
   JOB RENAME DS TO FINAL NAME                    ALTCAT   00 JOB10453    0000
   JOB COPY DATASETS TO SYSTEMPAC VOLS            RESTORE  04 JOB10464    0004
   JOB ALLOCATE SMP CSI'S                         SMPALOC  00 JOB10525    0000
   JOB LOAD CSIS FROM RIM TAPE                    LOADCSI  00 JOB10532    0000
   JOB UPDATE NEW SMPE ZONES                      UPDCSI   00 JOB10533    0000
   JOB UPDATE NEW SMP/E DDDEFS                    UPDDDD   04 JOB10534    0004
```

Figure 30. CustomPac - Job Selection List

Save: After installing your ServerPac you can save the configuration you have used. This means that you can re-use the information when you install future orders, reducing the amount of manual input on any further CustomPac installs.

4.1.4 ServerPac System Layout

The installation is now complete and your OS/390 system is ready to be IPL'd. Figure 31 illustrates where the ServerPac selects to place key system data sets.

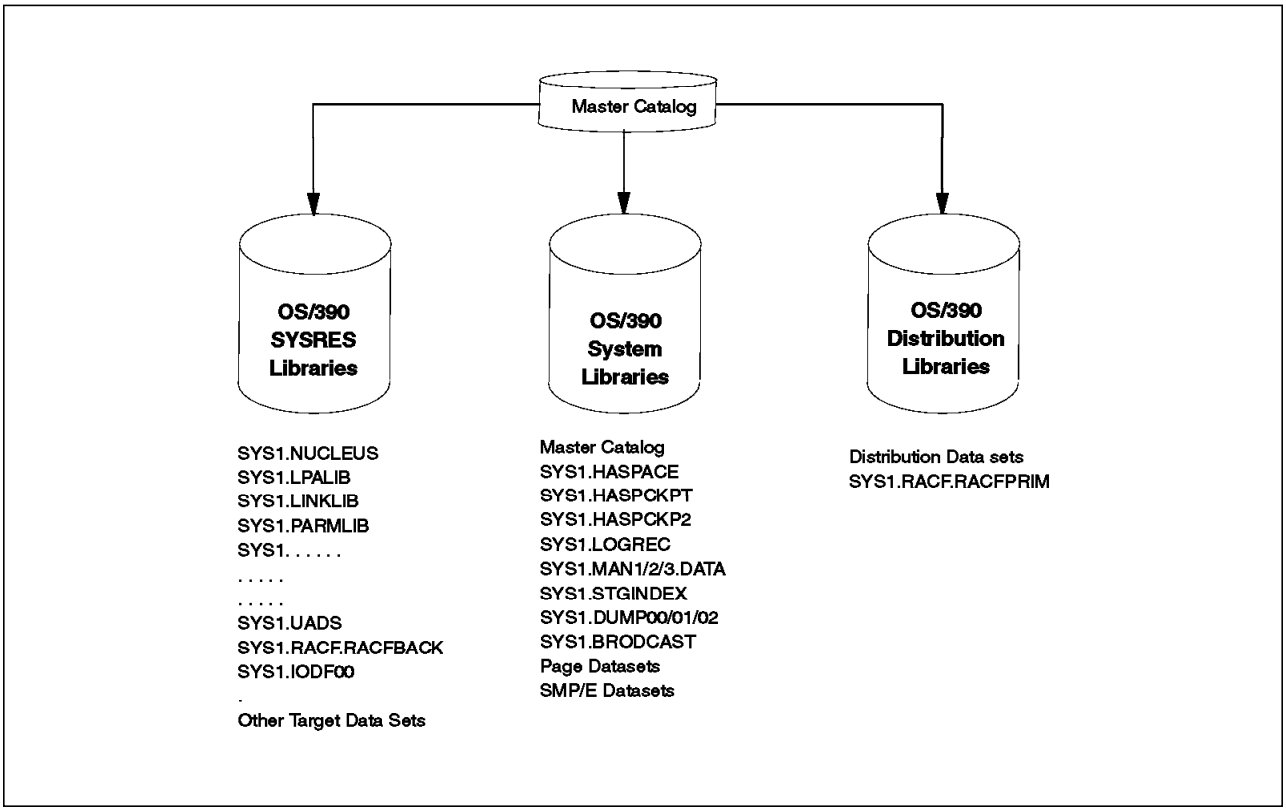


Figure 31. OS/390 System Layout As Supplied by the ServerPac

4.1.5 Post-Installation Tasks

The section lists some additional tasks that you may want to do upon completion of the ServerPac installation job stream.

- Move ISV product data sets back to the SYSRES volume as required.
- Re-install any ISV components that intersect with OS/390 components.
- Remove ServerPac system data sets from the SYSRES: (for example, SYS1.IODF00).
- Perform cross-zone SMP/E IFREQ check, comparing the ServerPac-supplied target zone (the upgraded maintenance level) with the production maintenance level target zone. This will allow you to keep the maintenance levels of different versions of a product synchronized. This check will highlight any maintenance that has been APPLYed to the production environment but for which the equivalent requisite PTF for an associated FMID has not been shipped with the ServerPac. The following extract of a PTF illustrates this:

```
++ PTF (UW24498)                                              .
++ VER (Z038)
   FMID(HBB5520)
   PRE  (UW24306,UW23948,UW21062,UW18643,UW16808,UW16731)
   SUP  (UW21779,BW17417,BW16233,AW17417,AW16630,AW16478,
         AW16477,AW16476,AW16295,AW16233,AW16232,AW16230,
         AW15289,AW15283,AW15281)                             .
++ IF FMID(HBB6601) THEN REQ (UW24499)
```

If UW24498 is APPLYed in the production environment for FMID HBB5520 but UW24499 has not been shipped with ServerPac supplied FMID HBB6601, then a cross-zone IFREQ check will flag this. The missing PTF should be RECEIVEd and APPLYed to the ServerPac-supplied environment.

The SMP/E REPORT CROSSZONE command will perform this check for zones in the same global zone. The ServerPac-supplied job CHKIFREQ can be used for IFREQ checks across zones that are not under the same global zone. This job builds packed copies of the zones and runs a comparison against the two packed zones. See the *ServerPac Installation Guide* for details of the CHKIFREQ job.

It is possible to set up a dummy global zone to point to all the zones that you want to report on. This global zone does not contain any HOLDDATA or SYSMOD information. It is simply used to tie together zones that are controlled by other global zones. Refer to the section “Using REPORT CROSSZONE with Zones Controlled by Different Global Zones” in the *SMP/E Reference* for information on how to set up a dummy global zone.

- Run an SMP/E REPORT SYSMODS command to list any maintenance in the production maintenance environment that is not in the upgraded maintenance environment. This report is only available for zones under the same global zone. The sysmod comparison report should be researched to determine which sysmods you want to RECEIVE and APPLY to the upgraded maintenance environment.
- Run an SMP/E LIST USERMODS report against the production maintenance target zone. It will probably be necessary to update USERMODS to reflect new FMID and maintenance levels. These reworked USERMODS should be RECEIVEd and APPLYed to the ServerPac SMP/E environment.

4.1.6 System Customization

Very often a system replacement introduces not only products with a higher maintenance level, but also products that are new to your installation. Among other items, parameters, procedures, data set structures, and formats may have changed. Products in the system replacement may be new release levels or even new versions. ISV products may also have been upgraded for compatibility reasons. All of these may require long customization times.

The customization part of a system replacement is generally the most time consuming exercise. For a more in-depth look at this topic, see 6.3, “Customizing the New System” on page 142.

4.1.7 Testing

The ServerPac has been IPLed and IVP (Installation Verification Program) tested during production. It will not be shipped until all IVPs have been executed successfully. The software is fully integration tested before being made generally available.

To verify that the OS/390 system has been correctly installed, you should make at least the following minimal checks:

- Initialize the system.
- Initialize JES.
- Submit a job.
- Check the job's output.
- Logon to TSO/E.
- If CICS or IMS is installed, initialize a region and signon to a terminal.

You may also like to consider the following more complete checks:

- Run critical production jobs.
- Support a representative interactive workload.
- Communicate with all networks.
- Test critical functions in applications.
- Check for completeness of accounting records.
- Test all non-IBM product functions.
- Ensure that performance goals stated in service level agreements can be met.

Refer to Chapter 5, "Testing" on page 123 for a more complete discussion of test objectives, categories, and recommendations.

4.2 System Upgrade

As discussed in Chapter 2, "Maintenance Philosophy" on page 9, periodic system upgrades may currently be the simplest and least disruptive way of updating your system. The key reason is that both pre-installation planning and post-installation customization have to be done only for the products that are being updated. If you perform a system replacement, products may have to be customized whether they are new levels of the product or copies of existing levels.

However, to benefit from the integrated component testing that is performed for each OS/390 release, the system replacement method using a vehicle such as ServerPac is probably the direction to aim for. This is not an insignificant amount of work and for shops that currently use an upgrade method and have many products already at the equivalent OS/390 level a phased upgrade path is recommended. See section 4.3, "OS/390 Upgrade Path" on page 102 for specific OS/390 upgrade information.

This section discusses techniques to simplify the process of upgrading your system.

4.2.1 Upgrade Planning

In section 2.3, “Approach for Keeping Current” on page 17, the activities required to update a system are discussed. A significant advantage of upgrading your system is that the pre-installation planning, cloning, SMP/E work, customization, and testing can all be “batched” together to save time over a less planned approach. In addition, less post-installation customization is required with an upgrade than for a complete system replace.

This section examines the planning that you should perform prior to upgrading your system.

4.2.1.1 Upgrade Management

An advantage of using an upgrade strategy is that you can plan when system upgrades will occur and how long it will take to perform the tasks. This means that you can also plan the people and machine resources that will be required for the upgrade project. In other words, an upgrade of your system should be subject to normal project management disciplines.

It is also important that you have well structured change and problem management procedures in place. Because you will be updating an existing system, it is important to record:

- What changes are made to the system.
- How each change might impact system operation.
- What action should be taken to recover from a problem introduced by any change.

If a problem does occur, facilities must be in place to:

- Record that the problem occurred, and obtain diagnostic information such as the symptoms of the problem.
- Direct the problem report to the appropriate personnel for resolution.
- Track the progress of the problem resolution process.
- Close the problem when it is resolved.
- Report on the number and types of problems encountered, number resolved, outstanding problems, and so on.

4.2.1.2 Pre-Installation Planning

When planning for an upgrade of your system, there is a great deal of information available from IBM. It is important that this material be reviewed in detail. In general, the more time you spend planning the upgrade, the less time you will spend on installing and testing the new level.

Some basic preliminary planning has to be done, such as to determine:

- How much DASD space will be needed to hold:
  - Program code
  - SMP/E tracking information for that code
  - JCL, parameters, and samples for the new functions
- What the installation tasks are
- When each task has to be performed
- What the impact will be on ISV products
- How will the upgrade affect user exits
- What changes will be made to the current operating environment:

  - Changed operating procedures that affect system operators
  - Changed application environment that affects applications programmers
- Whether the maintenance environment is set up so that it is easy to install and implement the changes.

Information to help with this planning is found in IBM documentation that is available to you:

- The *Program Directory* describes the software product, what comes with the package, how the product is supported, and so on. The Program Directory:
  - Lists where to go for more information about the product installation.
  - Directs the installer to an installation manual (if available) for the product.
  - Lists DASD space requirements for the product.
  - Gives installation instructions. Note that some products have a separate installation manual.
  - May list the SMP/E statements used to package the product.

  If a CBPDO is ordered that contains products, all the product program directories are shipped in machine readable format on the CBPDO tape.

  If a ProductPac is ordered, most of the above information has already been researched for you. The ProductPac build process requires that a product must be integrated into the process prior to being available for production. This integration requires that the essential information from the Program Directory be considered. DASD space requirements are calculated from the real module size in the database. SMP/E required statements and jobs are extracted and tailored to your CSI, and other product installation instructions are placed in the Installation Guide. In fact, there is no need to research the Program Directory, although it is also shipped in machine readable format on the RIM tape.
- ***Preventive Service Planning (PSP)*** files exist for both products and service. These files should be considered as updates or corrections to the Program Directory:
  - If a CBPDO is ordered, the PSP file is automatically included on the CBPDO tape.
  - If a CustomPac is ordered, the contents of the PSP is considered and automatically included in the package.
- ***Related Installation Materials (RIMs)*** are shipped with the different packages.
  - For CBPDO, the RIMs contain instructions unique to this package. The *CBPDO Memo to Users Extension* describes what is included in your order. The CBPDO RIMs describe how to receive the tape.
  - For CustomPac, the RIMs contain installation instructions unique to your system including all required information to RECEIVE, APPLY and ACCEPT the package. The Installation Guide describes what is included in your order. In terms of CustomPacs, “Related Installation Materials” are extended to include prebuilt or predefined jobs, plus customized product information from all the above sources.
- Some elements, such as DFSMShsm and RACF, have installation and customization manuals that should also be read prior to the install.

When care is taken with planning, less time is required to perform the upgrade.

4.2.2 Cloning

You should never directly upgrade products and service on your production system. Rather, you should install the changes on a copy of your production system, test the changes, and then migrate them into production. This process minimizes the risk of new code causing an outage of your system. The cloning process, and techniques for taking a copy of your system, are discussed in detail in Section 3.7, “Cloning an Existing System” on page 71.

4.2.3 Keeping a Log

When upgrading your system, change management is required to record what has been altered. This was discussed in section 4.2.1.1, “Upgrade Management” on page 90. An important part of your change management procedures should be to keep logs of the upgrade processing. The data that should be kept includes:

- The JES job log for each install step.
- The output from the SMPRPT, SMPOUT, and SMPLIST data sets for each install step. The ddnames used are:

  SMPRPT contains the reports produced by each command.

  SMPOUT contains the messages produced by each command.

  SMPLIST contains the output from the LIST command.
- An SMP/E log of the processing that occurred. Each zone should have its own log. Each of the SMP/E commands that you execute records what it does to the SMPLOG ddname. See *SMP/E Users Guide* for more information.

4.2.4 CBPDO Installation

This section provides a summary of the CBPDO installation process. Further information can be found in *Custom Built Offerings Planning and Installation*. This documentation is supplied with a CBPDO order.

To install any of the CBPDO features, follow these general steps. Make sure you have already installed SMP/E (the minimum required level is 1.8.1). You can use the SMP/E dialogs to load files from the CBPDO tapes and install your CBPDO:

1. Check the CBPDO package against the program shipping request to make sure you received all materials shipped with the CBPDO.

   Note: Some product materials, such as basic publications, are shipped separately from the CBPDO tape.
2. Review the CBPDO Memo to Users Extension for the sample job to load the RIMLIB data set from tape to DASD.
3. Modify, if necessary, and run the job (from the RIMLIB data set) to load the CBPDO Memo to Users Extension from tape to DASD, or run the job to print it.
4. Review the CBPDO Memo to Users Extension which includes:
   - Information on the programs and PTF service that are shipped on the CBPDO tapes
   - Deviations from the product installation described in the program directories

5. Modify, if necessary, and run the job (from the RIMLIB data set) to load the program directories and PSP information from tape to DASD, or run the job to print the program directories and PSP information. (Program directories are supplied on the CBPDO tape only if products were ordered in the CBPDO.)
6. Review the program directories for additional information about installing the products. Also check any PSP upgrade and subset files for the products along with the PSP for the PDO itself in upgrade SSCCINFO or CORPE (varies between countries). These upgrade files contain installation considerations that may be helpful.
7. If you are installing SMP/E from CBPDO tapes, also install all the associated SMP/E PTF service provided on those CBPDO tapes. The sample jobs provided in the RIMLIB data set help you do this.
8. Modify, if necessary, and run the job from the RIMLIB data set to receive programs and PTF service from the CBPDO tapes into your SMP/E PTS data set.

Once the products and service have been received into the SMPPTS, you use traditional installation methods to install the products and service. A CBPDO does not provide RIMs to assist in these tasks. You research, create, and run the required installation jobs that you define from program directories, PSP upgrade files, existing installation procedures, product installation and customization guides, or SMP/E manuals for the programs and PTF service you want to install. Exceptions to the program directories are documented in the CBPDO Memo to Users Extension.

The *CBPDO Memo to Users Extension* for the feature you order contains a customized description of the contents of your CBPDO package and the products contained in that package.

The description of the CBPDO package includes:

- A list of products by name, order number, and feature code
- A list of the copyrighted licensed programs in the package
- Volume serial numbers of the tapes contained in the package
- A list of the FMIDs contained in the package
- The PTF service levels contained in the package
- Driving system dependencies for the installation process

Once you have installed products and service from your CBPDO, you may still have to do some of these post-installation tasks:

- Do any product-specific customization needed to make your system fully operational.
- Install any necessary user exit routines.
- Tune the system to your requirements.
- Perform system-level testing as required.
- Activate the new system as a production system.

4.2.5 CustomPac Installation

This section discusses how to upgrade an SMP/E product set (SREL) in the maintenance environment described in Chapter 3, “Maintenance Environment” on page 39, while still maintaining the old level.

We assume the upgrade, either products or maintenance, is done using a CustomPac.

ServicePac is used as an example when examining the approach for upgrading a system.

Table 5 shows the typical installation path required to upgrade your system. It was taken from a ServicePac installation dialog and identifies all the tasks and work items that must be completed for a successful installation.

Table 5. ServicePac Installation Tasks		
Jobname	**Statement of Work**	**Description**
	====> Package Installation	
	DOC: Installation Steps	Section 4.2.5.1, “Pre-Installation”
PPS0OPT	JOB: UPDATE UTILITY/OPTION ENTRIES	
PPS0UNLD	JOB: UNLOAD SERVICEPAC FILES	
	DOC: Library Space	Section 4.2.5.2, “Library Space” on page 96
NSPCD	JOB: CHECK DLIB SPACE (CLIST)	
REXSPCD	JOB: CHECK DLIB SPACE (REXX)	
NSPCT	JOB: CHECK TLIB SPACE (CLIST)	
REXSPCT	JOB: CHECK TLIB SPACE (REXX)	
DLIBJ1	JOB: REALLOCATE DLIB DATA SETS	
TRGTJ1	JOB: REALLOCATE TARGET DATA SETS	
	DOC: Receive PTFs/FMIDs	Section 4.2.5.3, “Receive PTFs” on page 96
PPS0REC	JOB: RECEIVE INSTALLABLE PTFS	
PPS0REC1	JOB: RECEIVE ALL PTFS	
	DOC: Required Actions	Section 4.2.5.4, “Required Action Information” on page 96
	DOC: System Holds	Section 4.2.5.5, “Reports Requiring Action” on page 97
	DOC: PTFS Requiring UCLIN	Section 4.2.5.7, “PTFs Requiring UCLIN” on page 100
PPS0UCL	JOB: PTFS REQUIRING UCLIN	
IOACCT	JOB: ACCEPT FULLGEN PTFS	
PPS0STG1	JOB: STAGE 1	
PPS0JCLI	JOB: JCLIN	
	===> Pre-APPLY	
	DOC: SMP/E APPLY	Section 4.2.5.8, “SMP/E APPLY” on page 100
PPS0APPL	JOB: APPLY PTFS	
	===> Post Installation	
	DOC: SMP/E ACCEPT	Section 4.2.5.10, “SMP/E ACCEPT” on page 100
PPS0ACCT	JOB: ACCEPT SERVICE (ALLPTFS)	
	DOC: SMP/E CLEANUP	Section 4.2.5.11, “Post ACCEPT” on page 100
PPS0REJE	JOB: REJECT ALL PRODUCTS AND SYSMODS	

4.2.5.1 Pre-Installation

The ServicePac is installed using the installation dialog supplied with the package.

All jobs that are used to install this ServicePac contain variables that you must set. Refer to 4.1.3.1, “Getting Started” on page 79 for a description. A job is supplied to prepare your SMP/E environment (update UTILITY and OPTION entries). Another job is supplied to unload all ServicePac files from the tape.

Once all control and report data sets are downloaded, you can start and use the dialog for the installation.

4.2.5.2 Library Space

To prevent space problems (ABENDx37) during the APPLY and ACCEPT steps in the installation process, a REXX EXEC and TSO CLIST are provided. If your driving system′s TSO/E is at Version 2 or higher, you may use the EXEC; otherwise, use the CLIST.

Both procedures (REXX or CLIST) will read all DDDEFs from your distribution and target zone to verify that these data sets have enough space. If the procedure finds libraries that are more than 70% filled, then reallocation jobs will be built. One or more reallocation jobs may be generated. Submit the necessary reallocation jobs from the installation dialog. If you do not increase these data sets, you will have a greater chance of running out of space during the APPLY or ACCEPT processing for these data sets.

4.2.5.3 Receive PTFs

The Installation Guide indicates how many installable PTFs are shipped with this ServicePac. During production of the ServicePac, all available PTFs are researched, but not all available PTFs are installable. CustomPacs are built to an IBM recommended service level (ISL); therefore, a select list is used, containing all the installable (ISL) PTFs only. The select list does not include the uninstallable PTFs. All PTFs beyond the ISL are called reach-ahead service and are excluded from the select list unless they fix HIPER APARs or PTFs in error.

Notice that one file on the service tape shipped with the CustomPac contains all PTFs that are available at production time, including reach-ahead, installable service, and uninstallable service. The SMP/E RECEIVE statement is concatenated with the select list for all installable PTFs only.

It is important to use the HOLDDATA file supplied on the same tape, because the APPLY/ACCEPT run is based on these HOLD statements. ++RELEASE statements are created for APAR HOLDs that should be bypassed, which means that if you RECEIVE different HOLDDATA than that which is shipped in the package, the APPLY/ACCEPT install jobs may not work correctly.

Job PPS0REC is used to RECEIVE all installable PTFs and HOLDDATA. Job PPS0REC1 can be used to RECEIVE all available PTFs, including reach-ahead, installable service, and uninstallable service.

4.2.5.4 Required Action Information

Some actions defined in this section must be performed prior to the APPLY of this package. The report section of each Installation Guide contains two categories of reports:

- Reports requiring action
- Informational reports

Figure 32 shows all reports that are part of the Installation Guide shipped with each CustomPac.

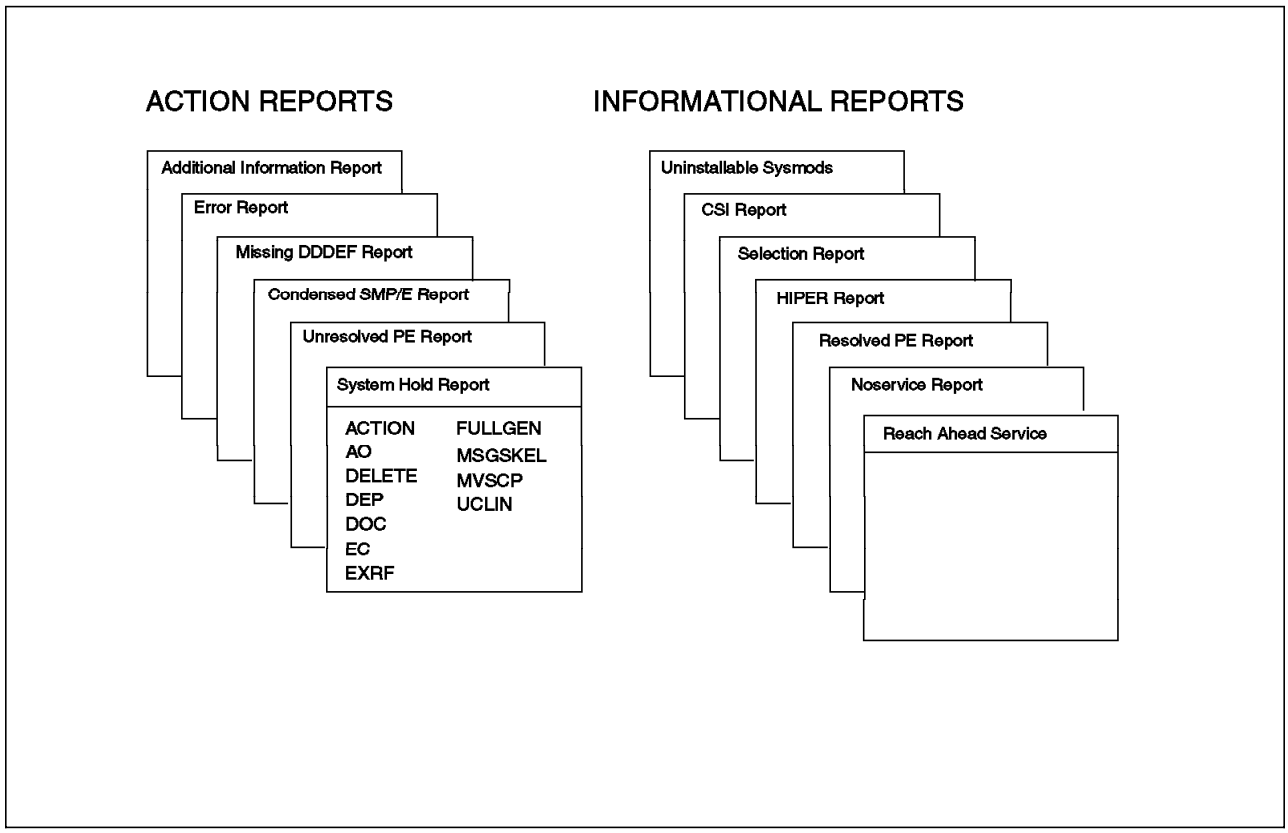


Figure 32. CustomPac Supplied Reports

4.2.5.5 Reports Requiring Action

The following reports may require action on your part either prior to, during, or after the installation of this package:

ERROR If there are any SYSMODs applied or accepted incorrectly on your system, they will have the error indicator on. The Error Report lists all those SYSMODs that are incorrectly installed.

These errors must be corrected prior to the installation of this package by reinstalling the SYSMODs to reset the error indicator. Otherwise all other PTFs selected on top of this SYSMOD will not be installed by SMP/E.

DDDEF During production of this package, a COMPARE was performed between the elements (for example, MAC, MOD, SRC) and the DDDEFs contained in the CSI in order to detect missing DDDEFs. An entry in the “Missing DDDEF Report” is created for each element that points to a non-existent DDDEF.

The report shows a summary of missing DDDEFs in your CSI. It is recommended that these DDDEFs be defined prior to the installation of this package.

AI This report lists any PTFs included in this package that require special attention or corrective action during the installation process. The description includes the activities required.

IBM Change Teams include Additional Information (AI) in PTFs whenever special handling or corrective action is required. This information is included in the AI Report. Review this report and perform the required actions.

CONDENSED SMP The condensed SMP Report lists any SMP/E warning messages that describe SYSMODs that will be regressed as a result of this package installation (for example, USERMODS, APAR fixes, and so on). These SYSMODs will have to be researched for possible retrofitting and reinstallation. Any other SMP/E warning or error messages that occurred during the APPLY CHECK or ACCEPT CHECK of the selected service that you should know about will also be in this CONDENSED SMP Report. Note that all condition codes greater than 4 will be resolved. Any USERMODS that are targeted to be regressed will be listed here.

The complete SMP Report is contained in a data set on the RIM tape. SMP/E SMPEOUT is also listed.

UNRESOLVED PE This report lists PE PTFs installed on your system that do not have an installable correction. The report includes the affected FMID and the APAR number describing the PE situation. The APAR descriptions are documented in the RIM tape file.

APAR This report contains a list of APARs that describe unresolved PE PTFs installed on your system or PE situations causing uninstallable SYSMOD chains. You should carefully read the UNRESOLVED PE and APAR reports, because they show known errors that may affect your system. We recommend that you examine these descriptions to see if there is a possible impact to your environment.

SYSTEM HOLD The SYSTEM HOLD Report in your installation guide contains a list of PTFs that require special attention. The report is the result of IBM′s APPLY run, and contains only the SYSTEM HOLDs for those PTFs that will be applied.

The SYSTEM HOLD Report shows all affected PTFs with imbedded system holds, sorted by the different HOLD reasons. Some actions (UCLIN/ACTION) must be performed before the APPLY/ACCEPT steps, and others (ACTION, IOGEN) are performed after the APPLY run. Normally the information about whether an action is required before or after the APPLY/ACCEPT step is contained in the PTF cover letter. The cover letters and hold information for these PTFs will be supplied in a data set on the RIM tape. Read this report and perform the necessary actions for the following documented HOLD SYSTEM REASONS that may be present:

ACTION Determine whether the special handling of these PTFs is applicable.

AO Determine whether the changes introduced by these PTFs will affect automated operations.

DELETE Review the load modules or elements that will be deleted from the system.

DEP Determine whether the software dependency is applicable by reviewing the PTF cover letter and ensure the condition is met.

DOC Read the cover letter and note the effects of the documentation change.

EC Determine whether the Engineering Change level of your installation hardware must be updated.

EXRF Applies to Extended Recovery Facility (XRF) users only. PTFs must not be applied to the active and alternative systems independently.

FULLGEN Determine whether a SYSGEN is required. ServicePac will supply special installation instructions and jobs.

IOGEN Requires a system or subsystem I/O generation.

MSGSKEL Indicates that the SYSMOD contains message changes that must be compiled in order for translated versions of the message changes to become operational on extended TSO consoles.

MVSCP Requires that the MVS Configuration Program be run to incorporate the configuration changes.

4.2.5.6 Informational Reports

The following reports provide information and require no user action.

CSI This report list SYSMODs installed in the CSI and SYSMODs selected in this order for installation.

SELECTION This report identifies all PTFs that are included in the ServicePac SMP/E select lists provided with this package.

HIPER This report list PTFs to be installed with this package that resolve High Pervasive APARs.

RESOLVED PE This report lists installed PE PTFs in your SREL. The correction is available and could be installed with this package.

NOSERVICE This report lists FMIDs in your target/DLIB zone set for which there is no new service available for your system.

AHEAD This report lists PTFs higher than the base service level selected to resolve HIPER APARS or PTFs in error.

UNINSTALLABLE SYSMODs This report lists any uninstallable SYSMODs that are included as part of the service tape. A SYSMOD chain may be uninstallable because of a PE or because a requisite PTF in the chain is unavailable. You may want to perform an SMP/E RECEIVE into your SMPPTS, as they may be used for corrective service. Refer to Section 4.2.5.3, “Receive PTFs” on page 96 for a brief description.

4.2.5.7 PTFs Requiring UCLIN

Unless otherwise stated in the PTF cover letter, any UCLIN steps must be performed before the APPLY/ACCEPT run. All required UCLIN jobs are researched during CustomPac production and provided in a data set called SMPCNTL. Member UCLIN includes the customized UCLIN for all installable PTFs within this package. Execute the job (PPS0UCL) to perform the UCLIN run.

4.2.5.8 SMP/E APPLY

As part of the build process for a ServicePac, IBM performs an SMP/E APPLY CHECK for all selected PTFs using the CSI target zone you submitted.
As part of the build process for a ProductPac, IBM performs an SMP/E APPLY for all selected products using a copy of the CSI target zone you submitted. The output of each run is placed in the CONDENSED SMP REPORT.

You may want to upgrade your system selectively. Again, select lists are used to split the installable service into different members. Those members reside in a control data set called SMPCNTL. This data set contains a specific member for each product (FMID) and a member ALLPTFS. Each specific member lists all PREs, COs, and IFREQs to APPLY all the PTFs in that member. ALLPTFS lists all the PTFs that need to be applied to reach the IBM recommended service level (ISL).

Job PPS0APPL from the dialog is used to apply the service. You may install all the PTFs in one run using the select list, ALLPTFS, or you may install PTFs selectively by FMID. In this case, the same job is used, but you choose the appropriate select list from SMPCNTL for the products you wish to upgrade.

If you install a ProductPac, an additional select list and jobs for each new function will be supplied.

4.2.5.9 Post APPLY

Once the APPLY has run successfully, perform any actions such as relinking of modules, JCLIN, or any other actions indicated by the SYSTEM HOLD or the ADDITIONAL INFORMATION reports.

4.2.5.10 SMP/E ACCEPT

When your system has been tested to your satisfaction, an SMP/E ACCEPT CHECK and ACCEPT should be done (See Chapter 5, “Testing” on page 123 for more information on testing). IBM has performed the ACCEPT CHECK only for installable FULLGEN/IOGEN PTFs. If you choose not to run the ACCEPT, any future SYSGEN activity may lead to the loss of the service you have just installed. The SMP/E ACCEPT uses a SELECT list statement to select ALLPTFS. It is recommended that you use this select list for the ACCEPT job. The supplied ACCEPT job is called PPS0ACCT. Once the ACCEPT check has run successfully, remove the ′Check′ from the SMP/E ACCEPT statement in job PPS0ACCT and rerun the job.

4.2.5.11 Post ACCEPT

If you want to clean up the global zone, SMPPTS, SMPMTS, and, SMPSTS entries after the installation of the ServicePac, job PPS0REJE will perform a SMP/E REJECT to delete the global zone and the appropriate SMPPTS, SMPMTS, and SMPSTS entries that were created during the installation.

4.2.6 Maintenance Testing

Maintenance testing ensures that newly installed PTF service has introduced no known, critical errors into the system. This testing should be standard practice after a system upgrade. You can use IBM tools and facilities to reduce the risk of failures impacting the availability of the system. This section discusses some of these situations.

HIPER APARs: Known problems with products are flagged in the PSP file for the product. The file should be read and a check made to ensure that all required service is installed. The PSP file is continuously updated and should be checked periodically for new problems. However, many enterprises find that it is impractical to constantly check PSP upgrades for all their products. Some countries offer a subscription service for critical error information. For example, the SMARTMVS service provides information about High Impact and PERvasive (HIPER) incidents, that can be reported using the SMPEUTIL utility. See Appendix B, “SMPEUTIL” on page 197 for more information. Consult your IBM Representative to determine what is available in your country.

PTFs in Error (PEs): Problems with PTFs are flagged using ++HOLD statements.

There are two types of PTFs in error that you should know about:

- The PE that prevents a chain of PTFs from being installed. If a PTF has a hold against it, SMP/E will not install the PTF or any SYSMOD that is dependent on it. When you next upgrade your system, a PTF to fix the situation should be available and the chain will be installed. See Section 2.5.3.11, “Selective Follow-On Service (SFS)” on page 31 for more information on the selective follow-on service (SFS) offering.
- The PE that introduces a High Impact or PERvasive (HIPER) error into the system. These are initially flagged as HIPER in the PSP bucket for the product. To automate recognition of these errors, SMARTMVS HOLDDATA has a comment field labelling the severity of the error.

  CBPDO attempts to differentiate between a PE with high-impact errors and a PE with low impact errors that are less important than the fix the PTF provides. These low-impact PEs are labelled with a HOLDCLASS(ERREL), and should be installed using the APPLY BYPASS(HOLDCLASS(ERREL)) operand.

++HOLD information should be used to check the service installed on the new system level. Note that you could also check your distribution libraries with this facility if you so desire. To do the check, you can use the SMP/E REPORT ERRSYSMODS command. It performs the same function as SMPEUTIL, except that the report produced is a standard SMP/E format report, and SMARTMVS HOLDDATA is not required. However, no distinction is made between errors (PEs) and HIPER errors. See *SMP/E Reference* for more information on the SMP/E REPORT ERRSYSMODS command.

Use the output of the report to resolve known errors on the system. If a resolution is not available for a PE you have installed, you have to evaluate the risk of the error actually occurring in your system. In general, if the error is flagged as HIPER and no resolution is available, RESTORE the SYSMOD in error. Otherwise, take note of the PE but leave it installed on the system.

Cross-Zone Check: It is likely that you will upgrade different system releases (SRELs) at different times. This may be because you have different business requirements for the different subsystems, or simply because different groups maintain different system releases. In any case, an upgrade to one SREL should be checked to see whether it impacts other SRELs. For example, assume two products exchange information in a control block. If PTF service for one product changes the format of the control block, the other product must be informed, and a PTF installed to recognize the new format.

Usually, PTFs that have this type of impact are labelled with ++IF statements. The ++IF describes what has to be done to the other product if the PTF is installed. PTFs containing an ++IF may also be held by a SYSTEM HOLD, so that you have to deliberately bypass the HOLD to install them.

As the products in the MVS environment become more complex and have more interdependencies, you can expect that the interdependencies will be documented so that SMP/E can manage them. Therefore you will see more ++IF statements in SYSMODs than you do now. You should get into the habit of reporting on cross-zone requirements and installing any missing service.

Always check for cross-zone requirements in your new system level after an upgrade. SMP/E automatically ensures that IFREQs are satisfied within a target or distribution zone. However, it is recommended that each SREL be maintained in a separate target zone and distribution zone. Therefore, you have to execute a cross-zone report to determine whether service applied to one target zone impacts a product in another target zone. The REPORT CROSSZONE command performs this function for zones in the same global zone. By running this report, you could, for example, compare a production target zone to a test target zone, and you could determine whether products are at the same level in the test environment and the production environment. See *SMP/E Reference* for more information on the REPORT CROSSZONE command.

4.3 OS/390 Upgrade Path

Section 4.2, “System Upgrade” on page 89 discussed general system upgrade issues and used the ServicePac as an example of the steps involved in an upgrade. This section discusses some of the issues to be considered if choosing an upgrade method, for example CBPDO, to migrate to OS/390. The steps required to position for subsequent system replace rather than upgrade are also considered. A more detailed discussion can be found in *OS/390 Up and Running!* and the *Program Directory for OS/390 CBPDO Install*.

4.3.1 Staging the Migration to OS/390

IBM is providing an accommodation for those who use the CBPDO method of install and want to stage the migration of OS/390. (There is no stand-alone product tape for OS/390. Use the CBPDO instead. See 4.2.4, “CBPDO Installation” on page 92 for an overview of the CBPDO installation methodology.) By choosing this approach you can run the base elements of OS/390 along with some OS/390 root products at lower levels than those whose functions have been integrated into OS/390. Furthermore, you can install and migrate individual non-exclusive elements separately. (See 1.4, “What Makes Up OS/390?” on page 4 for a definition of OS/390 base elements and features.)

It is IBM′s intent to offer this accommodation only for OS/390 releases 1,2 and 3.

The earlier the releases of your software, the more difficult the migration to OS/390. A migration from MVS V5R2.2 and the related product set is the easiest. If you already have plans to migrate to that level of MVS and related products, you should proceed with those plans. This staged approach is described in more detail in *OS/390 Up and Running!*.

The *Program Directory for OS/390 CBPDO Install* describes a *three wave* approach to installing the OS/390 CBPDO.

1. Wave 1 installs all FMIDs that do not install into an OpenEdition Hierarchical File System (HFS). This includes exclusive elements and those elements required to install other FMIDs. Ensure that the driving system is at the correct level to install the first wave.
2. Wave 2 installs those elements that install into an OpenEdition HFS. Because the driving system requirements of the second wave exceed those required for the first wave, IBM recommends that you use the target system of the first wave as the driving system of the second wave. If driving system requirements for the second wave are satisfied during the install of the first wave, you can combine the first and second waves.
3. Wave 3 installs the OS/390 level of the JES2 or JES3 component. This may be combined with the first or second wave. The driving system requirements for the third wave are the same as for the first wave.

4.3.2 Positioning to Use the ServerPac Method of Installing OS/390

If you have not previously used a system replace method, there are steps you can take to make the installation easier. Most of the work involves separating code that will still be required after installing OS/390 from that supplied with the OS/390 ServerPac. This includes:

- Independent Software Vendor (ISV) products
- IBM non-OS/390 ServerPac products
- User modifications
- User exits
- CSECTs that are linked into other load modules

The goal is to keep OS/390 from overlaying code that you need to bring forward into the OS/390 environment. It is important to identify if the code is really needed. Perhaps its function is redundant or replaced by OS/390 functionality. Once the ServerPac has been installed then you must make sure that the non-OS/390 code is put back in place. This might be achieved by data set concatenation, reinstalling the code or reassembling it.

ISV Products: Check that your existing level of the product supports the OS/390 environment. Ask the Independent Software Vendor how OS/390 affects the use of the product. If possible place the products in their own set of libraries and SMP/E zones. This means that unless an upgrade of the product is required for OS/390 compatibility, you will not have to reinstall it after the ServerPac install.

IBM Non-OS/390 Products: For products that are no longer marketed, make sure they will work with OS/390. If possible place them in their own set of libraries and SMP/E zones. This means that you will not have to reinstall them after the ServerPac install.

For MVS SREL products that are still marketed but not a component of OS/390, such as PSF, SDSF or NetView, make sure that they do not need to be upgraded

to run with OS/390. Such products can either be ordered with and installed with the ServerPac or moved to separate libraries and SMP/E zones and maintained separately.

For non-MVS SREL products, such as CICS, DB2 and IMS, make sure that they do not need to be upgraded to run with OS/390. If an upgrade is needed, order it with the OS/390 ServerPac. You will receive separate ServerPacs for each SREL.

User Modifications: These include:

- User exits
- User SVCs
- Updates to source code
- Zaps
- Changes to ISPF elements, such as panels, and CLISTs

If possible, isolate this code from the IBM code by using a separate library that can be concatenated to the IBM library. Use SMP/E to install all user modifications. Use documented exit points and avoid source code updates. Keep all user modifications in a single data set and ensure they are documented.

Other Considerations:

- SYSRES

  To avoid having to copy libraries from the current SYSRES to the new SYSRES, place on the SYSRES only the libraries that have to be there. Ideally only the ServerPac installed libraries should be on the SYSRES.

- PARMLIB and PROCLIB

  Compare the ServerPac supplied PARMLIB and PROCLIB members with your existing production copies. Make any appropriate changes. Copy any new PARMLIB and PROCLIB members to your production copy and tailor as needed.

  In a Sysplex environment try to share SYSRES, master catalog and system-type libraries (such as the IODF, SYS1.DAE and RACF data bases), between all systems. Use symbolic substitution to reduce the number of system specific PARMLIB and PROCLIB members.

- Catalog Aliases

  A number of elements are shipped with VRM (Version, Release & Modification level) information as part of the data set name. This allows different levels of the same product to run along side each other but can mean having to update other system components, such as CLISTS and logon procedures, to reference the new level. To save such updates, catalog aliases without the VRM qualifier can be set up to point to the appropriate level of libraries. System components can reference the alias rather than the real data set name. Implementation then requires the deletion of the existing alias and the redefinition of the alias to point to the new level. Using a NetView data set as an example:

```
DELETE NETVIEW.CNMCLST                -
       ALIAS                          -
       CATALOG(catname)

DEFINE ALIAS  (NETVIEW.CNMCLST)       -
       RELATE (NETVIEW.V2R4M0.CNMCLST) -
       CATALOG(catname)
```

It can be expected that VRM information will begin to disappear for elements being shipped in OS/390.

- SMP/E BUILDMCS command

  In OS/390 Release 2 the SMP/E element introduces the BUILDMCS command. The BUILDMCS command provides a more automated and less error-prone process for propagating products from one pair of target and distribution zones and libraries to another pair of target and distribution zones and libraries. The command generates the MCS and JCLIN required to reinstall the specified FMIDs. The BUILDMCS command is intended to be used for products that have no intersections with other products. The BUILDMCS command can be used to simplify the process of separating OS/390-supplied code from other code. Additional information can be found in Appendix C, “BUILDMCS Command” on page 199.

4.4 OpenEdition Installation

OpenEdition code comes in a different format from the usual MVS code and requires a specific environment to be installed and to run. For this reason, this section will give you a brief description of the environment and will go through the required steps to install and maintain the environment.

The support for OpenEdition services component (OpenMVS) enables two open systems interfaces on the MVS operating system: an application program interface (API) and, optionally, an interactive shell interface. With the API interface, the programs can run in any environment, for example, in batch jobs brought in through the job stream or submitted by interactive TSO/E users. The programs can request:

- Only MVS services
- Only OpenMVS services
- Both MVS and OpenMVS services

The optional shell interface is an execution environment similar to TSO/E, with a programming language of shell commands similar to the REXX Language. The shell work consists of:

- Programs run interactively by shell users
- Shell commands and scripts run interactively by shell users
- Shell scripts run in batch jobs

OpenEdition support is provided by:

The OpenMVS Component: This component provides OpenMVS services in response to requests from programs and the shell.

The OpenEdition Shell and Utilities: This feature interprets commands from interactive users or from programs, called shell scripts, and requests MVS services in response to the commands.

The OpenEdition Debugger: This feature helps an application programmer debug source programs written in C.

The OpenMVS component consists of OpenEdition System Services that provides services for program and shell, and OpenEdition Application Services to support full screen application, remote login and automount facility. Before you can use OpenMVS services, the OpenMVS System Services (kernel) must be installed and started. This means that you need an OpenMVS address space running in order to complete the installation of the Application Services code. This is because this code has to be installed into OpenEdition MVS Hierarchical File System (HFS) and all accesses to the HFS are controlled by the OpenEdition kernel address space.

The OpenEdition System Services are installed as part of the ServerPac process. To complete the OpenEdition installation, you have to either start OpenEdition on the driving system, or continue the installation on the new OS/390 system once OpenEdition has been started. If you go through the installation process from the driving system where an operational OpenEdition is currently running, you should mount the new ROOT file at a different mount point and, in case of service maintenance, update all the target SMP/E DDDEFs to address this new mount point directory.

There are some customization steps to go through before activating the OpenEdition kernel:

- SMS must be activated to manage the HFS allocation. HFS can be allocated only on SMS-managed volumes.
- APPC/MVS must be customized and started since OpenEdition uses address spaces provided by APPC.
- Security must be provided to enable the OpenEdition Services.

However, the OpenEdition Shell and Utilities and the OpenEdition Debugger are optional features.

4.4.1 Environment Set Up

This section will introduce the step required to set up the required environment to run OpenEdition MVS. However, we will cover only the customization efforts required in order to support OpenEdition.

4.4.1.1 SMS

SMS manages the HFS data sets used for OpenMVS processing.

The hierarchical file system allows you to set up a file hierarchy that consists of:

- HFS files, which contain data or programs. A file containing a load module is called an executable file. Files are kept in directories.
- Directories, which contain files, other directories, or both. Directories are arranged hierarchically, in a structure that resembles an upside-down tree, with the root directory at the top and the branches at the bottom. The root is the first directory for the file system at the peak of the tree and is designated by a slash (/).
- Additional file systems, which are mounted on directories of the root file system or of additional file systems.

To the MVS system, the file hierarchy is a collection of hierarchical file system (HFS) data sets. Each HFS data set is a mountable file system. The root file system is the first file system mounted. Subsequent file systems can be logically mounted on a directory within the root file system or on a directory within any mounted file system. Each mountable file system resides in a hierarchical file system (HFS) data set on direct access storage. SMS manages the HFS data sets and the physical files.

The following steps will help you in customizing your SMS environment to support OpenEdition MVS. If your installation doesn′t run SMS, you should also prepare a minimal SMS environment through the following activities. For additional information on the SMS setup, refer to *DFSMS/MVS DFSMSdfp Storage Administration Reference*, SC26-4920.

1. Allocate the SMS Control Data Sets (ACDS and COMMDS) on shared DASD if running in a multisystem environment.
2. Modify IEFSSNxx, IGDSMSxx and IEASYSxx in PARMLIB to define the SMS subsystem and its configuration.
3. Define the Source Control Data Set (SCDS) with the base definition. Examples and base configuration are provided in SYS1.SAMPLIB in the following members:
   - DFPSSDCR, sample Data Class ACS routine. Optional for OpenEdition support.
   - DFPSSMCR, sample Management Class ACS routine. Optional for OpenEdition support.
   - DFPSSSCR, sample Storage Class ACS routine.
   - DFPSSSGR, sample Storage Group ACS routine.
   - DFPSSIMP, sample job to unload an SCDS with a base SMS configuration.

Now, you need to define the following before activating the new SMS configuration on your system:

1. Define a Storage Class to control allocation of HFS files.

   Here is an example:

```
PROC STORCLAS

/*********************************************************************/
/*  The following ACS routines is managing OpenEdition files.        */
/*  In this configuration, the naming convention for OpenEdition     */
/*  files is hlq = OMVS.                                             */
/*********************************************************************/
/*             Start of FILTLIST Statements                          */
/*********************************************************************/
FILTLIST OMVS                INCLUDE(OMVS*)
/*********************************************************************/
/*             End of FILTLIST Statements                            */
/*********************************************************************/

SELECT
  WHEN (&HLQ= 'OMVS')
      SET &STORCLAS = 'OPENMVS'
  OTHERWISE
      SET &STORCLAS = ''
```

```
  END
END
/*********************************************************************/
```

2. Define a Storage Group to contain the enable DASD to HFS file allocation. Ideally, allocate the user HFS data sets on separate volumes from the system HFS volume to facilitate future system replacement or upgrade. Here is an sample SG ACS routine:

```
PROC STORGRP

/*********************************************************************/
/*  The following ACS routines is managing OpenEdition files,        */
/*  OE hlq = OMVS                                                    */
/*********************************************************************/
/*  Start of Mainline SELECT                                         */
/*********************************************************************/
SELECT
  WHEN (&STORCLAS = 'OPENMVS')
      SET &STORGRP = 'OPENMVS'
  OTHERWISE WRITE 'OPENMVS IS THE ONLY DEFINED STORCLASS'
  END
END
```

3. Initialize the volume belonging to the Storage Group with the SMS indicator.
4. Verify and translate the ACS routines.
5. Activate the new SMS configuration.

4.4.1.2 APPC/MVS

OpenMVS uses address spaces provided by APPC/MVS. If your installation has not been using APPC/MVS, prepare it. Whether you were using APPC/MVS or not, you need to customize it for OpenMVS.

All the APPC/MVS customization jobs are provided by the ServerPac and are located in SYS1.SAMPLIB with the specific prefix BPX. Follow the directions contained in *MVS/ESA OpenEdition MVS Installation and Customization Starter Kit*, SG24-4529.

1. Run the job to create a VSAM KSDS data set for the APPC/MVS transaction program (TP) profile for forked address spaces. Refer to *OS/390 MVS Planning: APPC/MVS Management* for a sample allocation job. Member ATBTPVSM in SYS1.SAMPLIB is a sample allocation job.
2. Run the job to add the necessary Transaction Profile (TP) for processes using kernel services. Sample JCL is in BPXISTPA member in SYS1.SAMPLIB. Refer to *OS/390 OpenEdition MVS Planning*, SC28-1890 to correctly size your TP file.
3. Add a LUADD statement for OpenMVS to the APPCPMxx PARMLIB member. If your installation has not been using APPC/MVS, add an APPCPMxx PARMLIB member containing one LUADD statement. You can find a sample member in IPO1.SPARMLIB.
4. Add a CLASSADD statement for OpenMVS to the ASCHPMxx PARMLIB member. Or, if your installation has not been using APPC/MVS, add an

ASCHPMxx member containing one CLASSADD statement. A sample ASCHPMxx member is in IPO1.SPARMLIB.

5. Start APPC/MVS using the operator command:

```
START APPC,SUB=MSTR,APPC=xx  where xx is suffix to APPCPM member
```

6. Start ASCH using the operator command:

```
START ASCH,SUB=MSTR,ASCH=xx  where xx is suffix to ASCHPM member
```

   At ASCH start up, the following message can appear and can be ignored:

```
ATBO52E LOGICAL UNIT xxxxx FOR TRANSACTION SCHEDULER ASCH
NOT ACTIVATED IN THE APPC CONFIGURATION. REASON CODE = 5A
```

4.4.1.3 Security

In order to start and to be able to use OpenEdition services, you need to define the new resources to the security environment.

RACF or an equivalent ISV security product is necessary to manage system and data security by verifying a user and verifying that a user can access a process or file. This chapter assumes that your OS/390 system uses RACF to provide security.

On UNIX systems, each user needs to be identified by a user name and a password. UNIX uses the /etc/passwd file to keep track of user name and encrypted password for every user on the system. Internally the UNIX system uses a numeric ID to refer to a user. So, in addition to name and password, the /etc/passwd file also contains a numeric ID called user identifier or *UID*. Generally these number are unsigned 16 bit numbers, ranging from 0 to 65535. The UID is the number that the operating system uses to identify the user in all his processes. So, if two users are assigned the same UID, UNIX views them as the same user, even if they have different user names and passwords. Two users with the same UID can freely read and write over each other′s file and can kill each other′s processes. Assigning the same UID to multiple users is generally not recommended.

UNIX systems also have the concept of groups where you group together many users who need to access a set of common files, directories, or devices. Like for the UIDs, groups have both group names and group identification number, known as *GID*. Each user belongs to a primary group that is stored in the /etc/passwd file on UNIX systems.

These UIDs and GIDs are used by OpenEdition to control the files and processes that a user may use. The RACF database contains the user profiles and group profiles. Within these user and group profiles is a new segment called the OMVS segment. The OMVS segment contains the UID and GID definitions for each user and group respectively. The detailed information contained in the OMVS segment of the user profile is:

- The user identifier (UID)
- Initial directory path name (HOME)
- Program path name (PROGRAM)

So, let′s go through the steps required to define these new resources to the security environment:

1. Define Groups for Open Edition MVS

You need to define the *TTY* Group that identifies pseudo-terminals used by some shell programs. When a user logs in or issues the OMVS command from TSO/E, the group name associated with these terminals is changed to TTY. This group should have a unique GID and no users connected.

You can optionally define a specific group where all the OpenEdition users are connected, or you can modify an existing one by assigning a specific GID in the OMVS segment.

2. Define a Superuser

   Once the users have been assigned to an OMVS Group, you have to give a Superuser authority to a specific user. A Superuser is a user who has special rights and privileges. You need to have a Superuser in order to run SMP/E and complete the installation.

3. Define the OMVS Kernel

   The OMVS cataloged procedure runs a program that initializes the kernel address space. This procedure needs to be related to a RACF user ID that is also a Superuser. This specific RACF user ID must be defined into the RACF Started Task Table (ICHRIN03) or into the RACF STARTED class.

For detailed instructions, refer to *MVS/ESA OpenEdition MVS Installation and Customization Starter Kit*, SG24-4529.

4.4.2 Allocation of HFS Root

The ServerPac process provides a ROOT file with all the new level code already installed. So the only step to install the new HFS level is a DFDSS job to unload the file. The job is in the HOT1130C item in the ServerPac dialog.

If you are running the job from the driving system, consider the following:

- The new Root file must be restored with a different name than the current one, if in use.
- Restore the root file on an SMS volume that is shared with the target SMS configuration.
- If you want to test the new code, mount the new Root file on a different mount point on the driving system.

If you going on with the installation from the target system, consider the following:

- Customize your system to run DFSMSdss creating the IFAPRD00 member in SYS1.PARMLIB and activating it via the SET PROD=00 command.
- Restore the Root file on an SMS volume.
- Make it available to OMVS with the ROOT statement in BPXPRMxx member in SYS1.PARMLIB, as described in 4.4.3, “Customize PARMLIB Members.”

4.4.3 Customize PARMLIB Members

OpenEdition customization material is defined in two SYS1.PARMLIB members:

- *BPXPRMxx* contains the parameters that control OpenEdition MVS processing and the file system. On the *ROOT* statement you have to address the Root file name that is mapping the HFS file environment. You are required to set up this member if you are running OpenEdition customization on the target system.
- *CTnBPXxx* specifies the tracing requirements for OpenEdition and is optional.

4.4.4 OMVS Activation

OMVS is the cataloged procedure provided to start the OpenEdition kernel address space.

To start OpenEdition, enter the following command:

```
START OMVS,OMVS=xx
```

where xx is the suffix of the BPXPRM member in SYS1.PARMLIB.

4.4.5 Connection to HFS User Data

In traditional MVS environments, a general user is given his own environment through the usage of an alias and the ability to create data sets under a certain high level qualifier. In OpenEdition MVS, this is also accomplished by allocating separate HFS data sets for each OpenEdition MVS user and mounting these separate HFS data sets on the root file system in the /u directory. These separate HFS data sets can be used to store data unique to this user.

This method allows each OMVS user to control and access his own data without impacting other OMVS users. It is also a way of isolating each OpenEdition MVS user′s data for systems management purposes. Having separated user files from the system makes system replace easier because you do not have to merge or copy the user data to the new environment. The only required activity is to make it available to the new configuration. The next paragraph gives the recommended way to set up this connection.

In the new target OpenEdition environment, you need to get your HFS user data mounted at a mount point off the root directory to make them available. Remember that the best choice to mount all user HFS data sets is the /u mount point.

You can use the current method used on the driving system or one of the following ways to accomplish this definition:

Using the Direct Mount Allocate an intermediate HFS data set to be mounted between the root file system and all user file systems. Create the mount point using the *mkdir* command and issue the mount command. You can also make the mount permanent adding the HFS data set name and its mount point to the BPXPRMxx member of SYS1.PARMLIB. If you used this method on your driving system, to connect the user HFS data sets you are required to mount this intermediate file on the new target system. This is illustrated in Figure 33 on page 112.

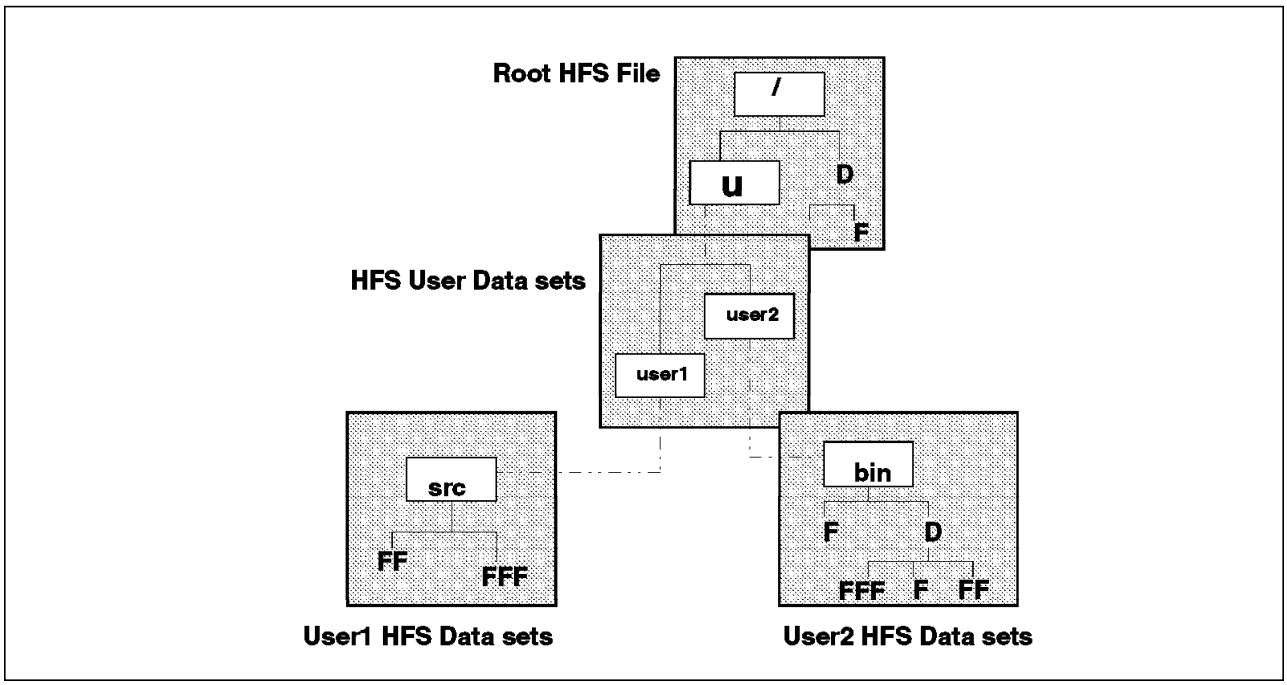


Figure 33. Direct Mount Methodology

Using the Automount Facility Using the Automount facility is the best way to manage HFS data sets because it saves administration time and does not require any activity during system replacement.

The Automount facility, as shown in Figure 34, controls all user file systems to automatically mount them when they are needed.

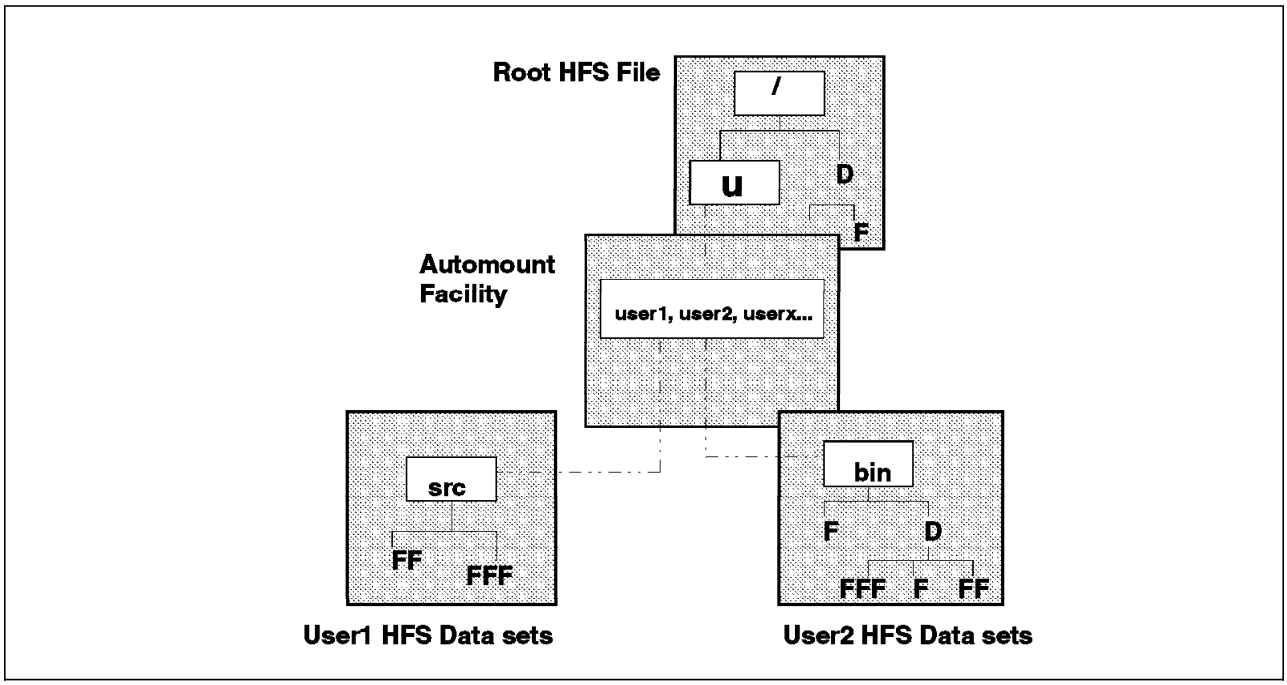


Figure 34. Automount Facility Methodology

Refer to *MVS/ESA OpenEdition MVS Installation and Customization Starter Kit*, SG24-4529 for further details on how to implement these facilities.

4.5 OpenEdition Maintenance

This chapter will discuss a general methodology of how to manage maintenance when OpenEdition is involved. Service for OpenEdition MVS is installed in the same way as it is installed for traditional products. Service is applied using SMP/E.

However, there are some differences:

- The target libraries for OpenEdition MVS are in HFS data sets. SMS must be up and running and managing at least one DASD volume (the volume that contains the OpenEdition MVS target libraries).
- The OpenEdition MVS kernel must be running on the driving system in order to apply service.
- The user ID that is running the SMP/E jobs must have an OMVS security segment and must be running as a superuser.

You should never directly upgrade your driving system. You should install changes on a copy or clone of your production system, test the changes, and then migrate the new level into the production environment.

The following actions are required and summarized in Figure 35.

1. Build the clone root file using DFSMSdss and the RENAMEU keyword in order to obtain a similar environment with different data set names, for example OMVS.ROOT.SERVICE. This new root file (OMVS.ROOT.SERVICE) has to be copied to an SMS- managed volume.
2. Mount the copied root file system into the directory within the active root file, OMVS.ROOT. To be able to mount the new HFS, a mount point directory has to be created. The common mount point directory is called SERVICE and it is a sub-directory of the root.
3. Change the SMP/E DDDEFs to point to the new target directories that are mounted off the /SERVICE directory. For OpenEdition MVS, members FSUMUSDD, FDBXUSDD and FOMISDDD in SYS1.SAMPLIB contain the list of all DDDEFs used by OMVS.
4. Install service with SMP/E. The OMVS Kernel must be active on the driving system because access to the HFS file is controlled by this address space.

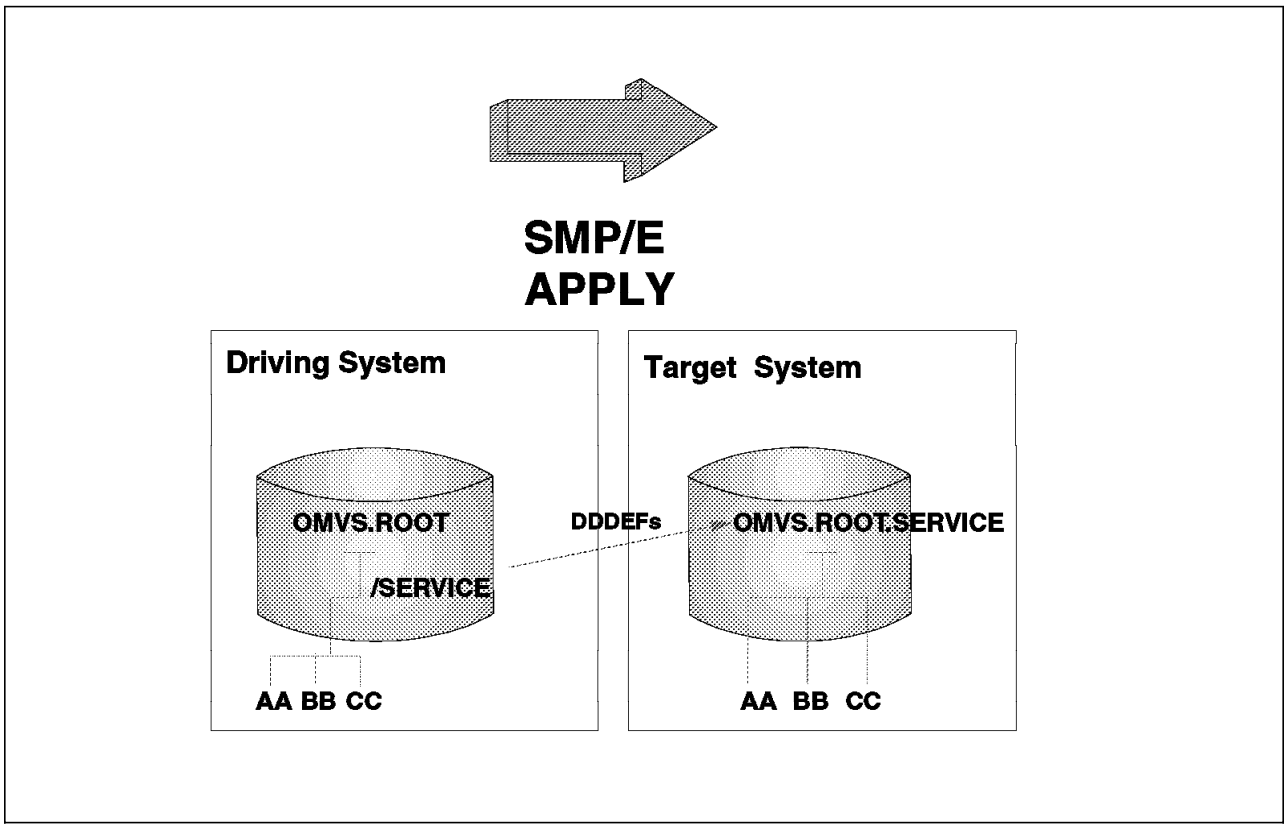


Figure 35. OpenEdition Maintenance Environment

5. Test your new environment. If the service hits SYS1.LPALIB, you have to IPL the system. If the service hits paths in the HFS file system, you have to change the ROOT statement in the BPXPRMxx member to point to the new root file, OMVS.ROOT.SERVICE, and stop/start the OMVS kernel.

Refer to *MVS/ESA OpenEdition MVS Installation and Customization Starter Kit*, SG24-4529 for any further details.

4.6 OpenEdition DCE

A distributed computing environment (DCE) allows several individual systems to share data, processors, applications, and devices and to operate as a single unit. It also allows the individual units that comprise the distributed environment to be on different vendor platforms and running different operating systems. The participating machines can range from personal computers to supercomputers; the network can connect machines in one building or on different continents.

OS/390 OpenEdition DCE is part of the IBM open and distributed strategy for OS/390. In this new OS/390 environment, called OS/390 OpenEdition, applications can get the following benefits:

- Exploit the traditional strengths of OS/390, using an integrated set of open system services
- Exchange data and access services transparently
- Be portable across multivendor operating systems that support open standards
- Be managed, along with their data, across the network

OS/390 OpenEdition combines DCE components and other interoperability-enabling services, forming a comprehensive, integrated, and all-encompassing structure to handle every aspect of distributed computing, from communications to resource management and application development.

The OS/390 DCE can be considered an infrastructure layer towards client/server model application. DCE itself uses services provided by OpenEdition MVS. Figure 36 is a view of the interaction between the different components and layers.

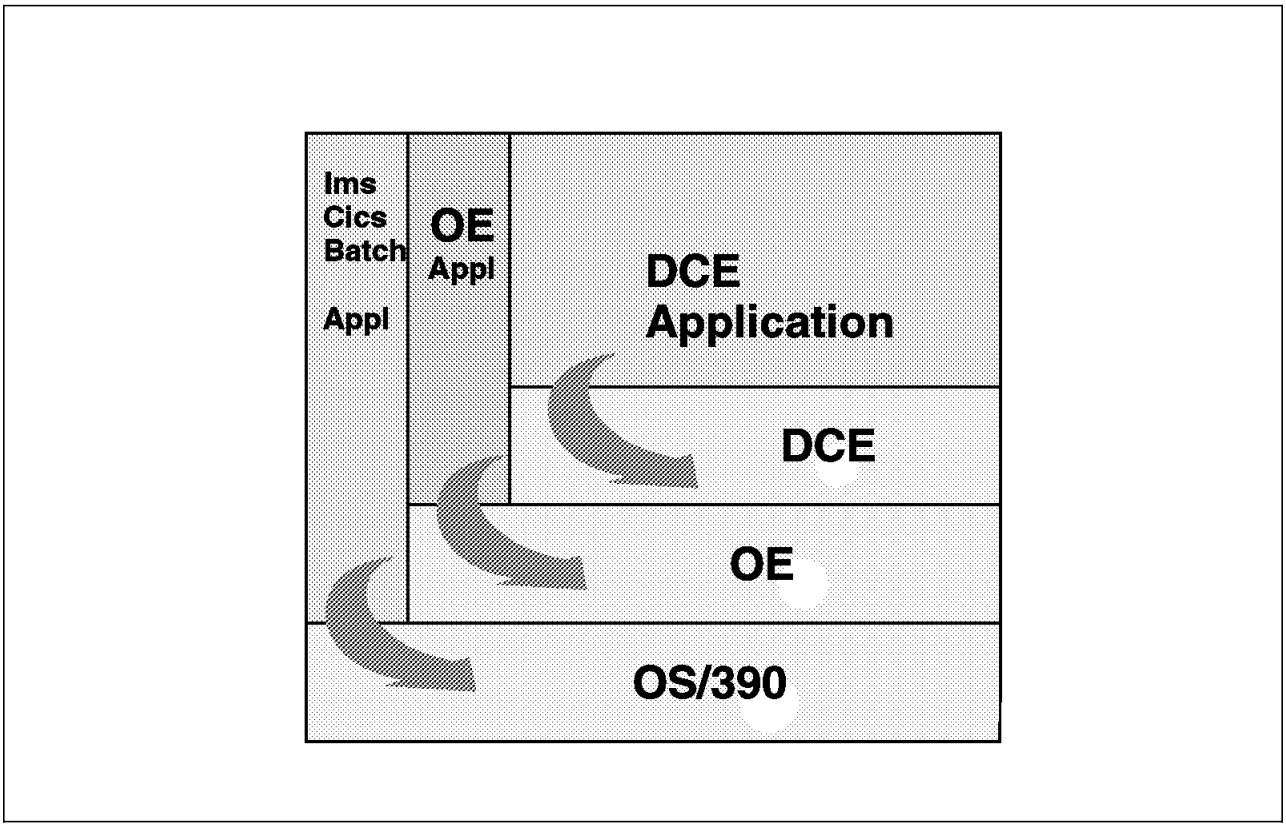


Figure 36. Layering of DCE and Related Software

DCE is a set of services that make up a high-level, coherent environment for developing and running distributed applications. DCE provides a set of services that make the interaction between a client and a server possible.

The four core services in DCE are Remote Procedure Call, Directory Service, Security Service and Distributed Time Service. The elements of these services run as long-running processes or *daemons*.

The OS/390 OpenEdition DCE run time environment is made up of the base DCE functions or services. The run time environment does not require all the services to be installed on a single machine; instead, they can be located on various machines throughout your DCE environment.

OS/390 OpenEdition DCE offers the following services that enable an OS/390 host to operate as a member of a DCE client/server configuration:

Remote Procedure Call consists of a development and run time service. The development tool consists of a language supporting the development of distributed

	applications following the client/server model. The run time service implements the network protocol.
Directory Service	is a central repository for information about resources in the distributed environment. Typical resources are machines, users and RPC services.
Distributed Time Service	provides synchronized time on the computers participating in a DCE environment.
Threads Service	supports the creation, management and synchronization of multiple threads of control within a single process.
Security Service	provides secure communications and controlled access to resources in the distributed system.

Figure 37 shows the DCE architecture and its technology components along with their relationship to applications, system support and future exploitations.

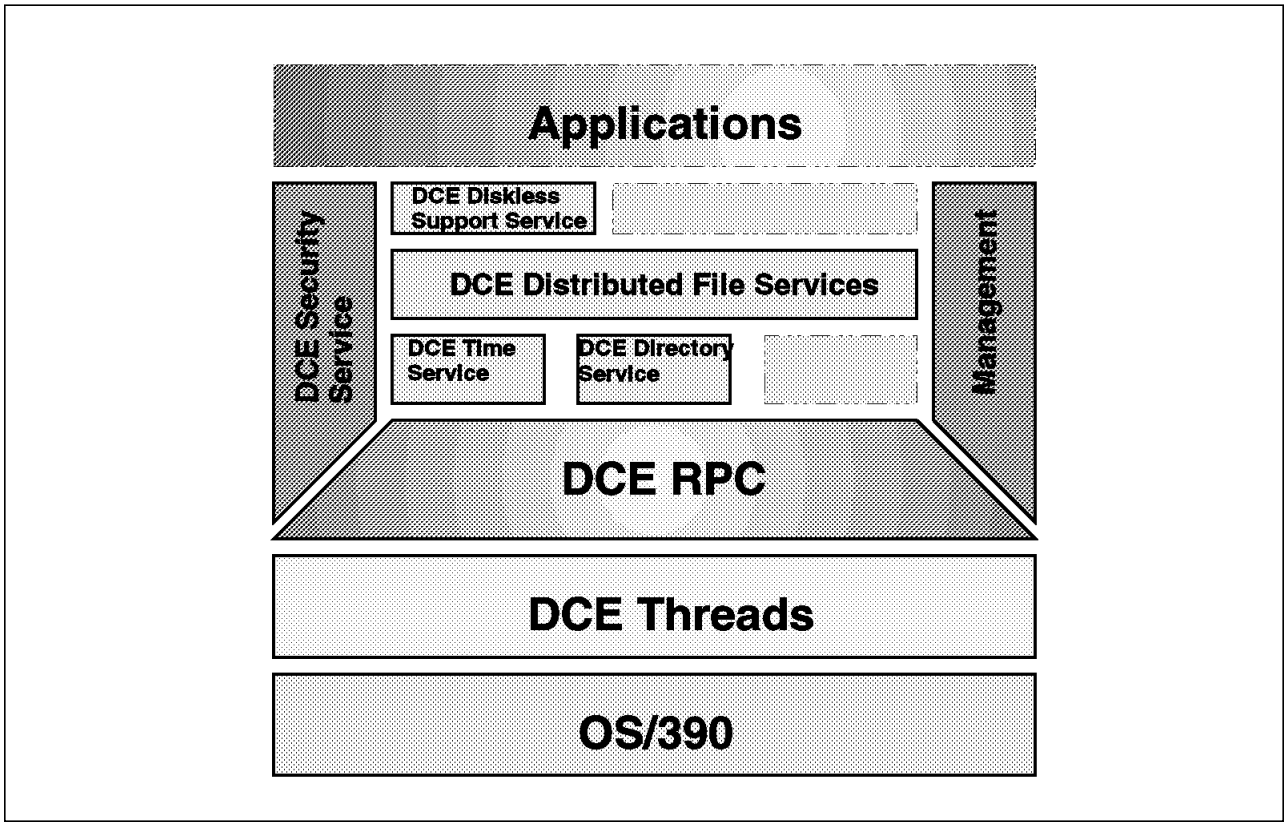


Figure 37. DCE Architecture

4.6.1 DCE Installation

This chapter discusses the steps required to install the DCE code on top of OpenEdition MVS. It does not include the customization of the DCE environment. For this specific subject, please refer to *OS/390 OpenEdition DCE: Configuring and Getting Started*, SC28-1583.

With the ServerPac you are not required to run any SMP/E jobs. The code for the DCE base services is provided by the ServerPac on a tape file. To complete the DCE installation, the OMVS address space must be running. The major steps of the DCE installation are as follows:

1. Run job HMB3120C to restore the data from the ServerPac source files SYS1.DCEHFS.GLOBAL and SYS1.DCEHFS.LOCAL to your new HFS files, DCEHFS.GLOBAL and DCEHFS.LOCAL or to any other data set name that matches your SMS environment. All file system directories needed for OE DCE BASE Services have already been created. Later during customization, you should mount as permanent these HFS files using the MOUNT statement in the BPXPRMxx PARMLIB member.

2. Mount the HFS data sets.

   Use the following TSO MOUNT commands to mount the DCEHFS.LOCAL and DCEHFS.GLOBAL data sets, which contain DCE.

   ```
   MOUNT FILESYSTEM('DCEHFS.LOCAL')
   MOUNTPOINT('/usr/lpp/dce/local') TYPE(HFS) MODE(RDWR)

   MOUNT FILESYSTEM('DCEHFS.GLOBAL')
   MOUNTPOINT('/usr/lpp/dce/global') TYPE(HFS) MODE(RDWR)
   ```

3. Make updates and additions to BPXPRMxx.

   Make the following changes to OpenEdition member BPXPRMxx:

   Make sure the name specified for the TYPE operand matches the name of the corresponding FILESYSTYPE statement.

   ```
   MAXFILEPROC(256)
   MAXTHREADTASKS(500)
   MAXTHREADS(500)
   NETWORK DOMAINNAME(AF_UNIX) DOMAINNUMBER(1) MAXSOCKETS(2000)
   TYPE(UDS)   <---- check for the corresponding FILESYSTYPE definition
                     from the OpenEdition customization


   MOUNT FILESYSTEM('DCEHFS.LOCAL')
   MOUNTPOINT('/usr/lpp/dce/local') TYPE(HFS) MODE(RDWR)

   MOUNT FILESYSTEM('DCEHFS.GLOBAL')
   MOUNTPOINT('/usr/lpp/dce/global') TYPE(HFS) MODE(RDWR)
   ```

   Replace or add if required the following parameters in BPXPRMxx:

   ```
   FILESYSTYPE TYPE(INET) ENTRYPOINT(BPXTIINT)
   NETWORK DOMAINNAME(AF_INET) DOMAINNUMBER(2) MAXSOCKETS(10000)
   TYPE(INET)
   ```

4. Set HFS Symbolic Links.

   Run the SYMLINK job to set the appropriate symbolic links for HFS files. This job is available in member EUVJOB06 of the SYS1.SAMPLIB data set.

5. Update the ISPF Command table ISPTCM in the SISPLPA data set.

   If any of the following modules are in LPA, you must update the ISPF Command Table ISPTCM. A sample source is provided in member ISPTCMA in the SISPSAMP library:

   EUVBCONF
   EUVCCP

EUVRCP
EUVRIDL
EUVRUGEN
EUVSAED
EUVSKDST
EUVSKINI
EUVSKLST
EUVSLGN
EUVSRED
EUVTCP

For further information, see *ISPF Planning and Customization*, SC34-4443.

6. Allocate the SEUVEXEC library to SYSEXEC.

   The DCE REXX EXECs are contained in the SEUVEXEC library. For DCE TSO users, ensure that the SEUVEXEC library is either allocated to SYSEXEC or concatenated with existing SYSEXEC libraries.

7. Customize IDL REXX EXEC (Optional).

   If you intend to use the IDL compiler under TSO, it is useful to make changes to the IDL REXX EXEC as specified in *MVS/ESA OpenEdition DCE: Installation and Configuration Experiences*, GG24-4480.

8. Copy the DCE procedure to PROCLIB.

   The PROCs needed to start the DCE started tasks as well as PROCs used by DCE application programmers and the DCE administrator are located in the target dataset SEUVPRC. Copy all the members and related alias into your PROCLIB dataset.
   Table 6 on page 119 is a list of the PROC names to be copied to your PROCLIB library.

Name	Alias	Description
EUVAP001	ACLEDIT	Executes the SECURITY ACL editor
EUVAP002	CDSCP	Executes the CDS control program
EUVAP003	DCEKERN	Executes the DCE kernel started task
EUVAP004	DCELOGIN	Executes the DCE login
EUVAP005	DTSCP	Executes the DTS control program
EUVAP006	IDL	Executes the IDL Compiler
EUVAP007	KDESTROY	Executes the SECURITY KDESTROY facility
EUVAP008	KINIT	Executes the SECURITY KINIT facility
EUVAP009	KLIST	Executes the SECURITY KLIST facility
EUVAP010	RGYEDIT	Executes the SECURITY registry editor
EUVAP011	RPCCP	Executes the RPC control program
EUVAP012	UUIDGEN	Executes the UUID generator facility

Table 6. DCE Cataloged Procedures

9. Customize the IDL Procedure (Optional).

   If you intend to use the IDL compiler under TSO, make changes to the IDL procedure as described in *MVS/ESA OpenEdition DCE: Installation and Configuration Experiences*, GG24-4480.

10. Define the security environment.

    For the security environment, please follow the same method and recommendations as specified for the OpenEdition MVS RACF definitions in 4.4.1.3, “Security” on page 109.

    - Define DCEKERN to RACF with the following actions:
      - Define DCEGRP as a group
      - Define DCEKERN as a user
      - Define DCEKERN as a started task

      For details, see the following: *MVS/ESA Planning: OpenEdition MVS*, SC23-3015, *OS/390 Security Server Security Administrator's Guide*, SC28-1915 and *MVS/ESA Open Edition DCE: Installation and Configuration Experiences*, GG24-4480.
    - Define the DCE Administrator user ID to RACF.
    - Create the DCEKERN environment in the “Facility” class through the definition of the following profile:

      ```
      DCEKERN.START.REQUEST
      ```

11. Update the TCP/IP profile.

Update the following parameters in the TCP/IP parameter file data set, which is allocated to the Profile DD statement in the TCPIP procedure:

```
SCBPOOLSIZE    768
UCBPOOLSIZE    256
```

12. Customize the IEEE 802 address.

    Edit the /usr/lpp/dce/local/etc/ether_addr file and update the 12 digit hexadecimal IEEE 802 MAC address with the address assigned to your system.

13. Update the DCEKERN procedure.

    If the LE/370 SCEERUN library is not in LPA or LINK list, you need a STEPLIB DD in the DCEKERN procedure before starting the DCEKERN address space.

14. Start the DCEKERN task.

    Start the DCEKERN procedure. Check for the following message:

    ```
    EUVP00024I  DCE kernel initialization is complete
    ```

    At this time the DCE kernel is ready for configuration.

4.7 DCE Maintenance

This paragraph will discuss a general methodology of how to manage maintenance when DCE is involved. Service for DCE code is installed in the same way as it is installed for traditional products, using SMP/E.

As for the OpenEdition code, DCE code is also distributed in HFS data sets. So far, maintenance on DCE code requires the same prerequisite environment as OpenEdition. DCE code can be considered like OpenEdition code from a maintenance point of view.

For this reason, you can refer to 4.5, “OpenEdition Maintenance” on page 113 and follow the same instructions and recommendations for DCE maintenance.

There are only two specific activities related to DCE:

- DCE does not require stopping and restarting the OMVS address space after applying maintenance. You should only mount the new HFS file at the specific mount point to get the new upgraded DCE code.
- Depending on which library was updated by the upgrade, you might close and restart the DCE address space DCEKERNEL.

4.8 OpenEdition DCE Distributed File Service (DFS)

The DCE Distributed File Service (DFS) allows users to access and share files stored on a file server anywhere on the network, without having to know the physical location of the file.

A user anywhere on a network can access any file, just by knowing its name. The DFS achieves high performance, particularly through caching of file system data. Many users can access files that are located on a given file server without a large amount of network traffic or delays.

DCE DFS includes a physical file system, the DCE Local File System (LFS), which supports special features that are useful in a distributed environment. These

features include the ability to replicate data, to log file system data - enabling quick recovery after a crash and to simplify administration by dividing the file system into easily managed units called filesets.

DCE also offers Diskless Support Service, which provides the tools that allow a diskless node to:

- Acquire an operating system over the network
- Obtain configuration information
- Connect to DFS to obtain the diskless node's root file system
- Perform remote swapping

When these tools are incorporated into the client's operating system and hardware, the diskless node can operate in a DCE environment.

4.8.1 DCE DFS Installation

This session discusses the steps required to install the DCE DFS code.

With the ServerPac you are not required to run any SMP/E jobs. The code for the DCE DFS has already been installed and comes on a ServerPac tape file. To go through the installation, the OMVS address space must be running in order to manage the HFS files. Following is a list of the major installation steps:

1. Run job H0H2110C to restore the data from DFS.V5R2M2.GLOBAL and DFS.V5R2M2.LOCAL as HFS files. All file system directories needed for DFS have already been created in the current ROOT file. Later during customization, you should mount as permanent these HFS files using the MOUNT statement in the BPXPRMxx PARMLIB member.

2. Mount the HFS data sets.

   Use the following TSO MOUNT commands to mount the DFS.V5R2M2.LOCAL and DFS.V5R2M2.GLOBAL data sets which contain DCE.

   ```
   MOUNT FILESYSTEM('DFS.V5R2M2.LOCAL')
   MOUNTPOINT('/usr/lpp/dfs/local') TYPE(HFS) MODE(RDWR)

   MOUNT FILESYSTEM('DFS.V5R2M2.GLOBAL')
   MOUNTPOINT('/usr/lpp/dfs/global') TYPE(HFS) MODE(RDWR)
   ```

3. Make updates to BPXPRMxx PARMLIB member.

   ```
   MOUNT FILESYSTEM('DFS.V5R2M2.LOCAL')
   MOUNTPOINT('/usr/lpp/dfs/local') TYPE(HFS) MODE(RDWR)

   MOUNT FILESYSTEM('DFS.V5R2M2.GLOBAL')
   MOUNTPOINT('/usr/lpp/dfs/global') TYPE(HFS) MODE(RDWR)
   ```

4. Set DFS Symbolic Links.

   Run the SYMLINK job to set the appropriate symbolic links for HFS files. This job is available in the member SYMLINK of the SIOESAMP data set.

5. Allocate the SIOEEXEC library to SYSEXEC.

   The DFS REXX EXECs are contained in the SIOEEXEC library. For DFS TSO users, ensure that the SIOEEXEC library is either allocated to SYSEXEC or concatenated with existing SYSEXEC libraries.

6. Copy the DFS procedure to PROCLIB.

The PROC needed to start the DFS started tasks is located in member DFS in the target data set SIOEPROC. Copy the procedure to your PROCLIB.

7. Define the security environment.

   For the security environment, please follow the same method and recommendations as specified for the OpenEdition MVS RACF definitions in 4.4.1.3, “Security” on page 109.

   - Define DFS to RACF with the following actions:
     - Define DFSGRP as a group
     - Define DFS as a user
     - Define DFS as a started task

     For details, see the following: *MVS/ESA Planning: OpenEdition MVS*, SC23-3015 and *OS/390 Security Server Security Administrator's Guide*, SC28-1915.

   - Create the DFSKERN environment in the FACILITY class through the definition of the following profile:

     ```
     DFSKERN.START.REQUEST
     ```

8. LNKLST consideration.

   Add to your current Linklist the DFS SIOELMOD target library and verify that this library has to be APF authorized.

9. Start the DCEKERN address space.

   Start the DFSKERN with the following command:

   ```
   S DFS,PARM='-nodfs'
   ```

   and check for the following message:

   ```
   IOEP00024I  DFS kernel initialization complete
   ```

   At this time the DFS kernel is ready.

4.9 DFS Maintenance

This paragraph will discuss a general methodology of how to manage maintenance when DFS is involved. Service for DFS code is installed in the same way as it is installed for traditional products, using SMP/E.

As for the OpenEdition code, DFS code is distributed also in HFS data sets. So far, maintenance on DFS code requires the same prerequisite environment as OpenEdition. DFS code can be considered like OpenEdition code from a maintenance point of view.

For this reason, you can refer to 4.5, “OpenEdition Maintenance” on page 113 and follow the same instruction and recommendations for DCE maintenance.

There are only two activities specific to DFS:

- DFS does not require stopping and restarting the OMVS address space after applying maintenance. You should only mount the new HFS file at the specific mount point to get the new upgraded DFS code.
- Depending on which library was hit by the upgrade, you might close and restart the DFS address space.

Chapter 5. Testing

This chapter deals with the testing phase of software management, as shown in Figure 38, and will discuss the following topics:

- What testing is.
- Why test?
- Testing categories.
- Testing tips.
- Creating and designing a test plan.
- Testing alternatives.
- Testing vehicles.
- Test environments.

5.1 Introduction

What is testing and why should you do it? This chapter briefly examines answers to these questions. However, you should be aware that testing is a complex topic, and cannot be considered in detail in this publication. You should give considerable thought to how you go about testing in your environment, perhaps using this chapter as an initial “discussion guide.”

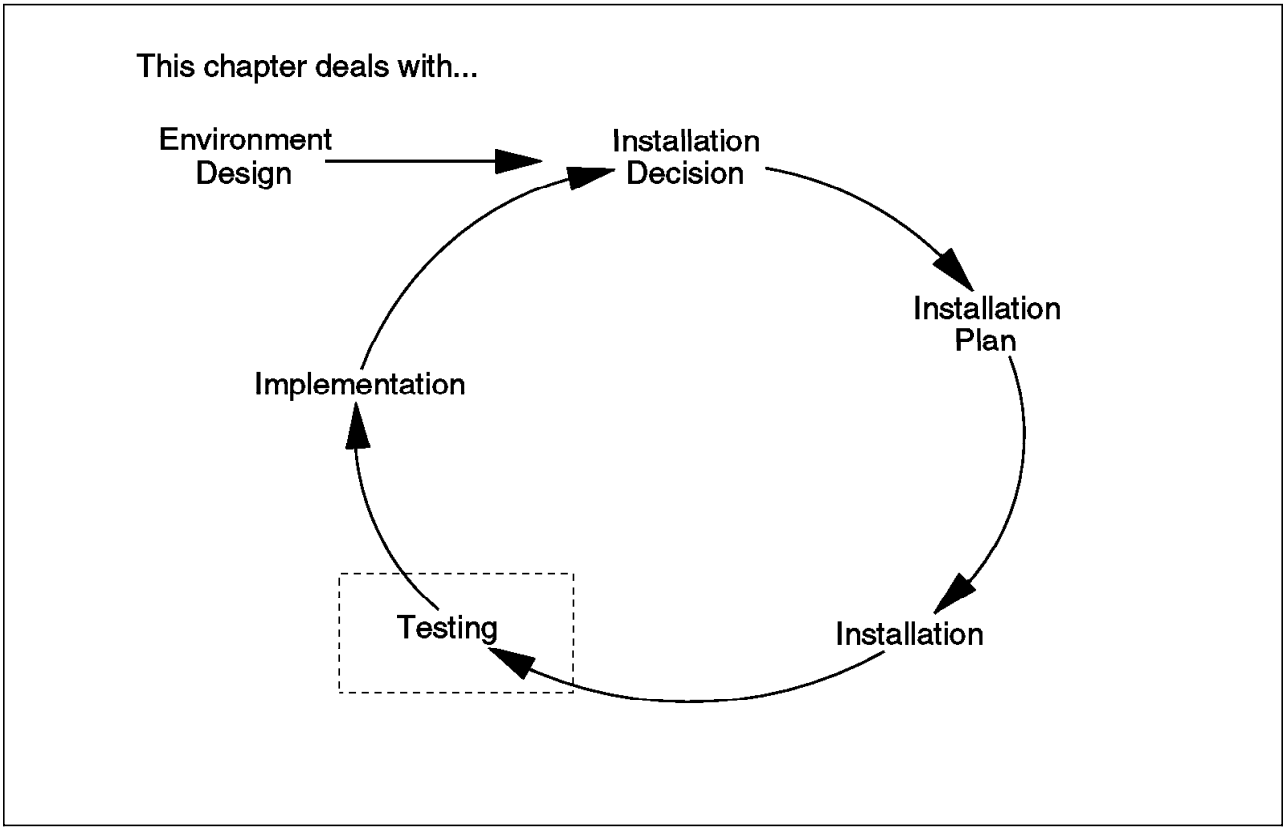


Figure 38. Testing

When deciding on a testing strategy, you will have to review the following:

- Why should testing be done (what is its purpose)?
- What are the phases of testing?
- What tools and techniques are available to assist with testing?
- How can success in testing be measured?

This chapter will give some pointers to help you get started.

Although testing could be seen as the “end” of the cycle, you should be thinking about the quality assurance of your software management practices all the way through the process. Do not make testing an afterthought.

5.2 What Testing Is

It is worth taking a few minutes to think about what testing is. “Testing” is pervasive in information systems. Every software developer, system programmer, and operations manager knows that testing is important, and that new software must be tested before implementation. Often, we test because it is something we have always done, rather than for specific business reasons.

Testing is often considered to be the most time consuming, and as a consequence, the least interesting phase. Testing is traditionally carried out last, and either because of inadequate planning, a shortage of time, schedule slippages, or pressing business needs, this important phase is often neglected. The result is that newly delivered systems are of poor quality, get backed out, and the development costs escalate. Figure 39 illustrates this point.

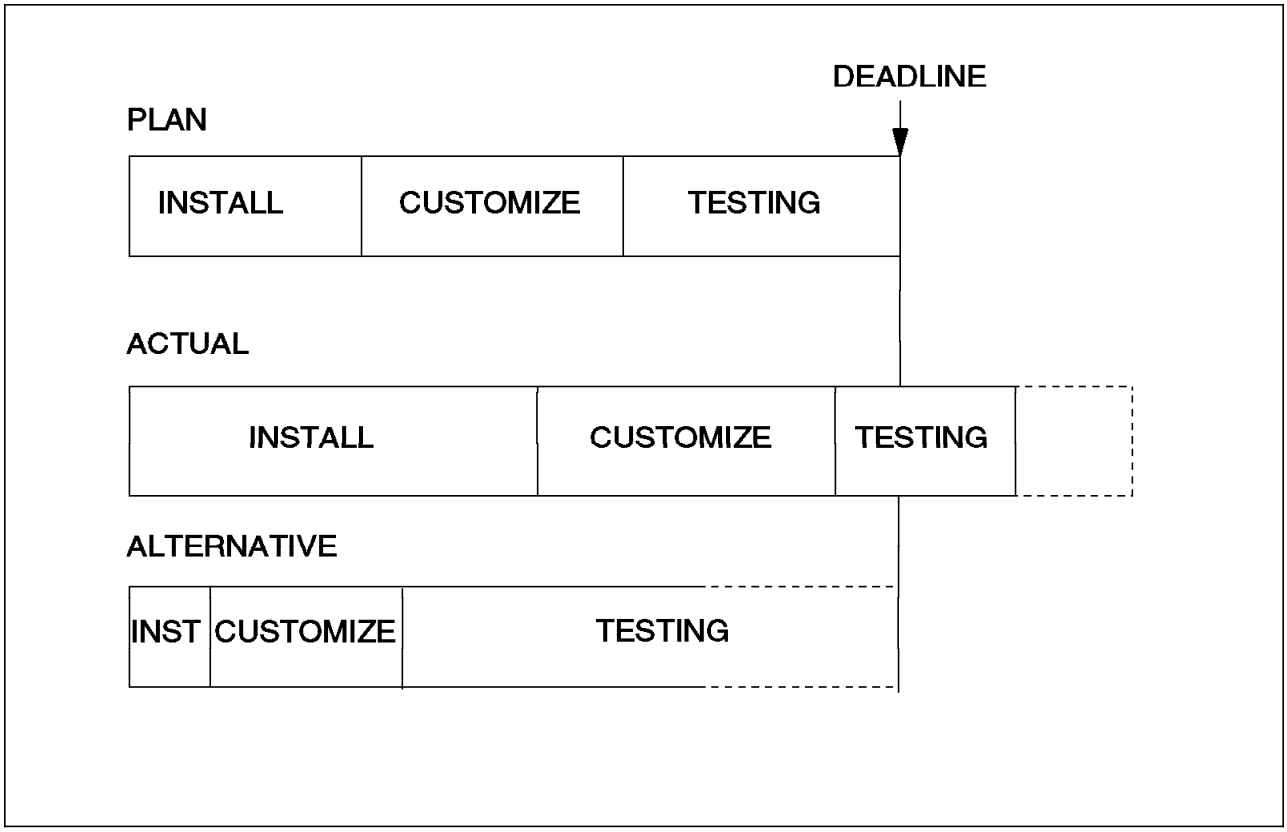


Figure 39. Testing As Part of Software Management

By careful planning and using the latest software management tools such as the CustomPacs, it is possible to shorten the installation and customization time, and allow enough time to carry out testing as planned.

OS/390 provides enhanced integrated testing procedures adding to the robustness of the S/390 platform. Additionally, the ServerPac system replacement vehicle provides a system that is already IPL and IVP tested. This means that customer testing can concentrate more on application-specific testing.

Testing is quality assurance. It is a set of techniques for assuring that your information systems function as they were designed. A good time to do such quality assurance is when you are making changes to the system. At that time you can focus on the effect of the changes that you are making, rather than trying to assure all the functions of the system.

Testing is a means of helping ensure that a program, or a collection of programs, conforms to specifications and works as it is supposed to work. Software is tested by executing it under simulated conditions to see when it fails. Testing should be a destructive process, executed with the intent of finding errors.

Types of testing are described in section 5.4, “Testing Categories” on page 126.

5.3 Why Test?

The growing demand for 24-hour-per-day, 365-day system availability is an indicator of how much businesses have come to depend on their information systems. Changing your system, by installing new software or maintenance can introduce unforeseen problems. Testing (or quality assurance) could be used to reduce the risk of such problems occurring.

Before you consider building a testing plan, you should clearly state what your objectives in testing are. In other words, you should be able to state the business benefit your testing activities will bring. For example, you may have an objective to reduce the number of unscheduled outages by 10% per year as a result of testing. If you have objectives, then it will be possible to measure your success against those objectives. This ability to measure brings a number of benefits to the business:

- You are able to determine the value of the investment you make in testing (hardware, software, tools, and people) to the business.
- You are able to track your improvement over time, and detect any problems in processes or people skills before they impact availability.

It is likely (almost implicit) that one of your objectives will be to improve system availability. Therefore, your objective in testing should be to find errors. Although you can never guarantee that your testing will find all errors, you should have the objective of finding all possible errors during testing. This means that you have to design tests that are most likely to find potential errors in the system. You can do this by looking at:

- Areas of the system that have changed; for example, new releases of products.
- Areas of the system that have caused problems in the past.

- Areas that are heavily used in your enterprise.
- Areas that your enterprise has modified.

The bottom line is that you are trying to break the system during testing so that it does not break in production. A test environment is a much better place to experience a failure than the production environment. The better the design and execution of the test cases, the less chance for production failures.

At the same time, you should not rely on testing “to find bugs” that should have been found earlier. In other words, everyone involved in software management must be responsible for quality, and must do their own “testing.”

5.4 Testing Categories

It is an installation′s responsibility to test installed code to ensure that it functions correctly in its unique operating environment. For the purposes of software management, testing can be divided into verification and validation activities as depicted in Figure 40, and they include:

- Unit testing
- Integration testing
- Function testing
- System testing
- Acceptance testing

There are also other forms of testing, for example:

- Regression
- Capacity
- Stress

Regression testing should follow all the other software validation testing activities. Capacity testing and stress testing should be considered part of normal systems management, and may also form part of software validation testing.

5.4.1 Verification Activities

Unit and integration tests are concerned with program code, structure, and the relationships between modules. This testing is based upon a programmer′s view and deals with internal and detailed design specifications.

5.4.1.1 Unit Testing

Unit testing is generally performed by programmers as part of a coding process and includes:

- Executing every statement and possible logical path in a module.
- Executing code to verify the basic structure of modules, and the correctness of the logic.
- Code inspection and data flow analysis to determine whether modules will do what they were designed to do, and to point out omissions.

Unit testing will be of interest to systems programmers who code USERMODS and EXITS.

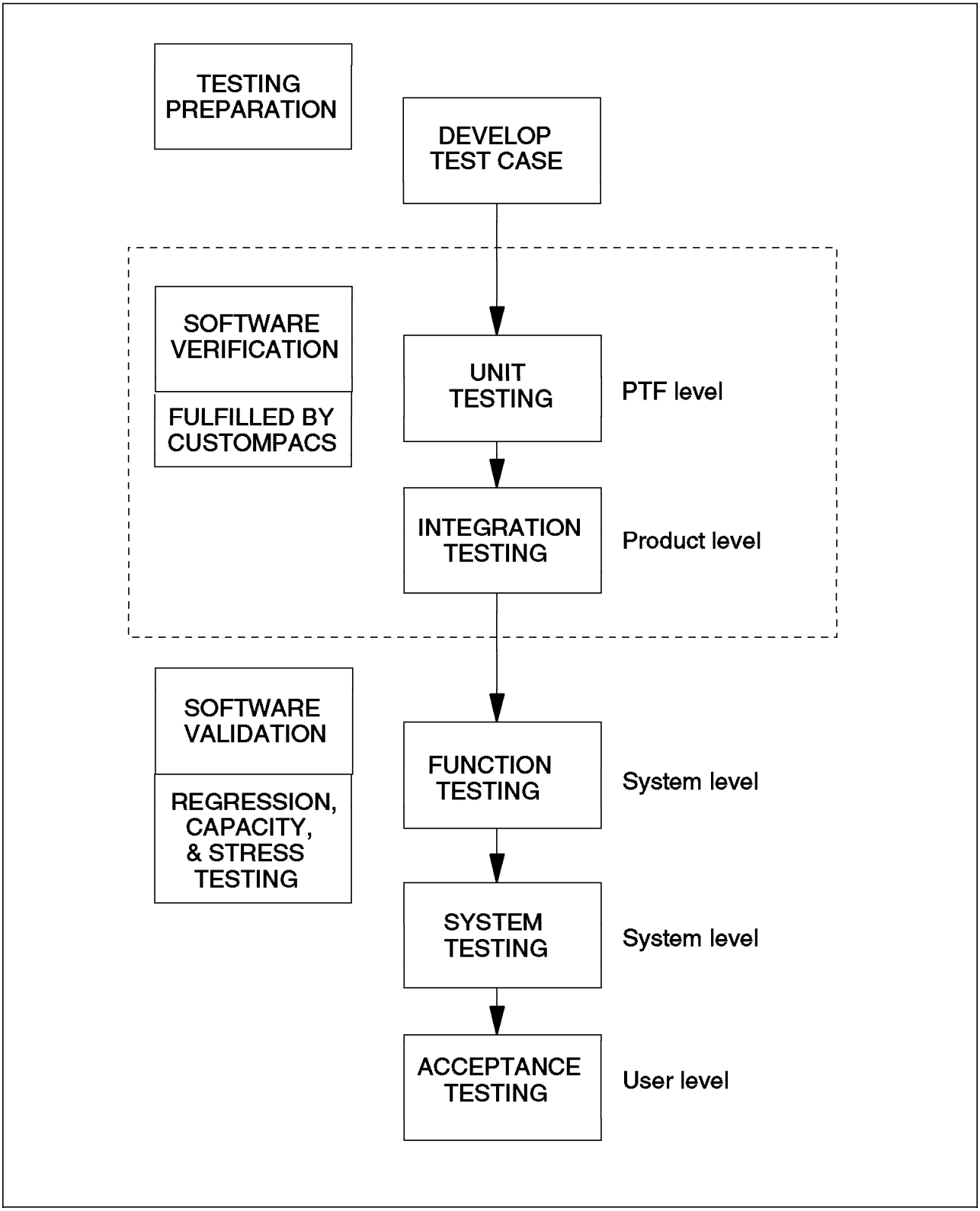


Figure 40. Testing Categories

5.4.1.2 Integration Testing

Integration testing concentrates on interfaces (communication) between modules and programs. Part of integration testing is concerned with maintenance testing or maintenance verification. Maintenance testing ensures that no known problems are installed on the system. It is performed after any maintenance (CBPDO for example), has been applied to the system. If you are using CustomPacs, this testing has been performed for you by IBM. Maintenance testing is described in detail in Section 4.2.6, “Maintenance Testing” on page 101.

5.4.2 Validation Activities

Function, system, and acceptance testing, unlike unit and integration testing, are concerned with the relationships between programs and collections of programs, and deal with external program relationships.

5.4.2.1 Function Testing

Function testing is designed to test a function of the system, independent of other functions, to determine whether the function works correctly. These tests ensure that system functions work and are understood. They include testing of operator commands, TSO functions, compilers, utilities, exits, and system components. They also provide invaluable “hands-on” education and experience for system programmers. Function tests should be performed early in the testing cycle, following the implementation of:

- New application transactions
- Log-on and log-off sequences
- Error transactions
- New hardware attachments
- New software products

5.4.2.2 System Testing

System testing is generally a combination of function tests conducted on a mirror image of the intended final production system. System testing will ensure that an entire system, made up of many separate functions, will work collectively and correctly. Functions may not fail function testing, but their interrelationships may be critical to the overall operation of a system. System testing should include system software and hardware, and the enablement of applications. It is important to have a complete system image dedicated to system testing. Figure 41 illustrates a test system, running in an LPAR, accessing a copy of the production application data. This test application data would normally be a representative subset of the production data.

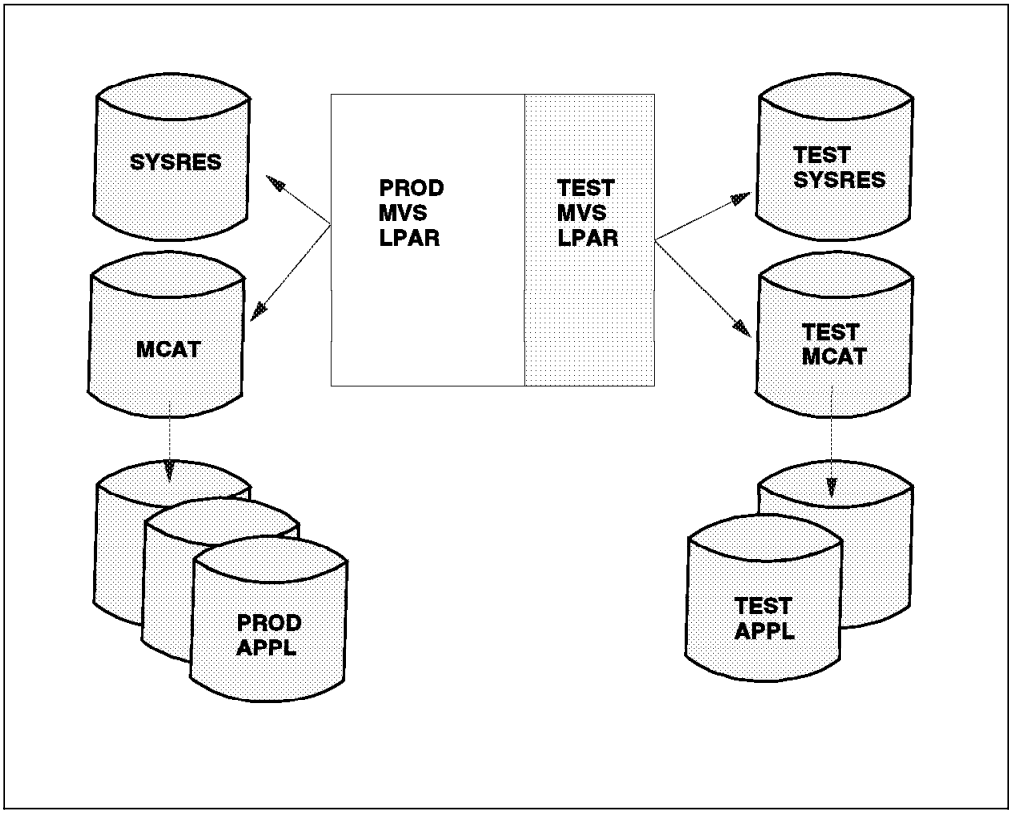


Figure 41. Accessing Test Application Data

Additional consideration should be given to testing of applications running in and exploiting a Parallel Sysplex environment. OS/390 has undergone considerable testing in an environment that simulates the workload stress levels, transaction rates and lock contention rates that customers might experience. This does not however remove the need to do customer-specific system testing. See the *OS/390 Parallel Sysplex Test Report* for a description of the system environment used and the findings of the test team. This report is published quarterly and is not available through the normal publications ordering process. IBM account representatives can provide copies if required.

It is not practical for customers to do system testing on an exact mirror image of the production sysplex, therefore some compromise will be required. Some alternatives for providing a valid sysplex system test environment are listed below:

- Use existing production environment outside of service hours. This is the ideal test environment but for most shops the opportunity for such testing slots will be very limited. Also, either no production data can be updated or the test must be run against copies of the live data.
- Use maintenance and development environment systems to provide a reasonable approximation of the production sysplex environment. Figure 42 illustrates this.
- Remove one or more images from the production sysplex environment and configure into the maintenance and/or development environment to provide a more reasonable approximation of the production sysplex environment. This would need to be done when the full capacity of the production sysplex is not required, for example, on weekends. Figure 43 illustrates this.

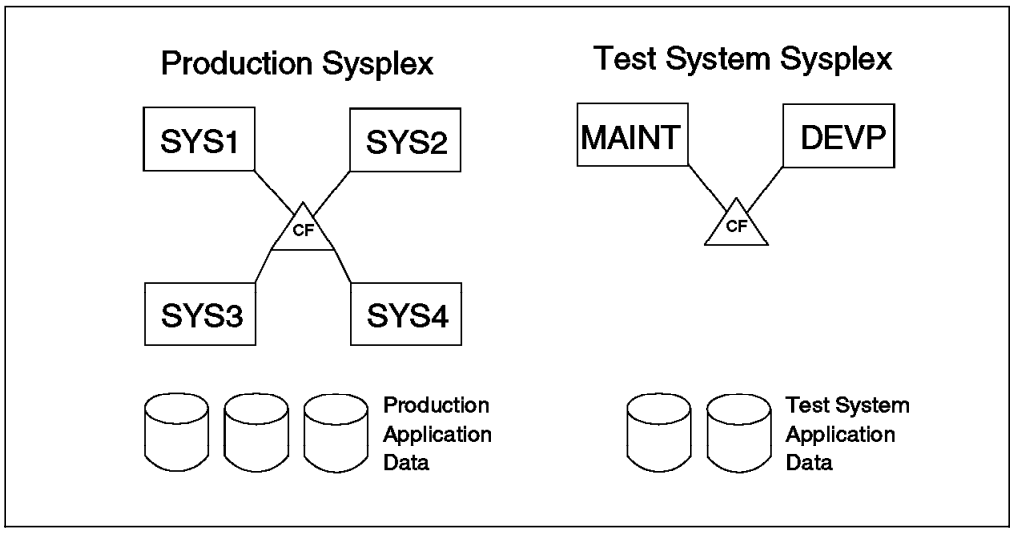


Figure 42. Test System Sysplex

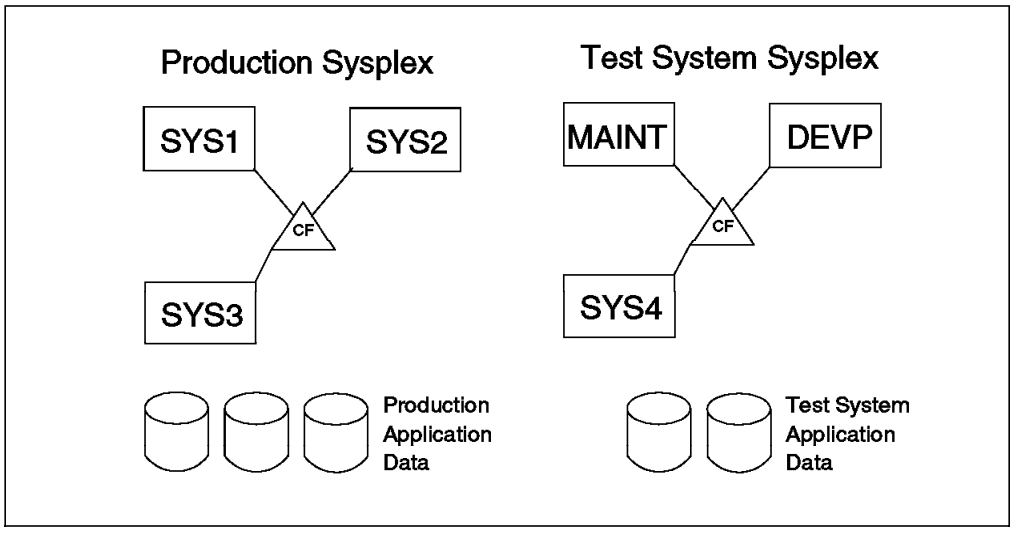


Figure 43. Enhanced Test System Sysplex

5.4.2.3 Acceptance Testing

Acceptance testing should fulfill two functions:

- User Community Testing

  Having demonstrated, through system testing, that applications can be enabled, all functions that are business dependent (the reason for having a computer) should be thoroughly tested by representatives of the appropriate user communities. Users should adhere to the same guidelines discussed in this chapter, and therefore plan their testing, and carry out testing in a structured, repeatable, and thorough manner.

- Performance Testing

  Performance testing assures that you will meet your service level objectives. Response times, throughput, job turnaround, and operator preparedness must be tested to ensure that you are ready for production work. These tests include the measurement and tuning functions. They test, among other

things, the effect of changes to system parameters prior to live implementation. The test vehicle used should provide:

- A controlled, repeatable load on the system
- Measurement and reporting capabilities to measure throughput, job turnaround, and response times, including the response time at the terminal level

These tests should be used before going into production with any significant change or new function.

5.4.3 Regression, Capacity, and Stress Testing

Regression, capacity, and stress testing are usually done as the final testing steps before putting a new system into production.

5.4.3.1 Regression Testing

Regression tests verify that a pre-existing function still operates correctly after new functions have been added, or after changes have been made to the system. These tests are used frequently to ensure that an enterprise′s application workload runs under the new system. Regression tests:

- Should be repeatable
- Should produce predictable results
- Can be used to check for errors in system responses
- Should run automatically

Regression testing should be conducted before putting any significant change or new function into production.

5.4.3.2 Capacity Planning Tests

Capacity planning tests predict how a system will behave when a new resource is brought online. These tests assess whether the system has the capacity to handle additional terminals or new functions such as a token-ring network.

5.4.3.3 Stress Tests

Stress tests find problems in interactions and resource contention by driving the system to extremely high transaction rates. Teleprocessing drivers such as TPNS, heavy batch workloads, service aids, sorting programs, dumps, and restores should be included in these tests. Stress testing is further discussed in section 5.8.1, “TPNS As a Test Vehicle” on page 135.

5.5 Testing Tips

Before testing begins, a decision must be made as to what changes will be tested. It is both costly and impractical to test everything. To assist with the assessment of what should be tested, the following questions should be answered:

- How complex is the change?
- How many components will be changed?
- How many transactions or transaction types will be affected by the change?
- How many people will be affected by the change?

- How difficult will backup and recovery be after the change?
- Has this type of change been performed in the past, and how significant was its impact?

There are some basic rules that must be obeyed when testing:

- Define the expected result (do not test “on the fly”).
- Do not stop testing even if parts of the program are correct.
- Test for invalid and unexpected results (things that should not occur).
- Avoid temporary tests (develop a plan).
- Concentrate on areas that already have errors.
- Simulate the production-level environment.

Simulating the production-level environment is important as this will prove the effectiveness of testing. For example, if you have had a problem with a user program that invokes IDCAMS to copy data into a VSAM cluster, do not simply test that the fixing PTF has cured the problem by running IDCAMS REPRO; test first using the failing user program. It may make its calls to IDCAMS in a completely different way. If the fix affects IDCAMS REPRO, then test that as well using the same data.

5.6 Creating and Designing a Test Plan

Before conducting or even planning a system test, test objectives must be established. The objectives should contain details on the purpose of the test and the expected results. The expected results must be stated in measurable terms.

The test plan documents the tests that will be conducted. It should have the following format:

- An introduction or a high-level overview of what is being tested, why it is being tested, and who will do the testing
- Objectives, to define the purpose of the test and its expected results
- Hardware resource requirements
- Software resource requirements
- Personnel resource requirements
- Test specifications. This section contains specifications for:
  - Defining the network to be tested
  - Operating the test
  - Test output
- Testing procedures, which are divided into the following sections:
  - Entry and exit criteria
  - Sequence of sample tests
  - Procedures for problem reporting and test status
- Schedule

Additional considerations when designing a test plan include:

- Using a separate system image for testing, if possible. You could use a S/390 Microprocessor or the PR/SM feature of the ES/9000 processor to provide a logical partition, in which a test OS/390 image could execute on a daily basis. If you do not have a separate image for testing, provisions should be made for as many test windows as possible, without impacting production workload and performance. If possible, weekday test windows should be provided, as this provides better turnaround for problem resolution activities. “Weekend only” testing results in long turnaround time for problems.
- Testing the most critical, or hardest to fix, applications first. These are the applications most subject to failure or degraded performance. This allows more time to modify or tune these applications as necessary.
- Testing all non-IBM software products early, so that you have time to contact vendors and resolve problems.
- Capturing all necessary documentation. Since you test to find errors, ensure that all diagnostic tools and procedures are available to record any failures. For example, you should ensure that space is available for TRACE tables, GTF data, LOGREC data, console logs, dumps, and so on. Allocate more trace entries than you normally would in the production environment.
- Testing all operations procedures such as backups, restores, recoveries, and restarts during the test period. Don′t wait for a crisis when in production.
- Making applications personnel responsible for choosing the application tests. They should also sign off when they are satisfied with the testing.
- Having systems programmers doing system programmer activities such as system maintenance, tuning, system monitoring, and updating system programs and tools during testing.
- Having operations personnel do the operating and testing. This is a vital part of their training, and ensures that normal operational procedures are followed during testing.
- Having operators experiment with new operator commands and facilities during tests.
- Having operators establish and maintain a console log book to record all errors, idiosyncrasies, new conditions, and notes during the testing period.
- Testing periodically scheduled programs and applications, such as end-of-year processing.

Each test phase can be considered complete when the activities described in the test design and the results dictated by the acceptance criteria are realized. The by-products of thorough testing are education, training, confidence, and success.

5.7 Test Alternatives

As an alternative to the testing discussed above, you have two basic choices as shown in Figure 44.

5.7.1 Real Users at Real Terminals

That is, you can bring in a large group of people during off-shift to enter data from real terminals, making sure that they follow some predefined script so you can repeat the results if necessary.

Although this method will work, it has a number of substantial problems:

- It is costly in terms of people and hardware.
- If you do find a problem, it is difficult to verify the problem solution because it is difficult to get people to do things exactly the same way twice.
- It is difficult to verify that the people followed the predefined script.

5.7.2 Test Tool or Test Driver

A test tool or test driver is a computer program that acts like one or more users, varying terminal types or network components without requiring a large group of people or a lot of hardware.

This saves people, resources, and time.

An external driver means that the test tool functions independently of the system under test. This is the preferred method.

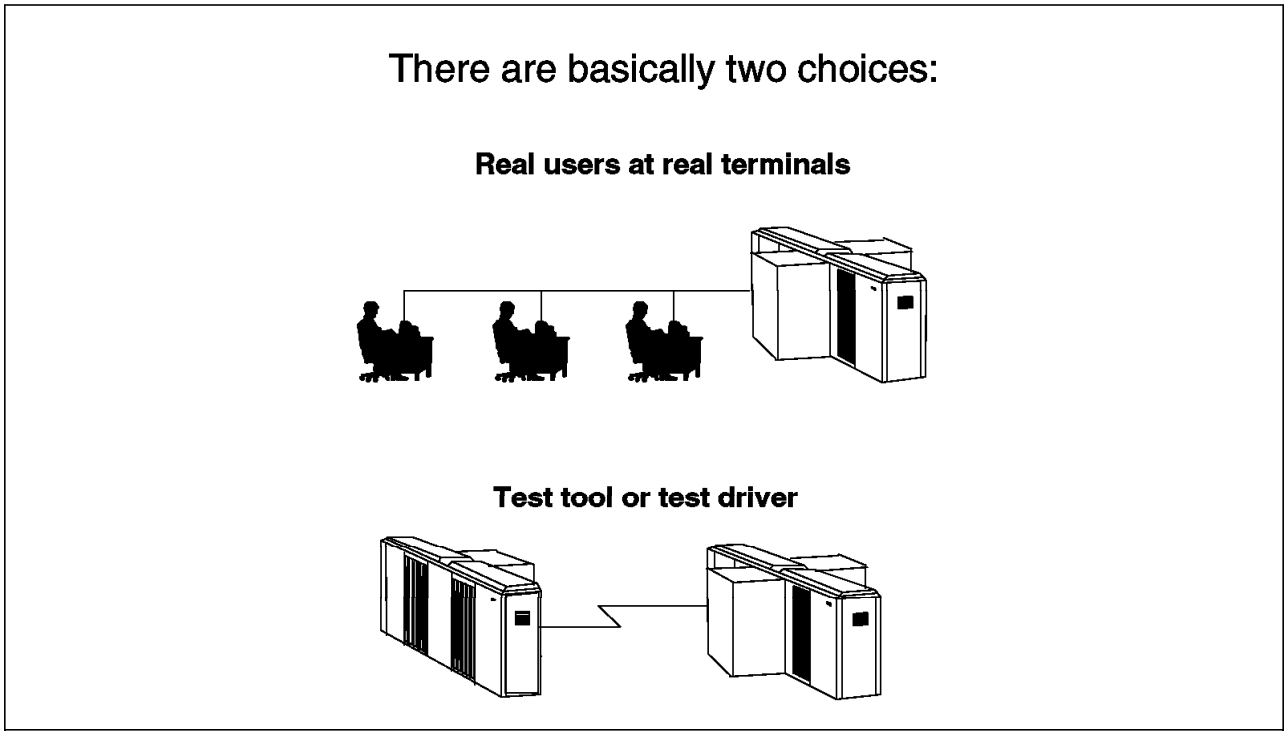


Figure 44. Testing Alternatives

5.8 Test Vehicles

This section contains a brief discussion on selecting test vehicles, followed by an overview of the teleprocessing test tool, Teleprocessing Network Simulator (TPNS), and the Workstation Interactive Test Tool (WITT).

The test vehicle must exhibit the following characteristics:

- It must provide for off-peak testing. This minimizes the impact of testing on end users, as it is not competing with production systems for machine cycles.
- It must be user friendly.
- It must be flexible and provide a choice in tests to be run.
- It must permit tests to be repeatable.
- It must utilize currently available, supported software.
- It must operate from a common hardware platform.

5.8.1 TPNS As a Test Vehicle

This section provides an overview of the Teleprocessing Network Simulator (5688-121) and the implementation of this product.

TPNS is a terminal and network simulation tool that allows application programs, communication access methods, and communication control programs to be evaluated without the use of actual terminals. TPNS can be used to estimate system performance and response time. TPNS provides controlled generation of message traffic into a telecommunications subsystem or application through the use of processing, rather than with terminal hardware and terminal operator time. TPNS provides the ability to simulate a specified network of terminals and their associated messages, allowing the user to alter network conditions and message loads during a run. It can therefore be used to stress-test telecommunications application programs with volume messages to evaluate the reliability and approximate performance characteristics under expected operating conditions. It provides facilities to perform system function tests and to automate regression tests. TPNS can also be used to evaluate network design and assist with capacity planning.

TPNS is used primarily by installations requiring the following:

- High system availability, such as systems providing essential business operation functions
- Fast response time to terminal messages
- The capability to handle heavy message traffic
- High system stability, implying the ability to predict and measure the effect of changes
- Feasibility assessments
- Confidence in meeting application development requirements

TPNS, by improving the comprehensiveness of testing and by setting consistent testing standards and successful completion criteria, assists in enabling an installation to meet these requirements.

To use TPNS, you must create a simulation program, run the simulation, and analyze the results. The TPNS/ISPF Interface allows you to do these tasks from an ISPF menu. Figure 45 shows this process and the components of TPNS.

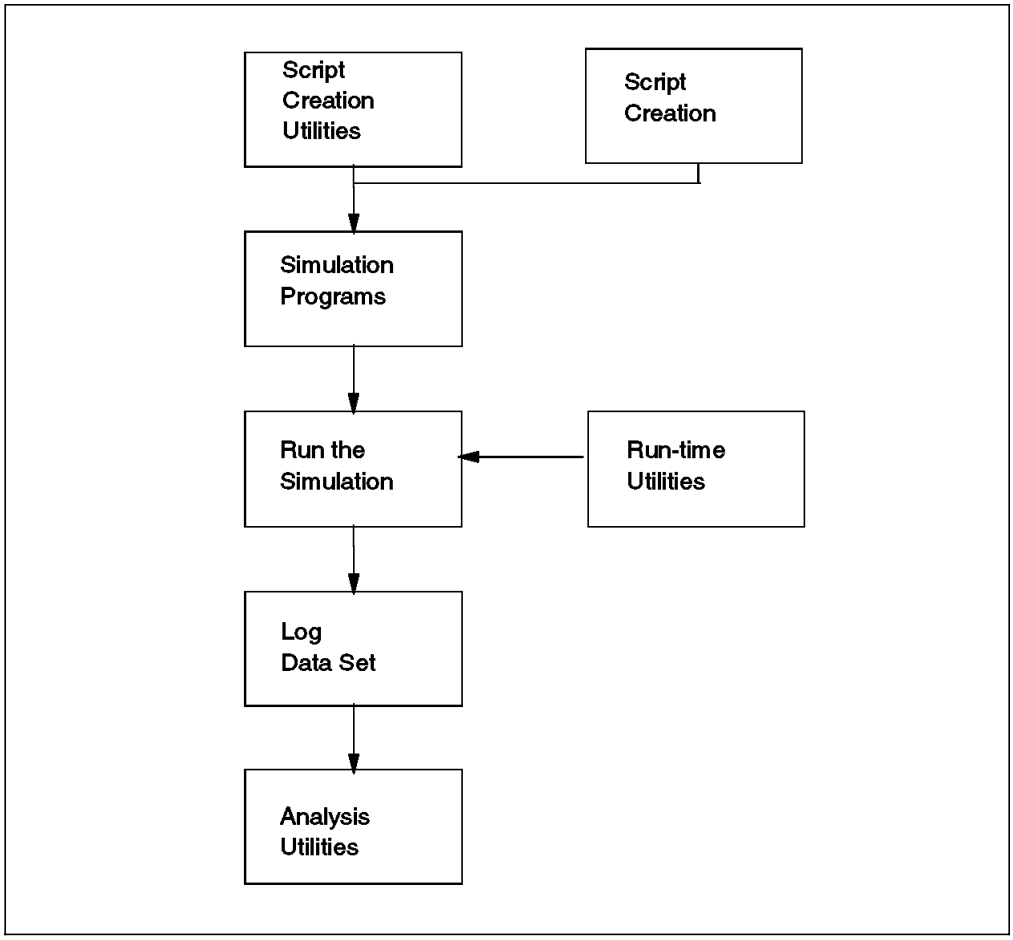


Figure 45. Using TPNS

5.8.2 WITT As a Test Vehicle

Workstation Interactive Test Tool (5648-031) provides automated regression testing services for interactive software systems. As such, it is helpful when:

- You migrate products from one system environment to another and have to prove that the products behave as expected. WITT can automatically test the product in the new environment for behavioral changes by re-executing test cases prepared in the old environment and comparing output screens captured in both environments.
- You install a new release of software that has more functions than the old one had. To ensure that the unchanged part of the new release behaves as it did before, you perform a regression test with WITT. As soon as the new release behaves as expected, you can enhance your test cases. The enhanced version of the test cases provides the test base for future changes to your software.
- You wish to record keystrokes. While a user interacts with the software, WITT is enabled in the background to record keystrokes. As a result, there is a record of the path the user took through the software. If the user dislikes something about the implementation of the new software, he or she

can use WITT to play back the keystrokes (in single-step mode if necessary). The user could then point out exactly what function he or she dislikes. The programmer can then make changes and use WITT to play back the user's scenario, (automated acceptance testing).

For further information refer to *Workstation Interactive Test Tool User's Guide*.

5.9 Test Environments

Testing will normally be performed in three distinctly different environments:

- Upgraded maintenance environment
- Production maintenance environment
- Test operational environment

The reasons for testing in each environment will differ, according to the purpose of each environment (see Chapter 3, "Maintenance Environment" on page 39):

- The upgraded maintenance environment will be used for extensive testing of an entire system image, including subsystems, networks, and applications. When completely tested, the system image will be propagated into production. This is further discussed in section 6.2, "Moving into Production" on page 140.
- The production maintenance environment will be used for testing fixes to the system image. Testing will involve testing any system, subsystem, or program that is associated with those fixes, or against the entire system. Once tested, the fixes will be propagated to the production environment. Testing in the production maintenance environment will ensure that applying concurrent maintenance as discussed in section 2.7, "Concurrent Maintenance" on page 35 will not affect the availability of the production operational environments.
- The test operational environment will be used for program, application and subsystems testing, and may include the subsystem upgraded maintenance levels and subsystem production maintenance levels (copies of the subsystem production operational levels minus maintenance).

Chapter 6. Production and Implementation

This chapter deals with the implementation phase of software management as shown in Figure 46, and will discuss the following topics:

- Moving into production
- Customizing the new system
- Determining the method for master catalog update
- Testing prior to production
- Propagating a new system into production

6.1 Introduction

Having installed a new system that will function at a basic level, there are many activities that you will have to perform to enable your system to operate in a manner that is pertinent to your business needs. This chapter suggests some software management techniques to enable you to move into production. You will have to make many changes during the life of your systems, and it is important that the design of your environment does not restrict your flexibility or cause duplicate work when making changes.

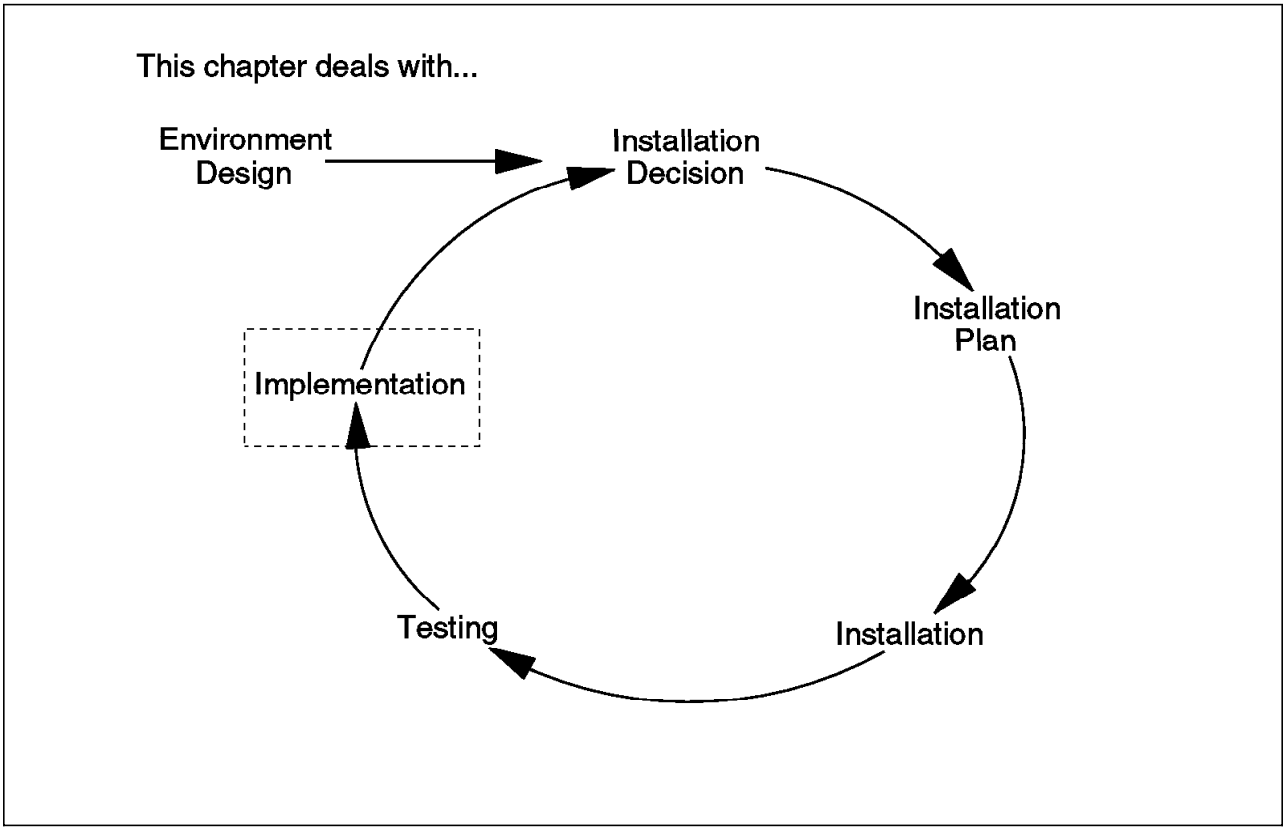


Figure 46. Production and Implementation

Methods discussed in this chapter should be used to plan, test, and implement changes. These could be anything from a single PTF to a complete system and subsystems replacement. All pertinent sources of installation information should be consulted, as required, to determine how each activity should be performed. The intent is to help you develop suitable processes to meet your business needs, using this chapter as a guide to successful implementation. Because of the potential complexity of an OS/390 system, and the uniqueness of almost every production system, there may be additional activities that have to be performed that are not discussed here.

6.2 Moving into Production

Having just installed a new system, subsystem, or product, a question frequently asked is, “What do I need to do now?” It is often correctly assumed that whatever system delivery method you have chosen will leave a fully functional system or subsystem. The problem is that it will not be fully functional for your business needs. To fully implement new functions, some customization will have to be performed.

Decide what customization has to be performed, and how to move the system into production. Cut over from the old system to the new system. What customization will be performed before cut-over (non-disruptive), and what customization will have to be performed after cut-over (disruptive)? These activities should be documented to form an implementation plan. The design of your system will define the level of disruptive activities. The aim of good software management techniques is to minimize disruptive activities.

The implementation plan must be agreed to and incorporated into your change management system. Included in this will be time scales, backup criteria, backout criteria, and a test plan. Again, the aims of good software management techniques ensure that most testing is performed on a test or development system; therefore the amount of disruptive testing required is minimal. Once cut-over has taken place, you will have entered a critical phase, and will have to be aware of the go or no-go breakpoints. Figure 47 shows the steps necessary to move a newly installed system into production.

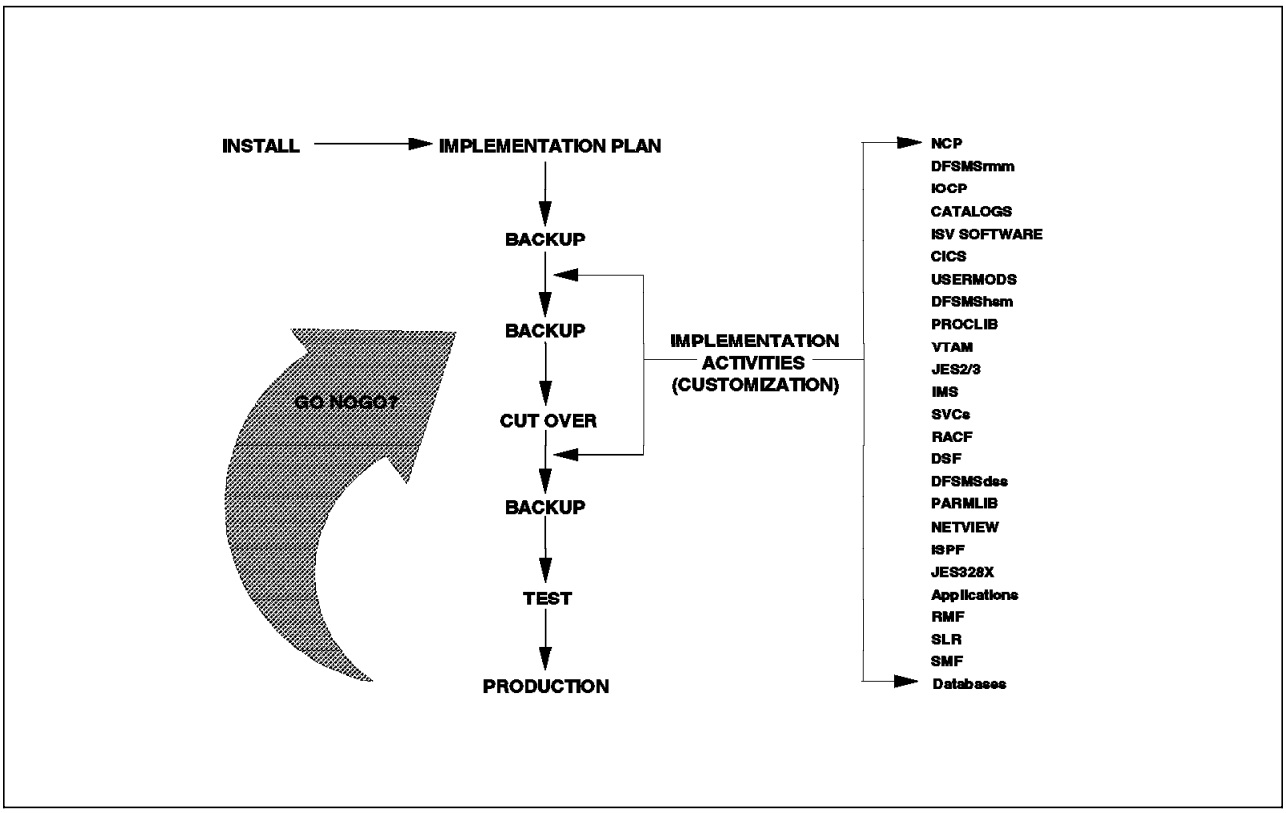


Figure 47. Moving into Production

6.2.1 Backup and Recovery

Before modifying your current test, development, or production system, take a backup. This action will ensure that recovery is possible. Your installation guidelines and specifications should document all backup and recovery actions. These should be tested before modifying your current working environment. Frequent backups should be made of the new system at convenient checkpoints, especially prior to cut-over, and prior to moving into production.

Planning the backout recovery procedure for your installation is necessary to ensure stability for your operating system. These procedures should balance the needs of the business with the assessment of the software needs. Outline in the recovery procedure whether to correct a problem that may occur during the implementation of the production system.

The system upgrade package you decide to install will be designed to operate with maximum reliability. Using the installation procedures provided with your upgrade package prior to, and during, installation should minimize problem determination efforts. However, something might go wrong. All production-system configurations and executing environments are different. If the simple operation of some necessary control does not remedy the problem, it is recommended that you use your backout recovery procedure to quickly restore your system or subsystem to a stable level.

6.3 Customizing the New System

Once you have installed a new operating system, you must customize it. Each production system will have its own unique requirements. Customizing an OS/390 operating system consists of changing or extending your operating system′s standard options to meet your data processing environment′s specifications.

When installing a new system, installation parameters provide a means of tailoring or tuning. Your particular installation requirements may be met by the use of the installation parameters. See details in *OS/390 MVS Initialization And Tuning Reference*.

Note: Do not be misled into attempting to customize your OS/390 operating system on the basis of the partial list provided below. The OS/390 operating system practices at your installation may vary. Therefore, you must compile an individual specification suitable to your installation′s needs. Consider carefully whether to copy existing definitions and update them to the new levels, or to use new definitions.

Typical customizing activities involve the following:

- New I/O definitions, including changes to IODF and to the JES3 INISH stream (in JES3 environments).
- Alter, update, or add PARMLIB definitions such as adding:
  - Libraries to the Authorized Program Facility (APF) list
  - New programs to the Program Properties Table (PPT)
  - New libraries to the LINK or LPA lists
  - Libraries to LLA and VLF
- IBM supplies a tailored SYS1.PARMLIB member, IFAPRD00, that enables the elements and features you ordered. Before you can use them, you must copy the contents of IFAPRD00 to an active IFAPRDxx member. For more information, see *OS/390 MVS Product Management*.
- Updates to the master and user catalogs, discussed in Section 6.4, “Determining the Method for Master Catalog Update.”
- New ISPF panels and dialogs will usually be a replacement of the old by the new. User-written panels and dialogs should be kept in separate libraries.
- TSO/E CLISTs and REXX EXECS can be troublesome and time consuming if the design of the environment allows users to use many different logon procedures and EXECS. This will force users to test their own TSO/E, ISPF, PDF, and SCLM environments. With a well-controlled and well-managed TSO/E environment, customization should not be troublesome, which will therefore reduce the need for user involvement and user testing.
- Install User SVCs. The SVC numbers for installed SVCs and those about to be installed cannot conflict if all are to be used. SVC number conflicts must be resolved (by renumbering one of the conflicting SVCs) prior to installing the user SVCs. Some SVC routines have required names, which differ by SVC Type. Type 1, 2, and 6 SVC routines must be NUCLEUS-resident. The remaining SVC types are LPA-resident. LPA-resident SVCs may be included in LPALIB, put in an LPA list library, or loaded in MLPA from a LINK list library. Nucleus-resident routines must be linked into the nucleus load module, IEANUC0n or added to the nucleus data set and included in a NUCLSTxx PARMLIB member.

- Rewrite any user modifications (USERMODS), and reinstall all needed user modifications on the new system.
- Convert and install user exits. Rewrite any user exits as required, and reinstall all needed user exits on the new system.
- Perform updates required by a new release or version of JES. Some other updates may be needed (for example, those for a different spool data set, checkpoint data set, or procedure library names).
- Perform any updates required by new functions of DFSMS, including those to DFSMShsm, DFSMSdss, or DFSMSrmm, (for example, those necessary to define additional DFSMShsm managed volumes).
- Perform any needed updates to VTAM parameters, which may be required by a new release or version of VTAM.
- Generate a current NCP for the new system. NCP generation decks should be used to regenerate the NCP after initial NCP testing is complete.
- Change the language options for IBM programming languages (for example, High Level Assembler, C/C++, COBOL), from default options to those preferred.
- Alter, update, or add PROCLIB definitions supplied with the system to conform to the existing system, or to the new system.
- Update RACF, for example, by adding new resource classes or adding new started procedure names.
- Update automated operations to reflect new programs, message changes, and environmental (for example, hardware configuration) changes.
- Convert accounting and billing applications for use in the new operating system environment. Accounting and billing application changes are most frequently required by SMF record content or format changes, and changes to SRM. With OS/390 MVS you have the option of processing with your existing IEAIPSxx and IEAICSxx members or use the workload manager service definition. Workload manager uses a goal-oriented management philosophy for MVS systems that allows an installation to state its system performance expectations in common business terminology generally used to document service level agreements (SLAs). Processing with existing IEAIPSxx and IEAICSxx members is called compatibility mode. Processing towards goals defined in the workload management service definition is called goal mode. The latter allows you to switch over to processing towards a service policy to match your own pace. For complete information about the workload management service definition, see *OS/390 MVS Planning: Workload Management*.
- Perform any required updates to the ISV products.

6.4 Determining the Method for Master Catalog Update

Customization activities that have to be performed on the master catalog will depend on the reason for generating a new system. For example, the system may have been created to apply service using a ServicePac or service-only CBPDO; or to apply products and service such as a ProductPac or CBPDO; or to replace a full system using a SystemPac or ServerPac. For a ServicePac, there will probably be no master catalog updates, and if there are, the master catalog can be manually updated. For a ProductPac, the master catalog may need

either updating as for a ServicePac, or for larger changes, it will have to be copied at the same time as the SYSRES. For a SystemPac or a ServerPac, a new master catalog will be generated. Some considerations are:

- The condition of the catalog (available space, integrity, format, and so on)
- The use of direct cataloging, or the indirect cataloging facility
- The classes of data sets that require inclusion in the Master Catalog (for example, SYS1 data sets)
- The possible use of the REPRO MERGECAT IDCAMS facility
- The definition of aliases
- The definition of any new user catalogs
- The possible use of the ServerPac CATCVTM job
- The connection of new or existing user catalogs
- The amount of update activity

To simplify maintenance, the master catalog should contain only those data sets that are necessary to IPL the system; all other data sets should be pointed to by aliases and user catalogs. Section 7.4, “Catalog Environment” on page 159 discusses this concept further with regard to installing subsystems. Section 3.5, “Catalog Environment” on page 61 discusses the catalog setup for the operational and maintenance environments.

6.4.1 Creating a New Master Catalog

When you are doing a system replacement, the new master catalog to be used in production must combine the new entries created by the ServerPac and those in the existing master catalog.

One way to accomplish this is using the CATCVTM job that comes with ServerPac. The output of CATCVTM is placed in a partitioned data set, which will contain the following members:

Member	Description
ALIAS	Contains all ALIAS definitions.
GDG	Contains all Generation Data Group (GDG) definitions.
IMPORT	Contains all IMPORT CONNECT definitions for USER catalogs including the related ALIAS definitions.
NONVSAM	Contains all non VSAM data set definitions except those with a high level qualifier of SYS1 (see SYS1).
MISC	Contains all other VSAM definitions which do not fit into the categories above.
SYS1	Contains all system data set definitions with a high level qualifier of SYS1.
SYSCTLG	Contains OS CVOL catalog definitions.

CATCVTM is run against the master catalog from the production operational environment and the new master catalog generated by ServerPac.

After CATCVTM has completed successfully, you can add, change, or delete entries in the previously mentioned output data set to meet your installation requirements and to clean up your catalog (delete old or unused entries).

If you are using indirect catalog function, the job CATCVTM will produce the output accordingly (VOLUME(******) and DEVTYPE(0000)) and you don’t have to worry about pointing to the data sets in the new SYSRES volume. Otherwise, you must edit the SYS1 member of the CATCVTM output to reflect the new SYSRES volume serial.

Job CATCONV in the CustomPac dialogs will define entries in the new master catalog using the control cards created by CATCVTM as input.

This procedure will transfer NONVSAM, GDG, and CVOL entries from those existing catalogs to your new master catalog and it will also IMPORT CONNECT any user catalogs and their associated ALIASes.

The MISC entries (VSAM objects) require some work if you wish to have the same objects defined in the new master catalog. If you limit the VSAM objects in the master catalog to those that can be recataloged (PAGESPACEs and objects beginning with SYS1), it is fairly straightforward to redefine the VSAM objects in the new master catalog. See Appendix E, “Recataloging VSAM Objects” on page 209 for a set of EXECs to create the definitions.

6.5 Testing Prior to Production

Before moving a new system into production, it should have been thoroughly tested. Chapter 5, “Testing” on page 123 details how to perform testing. Once a new system has been cutover, there will still be some important areas to test prior to production implementation.

6.5.1 Function Test for Major Components and Subsystems

System programmer and user testing are key elements of the propagation cycle. Implementers must evaluate their environment and decide which components and subsystems require testing. The operation of equivalent program products such as Independent Software Vendor (ISV) software should be included in your specific activity list for testing. Your list will depend heavily on the software dependencies your business is based on. Therefore, you must include all applicable software dependencies when compiling your list of resources.

6.5.2 User Community Testing

Evaluation and verification of the operation of your OS/390 operating system is based on your installation′s requirements. You will have to exercise business disciplines such as:

- Coordinate and conduct user testing on the new production system. This includes scheduling through Change Management and performing system IPLs.
- Communicate new procedures to the user community.

6.6 Propagating a System into Production

The amount of work required to propagate the upgraded maintenance level to production depends on where the target libraries are placed and how they are cataloged.

For the purpose of discussion for this section, imagine you have one upgraded maintenance level where the target libraries spill off the SYSRES volume. For the target libraries on the spill volume, you should consider having them cataloged in a user catalog instead of the master catalog to facilitate their migration into the production environment.

We will consider the possibility to propagate the SYSRES and spill volume independently. This can happen when you are applying maintenance that affects

only one volume at a time. When you are replacing the system, the two following actions listed can be combined.

1. To clone (copy) the SYSRES volume, the easiest method is to run DFSMSdss and perform a logical volume copy (copy by data set), taking care to exclude the SMP/E target CSI if you decided to keep it in the SYSRES volume. Having the SMP/E CSI in the SYSRES, redefine a new SMP/E target CSI and use IDCAMS REPRO command to copy the data across, or, alternatively, use IDCAMS Export/Import. If there are many VSAM data sets on the SYSRES, propagation could be more difficult and more time consuming, as these will all need managing in a similar way to the SMP/E target CSI. Placing VSAM data sets on the SYSRES should be avoided.

   With the indirect catalog function, you can introduce the new SYSRES volume into production using the current master catalog. In a Parallel Sysplex this will also allow master catalog sharing while doing the ripple IPL through images using the two different level of SYSRES.

   Since OS/390 provides full compatibility between N and N+2 releases, ripple IPL allows one system at a time to be shutdown with the following IPL using the most recent OS/390 level. This provides continuous operations in a Parallel Sysplex environment. After IPL on all the OS/390 images, we will be using the new shared SYSRES and the shared master catalog on all the running systems. Figure 48 shows a new SYSRES volume being propagated into the production sysplex, where system C will IPL using the new SYSRES volume and the current master catalog and spill volume.

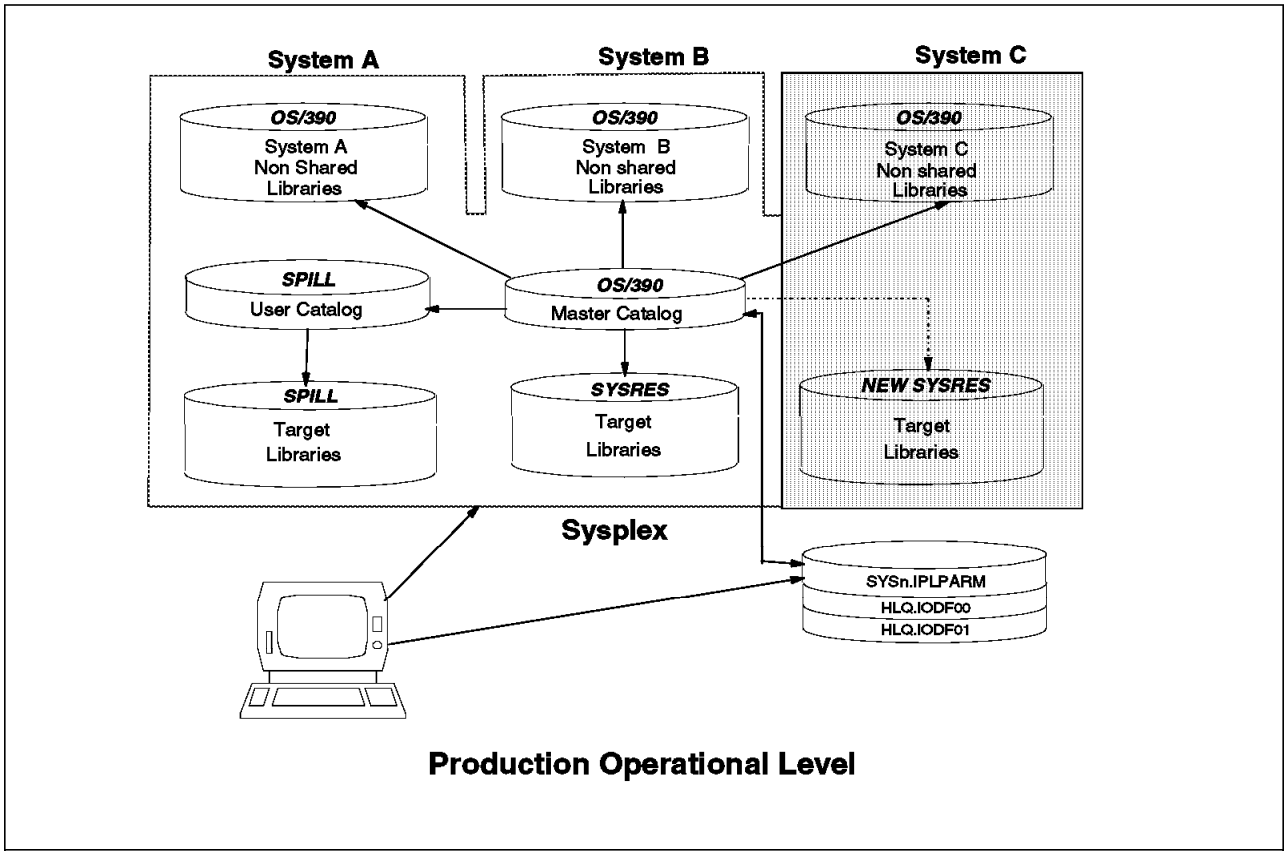


Figure 48. Propagation of SYSRES Volume

2. To clone the spill volume, redefine a new user catalog on the new volume and use DFSMSdss to perform a logical volume copy (copy by data set) with the option RECATALOG*(newusercatalog)*.

   To introduce the new spill volume in one image at a time in a Parallel Sysplex environment with a shared master catalog, you have to clone the current master catalog and update it to point to the new user catalog on the spill volume. This two level master catalog will last until all the OS/390 images are IPLed with the most recent level of software. During this time, some effort must be made to keep the two levels of master catalog synchronized. When updating the master catalog, it is imperative that records of all changes are kept to facilitate backout and recovery. Figure 49 shows a new spill volume being propagated into the production sysplex, where the system C will IPL using the new master catalog, the new spill volume and the current SYSRES.

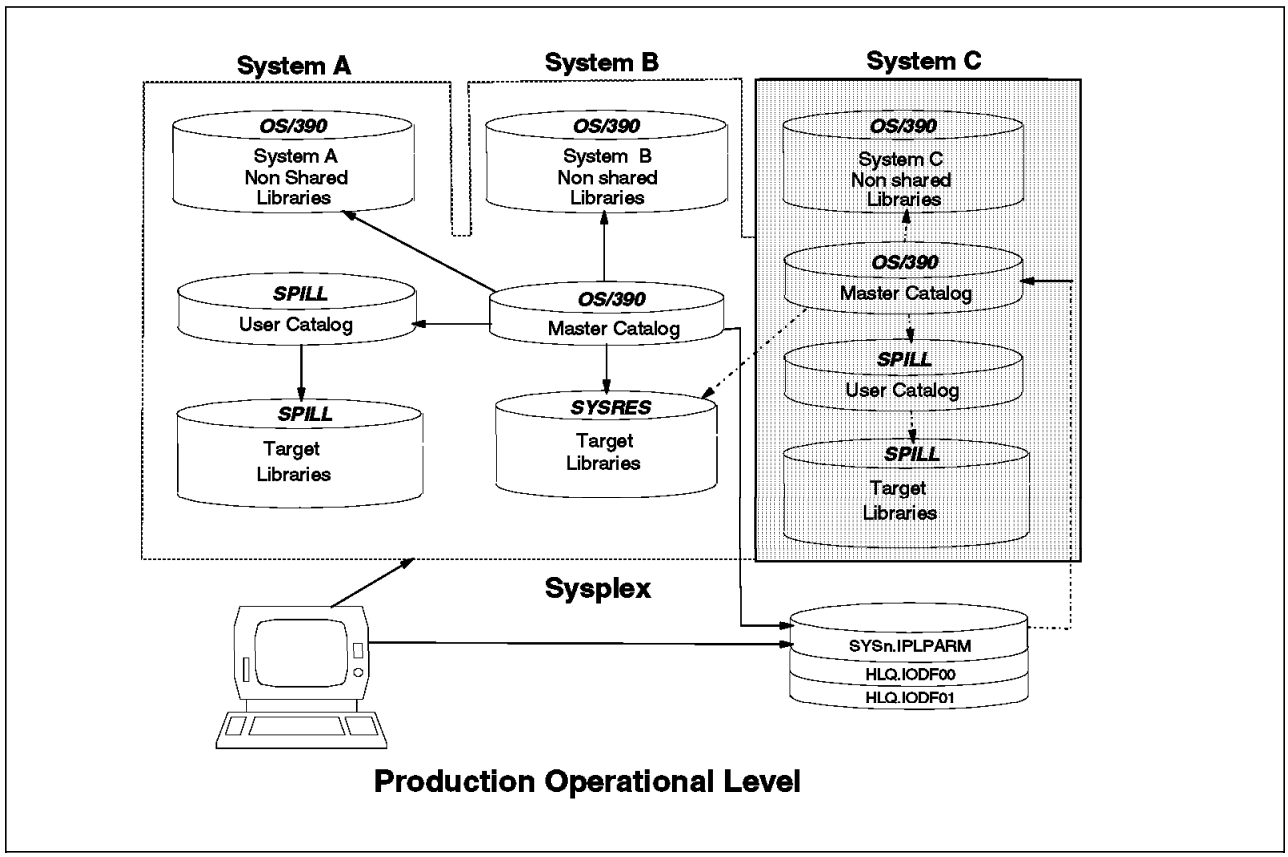


Figure 49. Propagation of Spill Volume

To identify the new master catalog, either a new LOADxx must be created containing the new catalog name (with a corresponding change to the load parameter to point to the new LOADxx), or the existing LOADxx must be updated to contain the new catalog name. The following section describes how to perform these tasks.

6.6.1 The LOAD Parameter and LOADxx

The LOAD parameter allows the user to effectively select the I/O configuration, master catalog, NUCLEUS and the SYSPARM suffix when IPLing the system. This parameter is entered at the hardware system console in the Operator Control Frame, System Control Frame for ES/9000 processors and using LOAD profiles for a 9672 Parallel Transaction Server and ES/9221 processor.

The LOAD parameter is a string of eight characters that holds the following information:

1. The first four characters specify the device number of the DASD volume that contains the I/O definition files (IODF). This is also the device on which the search for the LOADxx member of SYSn.IPLPARM or SYS1.PARMLIB begins. If you don't specify the device number, the system uses the device number of the system residence volume.
2. The fifth and sixth characters form the suffix of the LOADxx member in either SYSn.IPLPARM or SYS1.PARMLIB. Placing LOADxx members in SYSn.IPLPARM rather than SYS1.PARMLIB means that the IPL process will be faster, as this is the first data set searched.
3. The seventh character defines the prompting and message suppression characteristics that the system is to use at IPL. This character is commonly known as an initialization message suppression indicator (IMSI).
4. The last character defines the alternate nucleus identifier (0 - 9). The default member is IEANUC01.

As described above, the volume defined in the load parameter is first searched for data set SYSn.IPLPARM, where the "n" can be any number from 0 to 9. When found, the LOADxx member is used to further control the IPL process. When a SYSn.IPLPARM data set cannot be found, the same volume is searched for SYS1.PARMLIB. When found, the LOADxx member is used to control the IPL. If neither a SYSn.IPLPARM nor a SYS1.PARMLIB is found on the defined volume, the IPL volume will be searched for SYS1.PARMLIB to fetch the control information from LOADxx. When the LOADxx member is placed in SYSn.IPLPARM, the data set must reside on the same volume as the IODF data set.

In the LOADxx member, use the SYSCAT statement to identify the master catalog. The operator can override the value specified on this parameter, using the LOAD parameter on the system console with an appropriate initialization message suppression indicator (IMSI).

An exhaustive description of the load parameter can be found in *OS/390 MVS System Commands*. The content and use of the LOADxx member is described in *OS/390 MVS Initialization and Tuning Reference*.

6.6.2 Production Implementation

Propagating from the least critical to the most critical image and following your production policy for implementation are key elements in this stage of the propagation cycle. The production policy provides the standards for modifying your current operating system. Included in this policy will be specific instructions on certain tasks and events during the operating system upgrade cycle. Tasks and events include, but are not limited to:

- Performing SPOOL OFFLOAD, RELOAD, and JES cold start, if required.

- Implementing the final PARMs, CLISTs, ISPFs, RACFs, and so on.
- Communicate operational changes: IPLs, commands, IPL sheets, and so on.

Chapter 7. Subsystem Topics

This chapter deals with topics related to subsystems software management, including the following:

- Maintenance environment for subsystems
- Operational environment for subsystems
- Catalog environment for subsystems
- CICS subsystem replacement
- DB2 subsystem replacement
- IMS subsystem replacement
- NCP subsystem replacement
- Subsystem cloning

7.1 Introduction

Figure 50 is an overview of the areas discussed in this chapter. This chapter covers almost all aspects of the software management cycle. Ideas presented in this section will assist you in forming your own subsystem management strategy. For topics relating to subsystem testing, refer to Chapter 5, “Testing” on page 123.

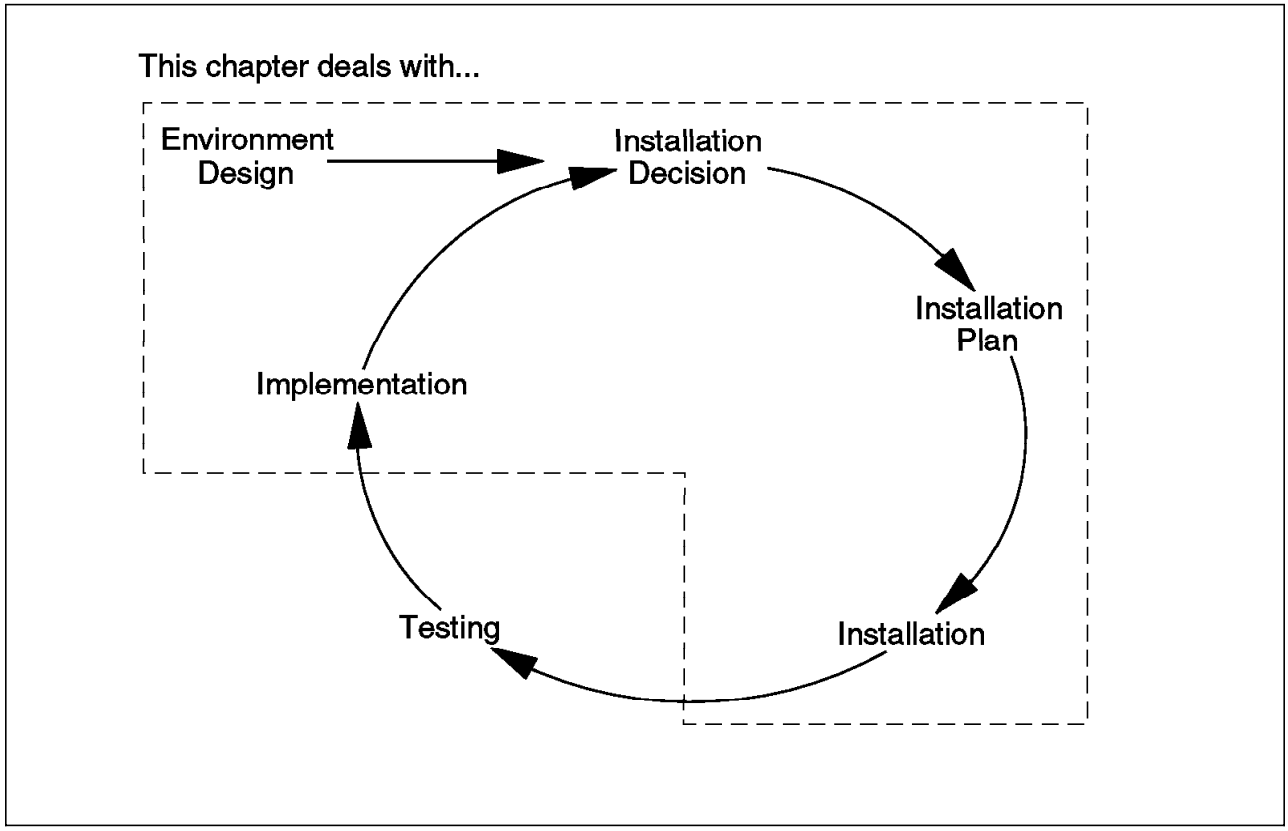


Figure 50. Subsystems

In the preceding chapters, the subsystems were in most cases not explicitly discussed. This does not imply that most recommended approaches for maintaining OS/390 systems are not also true for subsystems.

The main difference is that subsystems depend on the services and functions provided by an OS/390 system. An error in a new level of a subsystem does not necessarily cause an IPL of the system nor does it mean that other subsystems will suffer an outage. The error may prevent you from starting the subsystem, but after correcting the error, most likely, you will have your subsystem back to normal (in some cases, an IPL with the CLPA option may be necessary).

The terms used in this chapter to describe the various components are defined in 3.1, “Term Definitions” on page 40.

7.2 Subsystem Maintenance Environment

As discussed in Chapter 3, “Maintenance Environment” on page 39, for subsystems, it is recommended that the maintenance environment consists of two maintenance levels; namely:

- The production maintenance level
- The upgraded maintenance level

Each of the maintenance levels of an installation could have the following subsystem SMP/E product sets:

- Data Base Subsystems (DBS):

  DBS consists of two products, IMS and DB2, which use the same SREL in SMP/E. When you order the DBS subsystem using ServerPac, you may order one package containing IMS and DB2. In that case, IMS and DB2 would share the same target zone and distribution zone. Even with SystemPac you can now order IMS and DB2 together. However, if you choose to replace your subsystem, it is recommended that you order ServerPacs for IMS and DB2 separately. This provides flexibility in replacing or upgrading IMS and DB2 independently:

  – A production maintenance level for IMS and an upgraded maintenance level for IMS
  – A production maintenance level for DB2 and an upgraded maintenance level for DB2

- CICS Subsystems:

  – A production maintenance level for CICS and an upgraded maintenance level for CICS

- NCP Subsystems:

  – A production maintenance level for NCP and an upgraded maintenance level for NCP

Each SMP/E product set consists of:

- An SMP/E target zone and its associated target libraries
- An SMP/E distribution zone and its associated distribution libraries
- A user catalog containing entries for all the above

To replace or upgrade a subsystem in the maintenance environment, follow the recommendations discussed in Chapter 3, “Maintenance Environment” on page 39, and Chapter 4, “Installation” on page 75, especially section 4.2, “System Upgrade” on page 89. Subsystem product sets generally contain only a few products (for example DB2 and QMF) compared to the OS/390 product set. New versions and releases of such subsystems normally require that you upgrade related products to use new functions and services. Customizing the subsystem to use new functions is also usually required. The subsystem replacement offering that you choose often provides assistance with that customization.

The first decision you have to make for the subsystem maintenance environment is whether to use the SMP/E global zone of the OS/390 system maintenance environment, or to build a separate maintenance environment for each subsystem. There are some pros and cons around this discussion, and it is really hard to tell which is better. It depends on your environment and organizational situation. For example, a single global zone for everything could simplify the software management process for less complex environments, but it is not quite suitable for a complex environment that needs flexibility for changing. The following discussion describes some considerations that you may use for maintaining the maintenance environment with either a separate global zone for each subsystem or a single global zone.

7.2.1 Separate Global Zones

Some considerations:

- It may be easier for separate organizations or functions that are responsible for one subsystem to maintain their own SMP/E environment.
- It may be easier to replace or maintain each of the subsystems independently.
- Each group that is responsible for each subsystem can run SMP/E jobs concurrently with no contention on separate SMP/E global zones.

Figure 51 depicts the maintenance environment using separate global zones.

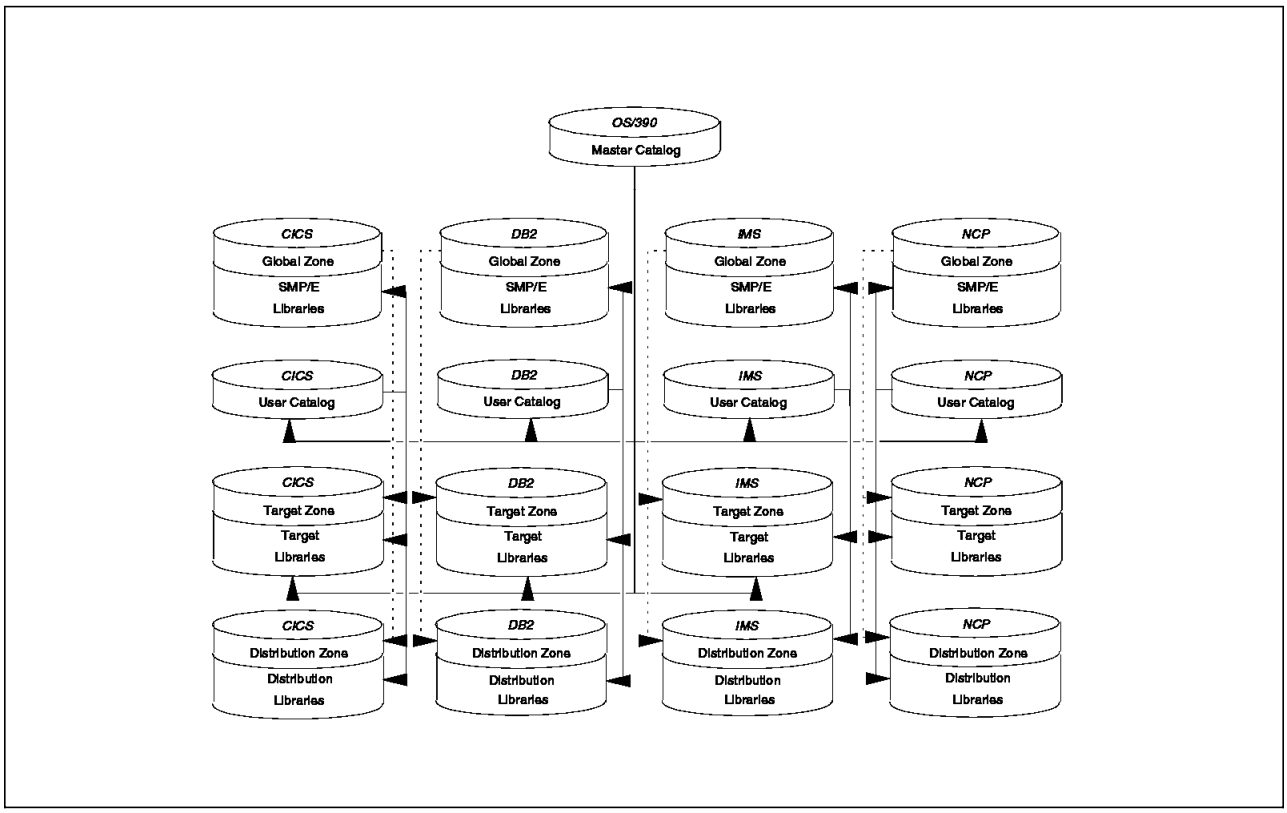


Figure 51. Subsystems Maintenance Environment Using Separate Global Zones

One of the drawbacks to using a separate global zone is that it is not easy to perform the ERRSYSMODS and CROSSZONE reporting.

ERRSYSMODS and CROSSZONE are SMP/E REPORT command options that do the following:

- ERRSYSMODS - Helps you determine whether any SYSMODs that you have already processed are now exception SYSMODs. It also helps you determine whether any resolving SYSMODs are available for held SYSMODs.
- CROSSZONE - Helps you synchronize the service levels of different products when the products are installed in different zones.

Both reports need a global zone that contains ZONESET entries for all the target zones that you want to report on. This presents a problem if your target zones are connected to multiple global zones.

One possible solution is the use of a dummy global zone. This dummy global zone will contain all of the target zones that we want to include in both reports. The following steps could be used for preparing the dummy global zone and running the SMP/E reports:

1. Define a global CSI data set for the dummy global zone.
2. Prime this CSI using GIMZPOOL.
3. In this dummy global zone, add entries for all of the subsystem target zones (and system target zone), that you want to report on in the ZONEINDEX.
4. Put the target zone names that will be included in the SMP/E reports in the ZONESET entries.

5. Prepare the SMPHOLD data in this dummy global zone up to current levels. SMARTMVS HOLD data is a convenient way of receiving this HOLDDATA on a regular basis. See Appendix B, “SMPEUTIL” on page 197 for more information on SMARTMVS HOLD data.
6. Receive all the HOLD data into the dummy global zone.
7. Run the REPORT CROSSZONE or REPORT ERRSYSMODS command using the prepared ZONESET. Refer to Appendix B, “SMPEUTIL” on page 197, *SMP/E Reference*, and *SMP/E User Guide* for detailed information on these commands.

Figure 52 shows the dummy global zone implementation for SMP/E reporting. It is recommended that you also include the OS/390 target zone in the ZONESET for a complete report on the health of all the SMP/E installed products in the complex.

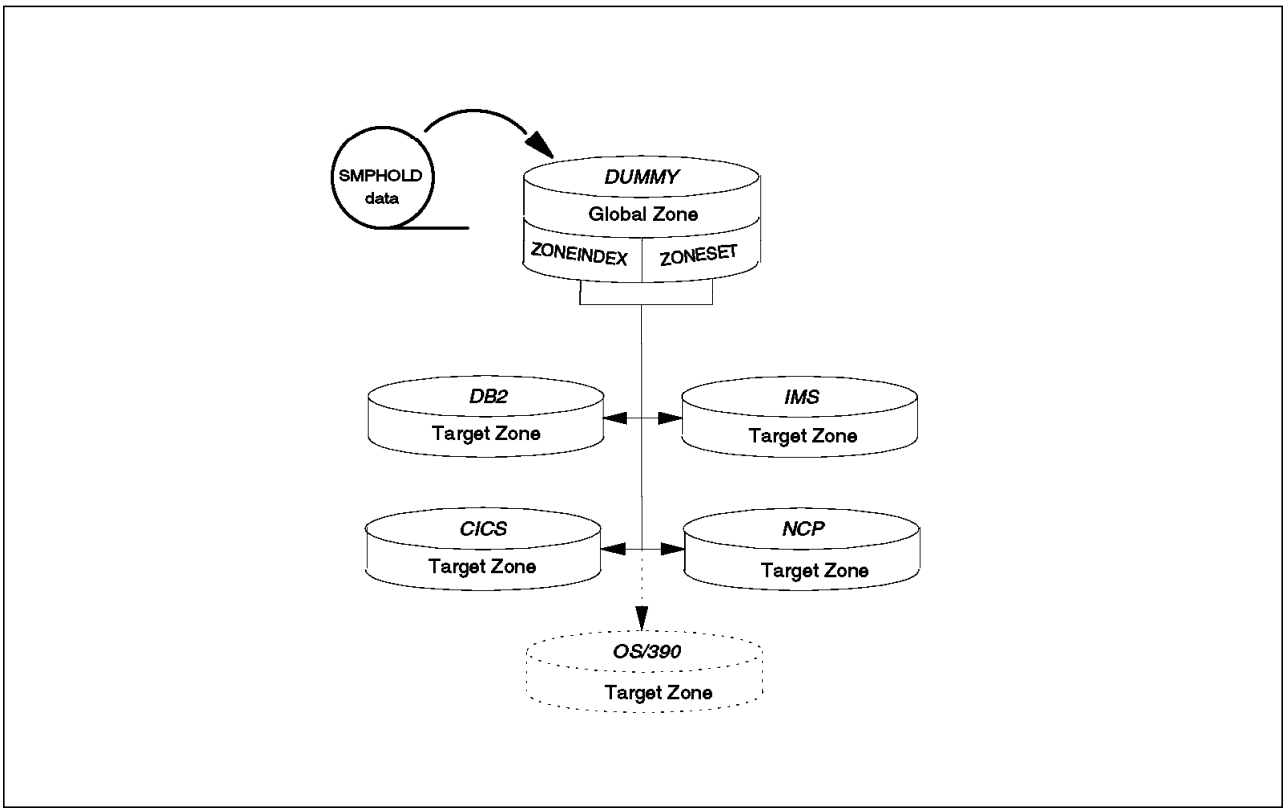


Figure 52. Dummy Global Zone Implementation of SMP/E Health Reports

Refer to section 3.3, “Logical Design” on page 43 for a further discussion of the advantages and disadvantages of using separate global zones.

7.2.2 Single Global Zone

Some considerations:

- For a small organization that supports multiple subsystems, this configuration would make the SMP/E environment easier to maintain.
- It provides a centralized point of all SMP/E information.
- It is ready to run SMP/E health reports at all times.

One of the drawbacks to using a single global zone is the contention on the SMP/E global zone if you have to run concurrent SMP/E jobs for update. If the system and subsystems are maintained by different functions or organizations, this approach could be a disadvantage.

Refer to section 3.3, “Logical Design” on page 43 for a further discussion of the advantages and disadvantages of using a single global zone.

Figure 53 depicts the maintenance environment using a single global zone.

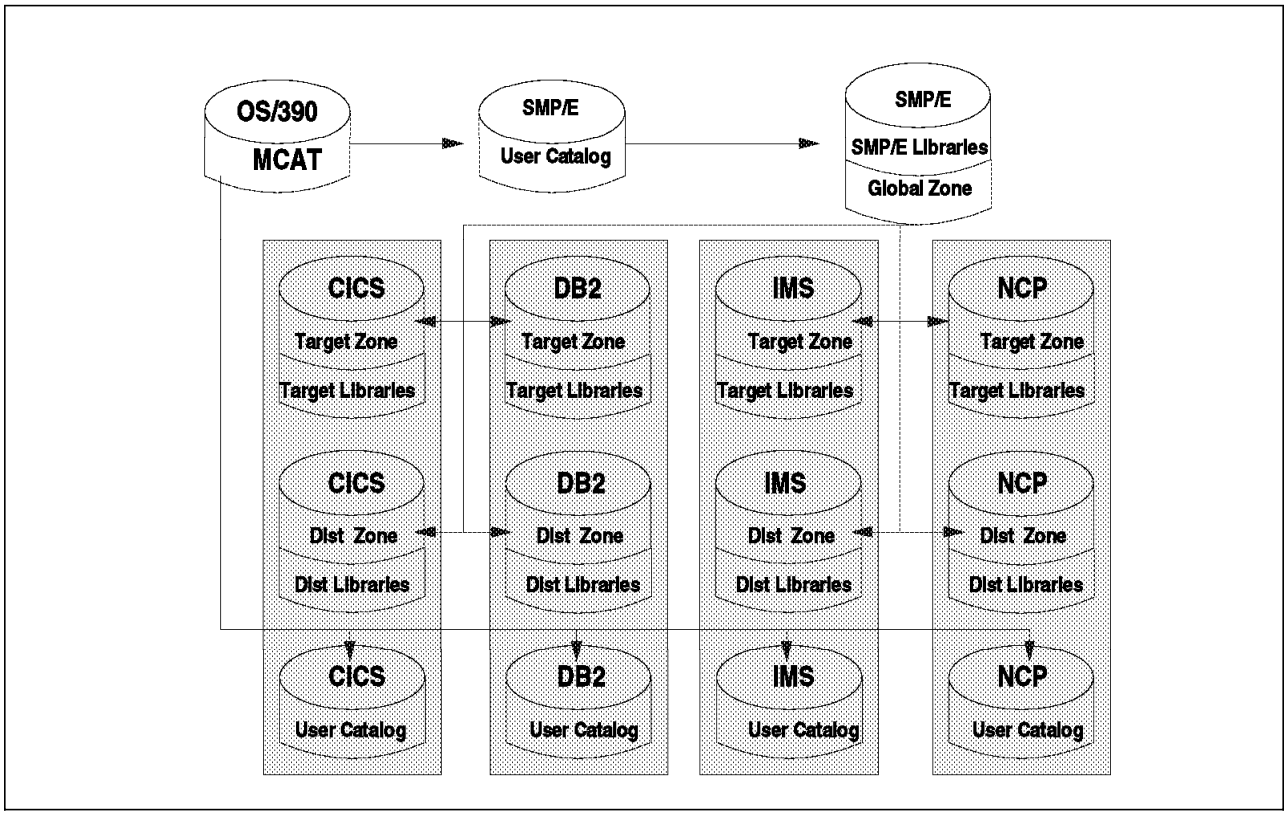


Figure 53. Subsystems Maintenance Environment Using a Single Global Zone

Another potential consideration when using a single global zone is that the CustomPac installation process builds a separate global zone for each subsystem. If you use CustomPac to install your subsystem and you want to share the SMP/E global zone between subsystems, the CustomPac jobs will have to be modified accordingly.

ServerPac and SystemPac uses pre-generated SMP/E zones (CSIs); thus it is not easy to modify the global zone to comply with the single global zone structure. Should you decide to use a single global zone, the following CustomPac jobs should be carefully examined:

- SMPALOC - Delete and define the global zone CSI, target zone CSI, distribution zone CSI, and all SMP/E-related data sets.
- LOADCSI - Copy or reproduce the CSIs from tape into the allocated CSIs.
- UPDCSI - This step updates the new system′s CSI′s in the following way:
  - It puts the correct TARGET and DLIB zone names into the GLOBAL zone
  - It puts the correct RELATED DLIB zone name into the TARGET zone

- It puts the correct RELATED TARGET zone name into the DLIB zone
- It updates all DDDEFs with the correct data set names
- It removes the UNIT and VOLUME definitions from all DDDEFs

Be careful when running UPDCSI. The default for the first zone is “REP GZONE,” which will wipe out any existing definitions. Thus user should change this to “ADD GZONE” when using a single global zone.

- UPDDDD - This step updates all SMP/E owned DDDEFs (like e.g. SMPPTS, SMPSTS, SMPMTS, SMPLTS etc.) in the target system′s CSI′s (GLOBAL, TARGET, DLIB). The dataset names for the CustomPac (ServerPac or SystemPac) can be changed by using the change option included in the CustomPac Installation Dialog.
- UCLIN - Update SMP/E with customized values. This is only supplied for certain subsystems.

If you decide to use a single or shared global zone, make sure that the following actions are done properly:

- Take out all parts of the SMPALOC and LOADCSI jobs that delete, define, and copy/repro the global zone and SMP/E global zone related data sets.
- Take out all parts from the UCLIN job that modifies the SMP/E global zone related DDDEFs, such as SMPLOG, SMPLOGA and so on.
- Add the new entries in the global zone UCLIN job for:
  - FMID - The program product FMIDs that are being installed.
  - SREL - The Software RELease identifier for relating program product.
  - OPTIONS - The processing options that are to be used by this program product.
  - ZONEINDEX - The target and distribution zone names and libraries that are to be used. Make sure the zone names are not currently being used for other target and distribution zones.

  See the *SMP/E Reference* for more information on Options and ZONEINDEX.

These jobs could corrupt the existing SMP/E environment if not properly tailored, so take precautions prior to submitting them.

The dialog panel allows you to modify the supplied jobs, or to rebuild them. Since modifications to the supplied jobs in the dialog are not saved, it is recommended that you create your own jobs in the dialog for any modifications. Both the *SystemPac Installation Guide* and *ServerPac Installation Guide* contain the detailed job descriptions and should be used in conjunction with this process.

7.3 Subsystem Operational Environment

Figure 54 shows the logical subsystem operational environment, which consists of the following:

- Upgraded and operational maintenance levels
- Multiple subsystem images of the production operational level, including the operational data sets
- Production operational subsystems, target libraries, and backup

- Extended Recovery Facility (XRF) shared data sets

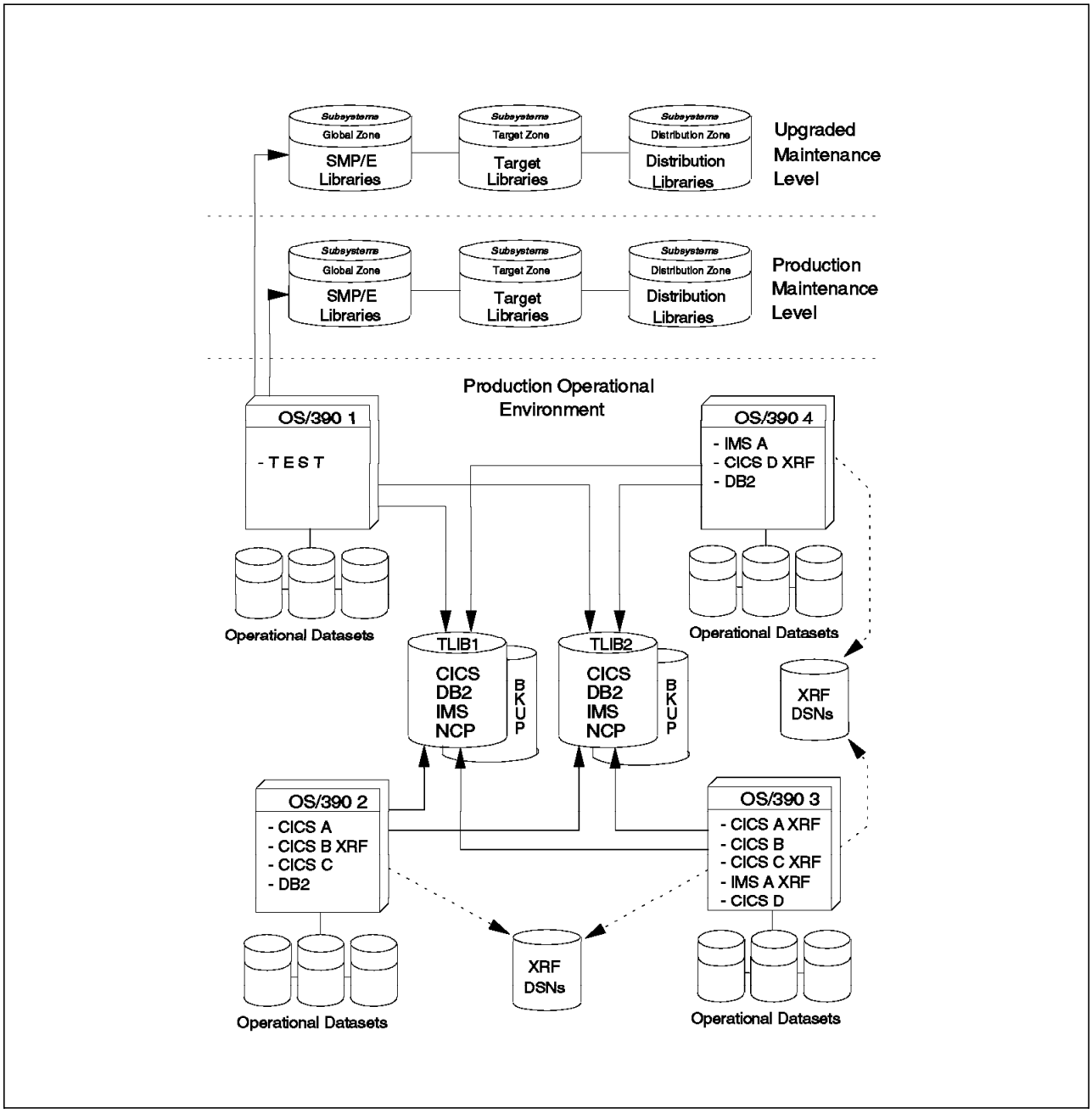


Figure 54. Logical Subsystems Operational Environment

An operational environment differs from a maintenance environment in subsystem availability and performance. Concurrent maintenance in an operational environment must be done with care. Any disruption in its operation must be minimized.

As shown in Figure 54, the operational environment consists of only subsystem target libraries, with an exception for IMS, which also needs its distribution libraries for IMS system definition. These target libraries are propagated from the maintenance environment, which has been tested using the OS/390 1 test system. Refer to section 7.9, “Subsystem Cloning” on page 186 for a detailed discussion of the propagation process.

To minimize the down time during maintenance propagation or in cases of DASD failure, two sets of target libraries are recommended. These two sets of target libraries are also meant for CICS and IMS XRF implementation. As shown in Figure 54, OS/390 2 and OS/390 3 are CICS XRF shadows of each other; therefore OS/390 2 and OS/390 3 must use separate TLIBs for each CICS. For example, OS/390 2 uses TLIB1 while OS/390 3 uses TLIB2. The same situation exists between OS/390 4 and OS/390 3 for IMS XRF implementation.

The backup volumes for each TLIB could be a dual copy backup, which is a hardware (DASD controller) feature. These backup volumes are used to implement the new updated subsystem that is propagated from the production maintenance level. For example, with TLIB2, the new upgraded subsystem can be copied onto the backup volume, then the OS/390 3 workload (which is using TLIB2) is moved to their respective XRF shadows. OS/390 3 can then be IPLed or started using the TLIB2 backup volume, which now becomes the primary volume. If something were to go wrong, it is still possible to fall back using the old TLIB2 volume. If everything goes well, the workload can be moved back to the OS/390 3 system where the old TLIB2 volume will become the dual copy backup volume. By using this method, there should be no disruption during propagation.

For further details on XRF processing, refer to *IMS/ESA Version 3 Customization Guide* and *CICS/ESA Customizaton Guide*.

Parallel Sysplex has a great commitment to high availability during the maintenance process. All current levels of IBM subsystems (CICS, DB2 and IMS/DB) are able to distribute the workload on different cloned elements running on different OS/390 images (this is known as “data sharing”). This gives the capability of shutting down an element and restarting it with the upgraded code. Toward this end, in a Parallel Sysplex configuration, three consecutive releases of OS/390 can coexist, providing full compatibility. As of April 1996, IBM has made the following commitment: “Two consecutive releases of key subsystems running on OS/390 can coexist within a Parallel Sysplex.”

7.4 Catalog Environment

It is recommended that subsystem data sets are cataloged in their own user catalogs. During the CustomPac installation process you have the option to chose on which volume or which high level qualifier to use for your subsystem DLIB and TLIB data sets.

The subsystem libraries do not use a high-level qualifier of SYS1 and, for most subsystem libraries, there is no reason to catalog them in the master catalog. The only subsystem libraries that have to be cataloged in the master catalog are those that are included in the LPALSTxx member of SYS1.PARMLIB. A simple naming convention for these data sets would be to use SYS1 for the high-level qualifier and use the subsystem name as a variable for the middle-level qualifier (for example, SYS1.CICSnnn.LPALIB). This will help keep the number of definitions in the master catalog to a minimum, which in turn simplifies the process of cloning a master catalog.

There is no need to catalog any of the NCP libraries in the master catalog.

For all other data sets, define aliases in the master catalog and relate them to the subsystem user catalogs. To differentiate between different maintenance

levels and operational levels, use unique aliases. The multi-level alias function of MVS/ESA allows more granular naming conventions for your subsystem data set names than are available with a single-level alias.

You may have one user catalog for each subsystem product set, as shown in Figure 51 and Figure 52. It is recommended that you put the subsystem SMP/E data sets (CSI, SCDS, MTS, and others) under their own user catalog in either single or separate global zone environments. Refer to section 3.5, “Catalog Environment” on page 61 for further discussion of the catalog environment.

7.5 CICS Subsystem Replacement

To replace a CICS subsystem, use either of the following:

- CICS ServerPac
- CICS SystemPac

The term “replacement” does not just mean to replace the existing subsystem; it could also mean a new CICS installation.

7.5.1 CICS ServerPac

CICS ServerPac is a package that is free once you have an OS/390 license. You need a license for the CICS products contained in the ServerPac as well. It is available in a dump-by-dataset format and provides a system that is pre-tested with OS/390 during System Integration Test.

CICS ServerPac uses installation dialogs to install the package. The dialogs simplify the task of installing and customizing the package. The ServerPac installation dialog includes the following package-related sections:

- Installation
- Customization
- Migration
- IVPs
- Customer-specific installation tasks

Using the supplied installation dialogs, you can tailor the install to your existing CICS environment. This flexibility makes it unnecessary to reorganize the new subsystem after SystemPac installation.

CICS ServerPac can be used to replace an existing CICS subsystem or to install a CICS subsystem for the first time.

7.5.2 CICS SystemPac

CICS SystemPac is a fee-based software package consisting of installed products for a ready-to-run CICS subsystem. It is available in a Full-Volume-Copy or Dump-by-dataset format which is device-independent.

The package has been customized in response to information provided to IBM or by using product-supplied defaults. CICS SystemPac can be used to replace an existing CICS subsystem or to install a CICS subsystem for the first time.

CICS SystemPac uses installation dialogs to install the package. The dialog simplifies the task of installing and customizing the package. The SystemPac installation dialog includes the following package-related sections:

- Installation

- Customization
- Migration
- IVPs
- Customer-specific installation tasks

Using the supplied installation dialogs, you can tailor the install jobs to match your existing CICS environment.

7.5.3 CICS Volume and Catalog Structure

In both CICS ServerPac or SystemPac you can put the TARGET and DLIB data sets on several volumes. To reduce the number of volumes needed, you can change the VOLSER value in the dialogs to use the same volume name at unload time.

Figure 55 shows an example of a CICS SystemPac catalog structure.

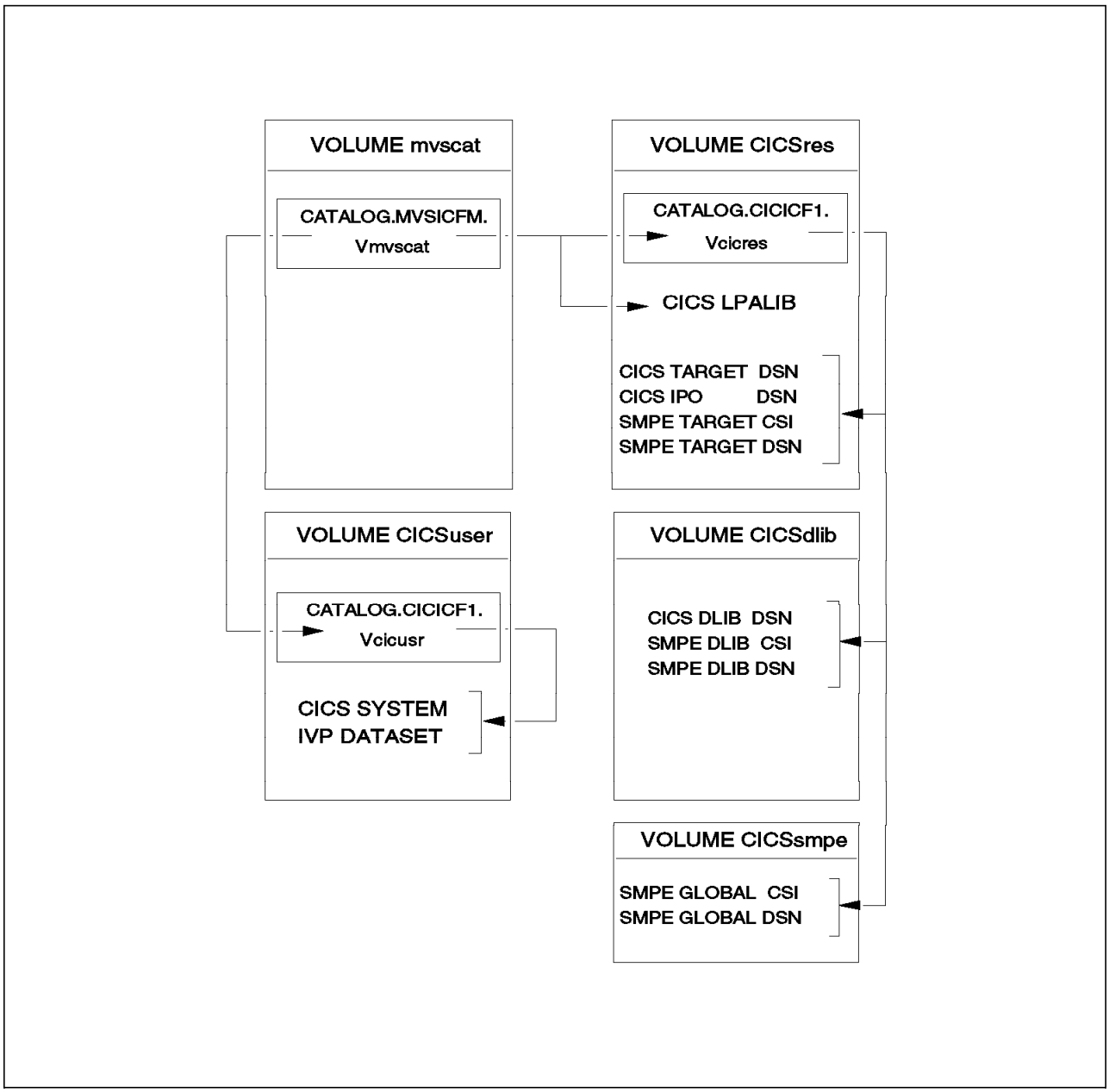


Figure 55. CICS SystemPac Catalog Structure

Note that CICS LPALIB has to be cataloged in the master catalog. The default name for the CICS LPALIB library is:

- SYS1.CICSnnn.LPALIB with ddname SDFHLPA for CICS LPALIB

The last qualifier must match the default SMP/E DDDEF name for the library. If you decide to change it, make sure you do all the tailoring needed in the supplied jobs.

One way to solve the problem when you have to catalog multiple data sets with the same name, is to use System-Specific Aliases (SSAs). These aliases are used to locate any data set, systemwide, even if it has the same name, without requiring a JOBCAT or a STEPCAT. This is discussed in more detail in Section 3.5.2, “System-Specific Aliases” on page 63.

7.5.4 CICS SMP/E Zone Structure

The CICS CustomPac installation processes provide for the implementation of CICS with its own global, target, and distribution zones. This default CustomPac installation process defines a separate global zone for each offering that is installed. Figure 56 shows this structure using a CICS SystemPac.

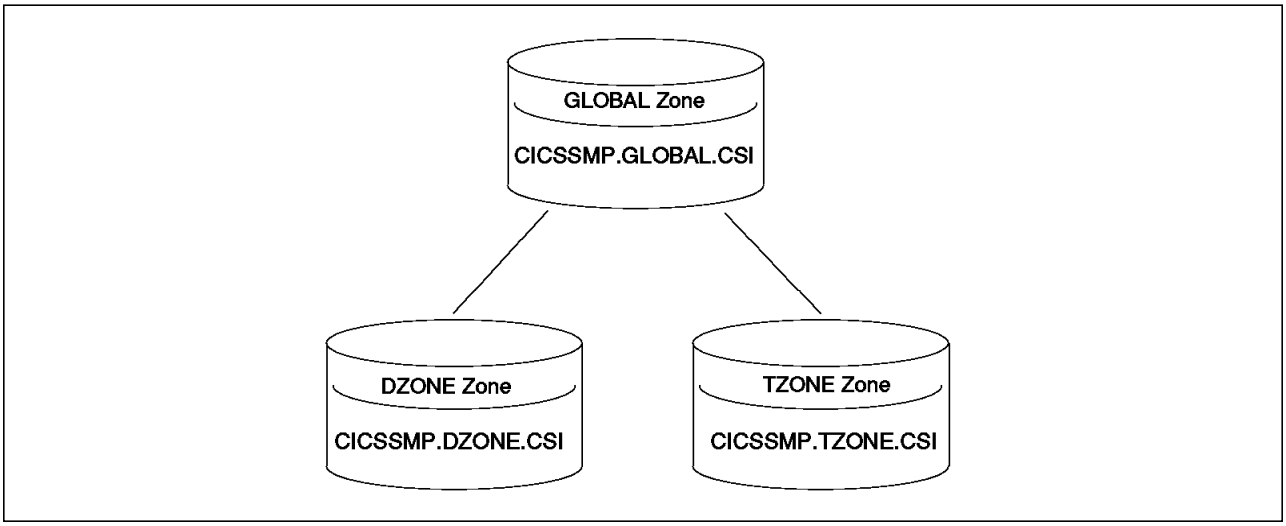


Figure 56. SMP Zone Structure for CICS SystemPac

Each zone is represented by a single CSI. The data set and zone names are sample names. The names of the zones and of the CSIs can be easily changed by entering the new names in the installation dialog panels.

Using a single global zone approach, as one of the alternatives, requires modifications to the ServerPac or SystemPac implementation process. This is because the CICS ServerPac or CICS SystemPac installation process, by default, always defines a unique global zone for each subsystem. Whether you use single or separate global zones, each of the subsystems will be implemented with its own target and distribution zones (target zone and distribution zone are SREL-dependent).

As discussed earlier, you may modify the ServerPac or SystemPac installation process to comply with the single global zone structure. If you do so, you have to use unique target zone and distribution zone names, making sure they are not defined in the single global zone to describe other target and distribution zones. Refer to section 7.2.2, “Single Global Zone” on page 155 for further information. Since some jobs could corrupt the existing SMP/E system if not properly tailored, be careful when submitting them. It is suggested that you make your own tailored jobs to substitute the CustomPac-supplied jobs, since modifications in supplied jobs are not saved. However, it is possible to save a copy of the tailored JOB in the SCPPBENU PDS. You can then edit this data set as you do any other data set. Refer to *CICS ServerPac Installation Guide* and *CICS SystemPac Installation Guide* for more information regarding this modification.

Figure 57 depicts the single global zone structure for the CICS maintenance environment.

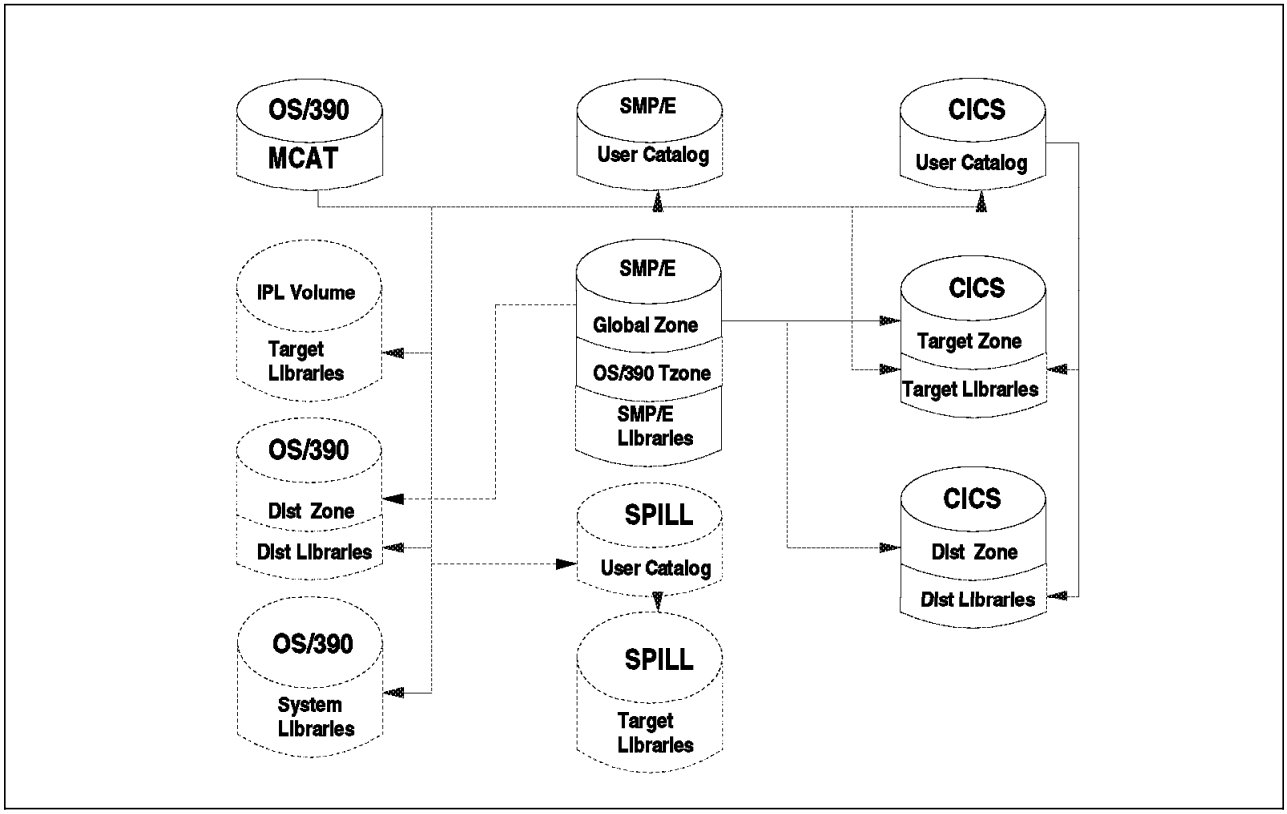


Figure 57. Single Global Zone Structure for the CICS Maintenance Environment

The CICS SMP/E related data sets can be cataloged either in the CICS user catalog or in the SMP/E user catalog. Unique, meaningful qualifiers should be used.

7.5.5 CICS Installation and Post-Installation Tasks

The CICS CustomPacs are installed using the installation dialogs delivered with each offering. The installation dialogs contain everything you need to successfully install CICS. You may choose to use the information provided online rather than the hardcopy supplied in the box. The Installation section, which is made up of the package-specific install, product-specific install, and post-installation and customization, is quite easy to follow. Just make sure that you perform the variable tailoring (and job customization, if any) correctly. At this point in the installation you should have:

- CICS user catalogs and their aliases created.
- CICS SMP/E related data sets (CSIs and all non-VSAM data sets) established.
- CICS distribution libraries allocated and loaded.
- System (target) libraries allocated.
- CICS XDFHINST library created, which contains tailored post installation and installation-verification jobs from SDFHINST.
- CICS LPALIB and LINKLIB built and cataloged in the master catalog.
- CICS SVCs in the NUCLEUS linked and the entry in IEASVCIC (member of SYS1.PARMLIB) updated.
- SYS1.PARMLIB, SYS1.PROCLIB and SYS1.VTAMLST updated.

- RACF definitions defined, if external security is used.
- CICS common data sets, CICS system primary data sets and CICS message data sets created.
- IVP jobs installed.
- CICS product related information (PGMDIR, PSPINFO and HOLD information) loaded.

All that is left to make this newly installed CICS ready for verification using the IVP jobs is:

- IPL CLPA - To activate the CICS LPALIB modules concatenated in the LPALSTnn and also CICS LINKLIB modules concatenated in LNKLSTnn. Both LPALSTnn and LNKLSTnn are SYS1.PARMLIB members.
- Activate CICS APPLID - To activate the application program major node (APPLID) in a member of SYS1.VTAMLST.

7.5.6 CICS Installation Verification Process

CICS IVP jobs are comprised of:

- CICS Batch IVP Job (DFHIVPBT)
- CICS Online IVP Job (DFHIVPOL)
- CICS XRF IVP Job (DFHIVPAL)
- CICS DL1 Support IVP Job (DFHIVPDL)

Each of the above IVP jobs is tailored using the installation dialogs and saved in the XDFHINST CICS library. Some additional jobs must be run prior to running the IVP using ServerPac; the required installation jobs can be found in the CICS IPO1.SAMPLIB.

Refer to *CICS/ESA Installation Guide*, *CICS SystemPac Installation Guide*, and *CICS ServerPac Installation Guide* for details concerning the installation verification process.

Once the batch and online installation verification procedures have been executed successfully, installation-specific customization of the CICS program product should be performed. Detailed information regarding customization can be found in *CICS/ESA System Definition Guide* and *CICS/ESA Customization Guide*.

7.6 DB2 Subsystem Replacement

DB2 and IMS, while separate products, share the same SREL name (P115) and are commonly known as the Data Base System (DBS) feature. IRLM is also included in the P115 SREL. As they share the same SREL, they can also share the global zone, the target zone, and the distribution zone. This section discusses DB2, and also makes reference to IMS where applicable. IMS is discussed in Section 7.7, “IMS Subsystem Replacement” on page 175.

To replace a DB2 subsystem, use either of the following:

- DB2 ServerPac
- DB2 SystemPac

The term “replacement” does not just mean to replace the existing subsystem; it could also mean a new DB2 installation.

7.6.1 DB2 ServerPac

DB2 ServerPac is a package that is free once you have an OS/390 license. You need a license for the DB2 products contained in the ServerPac as well. It is available in a dump-by-dataset format and provides a system that is pre-tested with OS/390 during System Integration Test.

DB2 ServerPac use installation dialogs to install the package. The dialogs simplify the task of installing and customizing the package. The ServerPac installation dialog includes the following package-related sections:

- Installation
- Customization
- Migration
- IVPs
- Customer-specific installation tasks

Using the supplied installation dialogs, you can tailor the install to your existing DB2 environment. This flexibility makes it unnecessary to reorganize the new subsystem after SystemPac installation.

DB2 ServerPac can be used to replace an existing DB2 subsystem or to install a DB2 subsystem for the first time.

7.6.2 DB2 SystemPac

DB2 SystemPac is a fee-based software package consisting of installed products for a ready-to-run DB2 subsystem. It is available in a Full-Volume-Copy or Dump-by-dataset format which is device-independent.

The package has been customized in response to information provided to IBM or by using product supplied defaults. DB2 SystemPac can be used to replace an existing DB2 subsystem or to install a DB2 subsystem for the first time.

DB2 SystemPac uses installation dialogs to install the package. The dialog simplifies the task of installing and customizing the package. The SystemPac installation dialog includes the following package-related sections:

- Installation
- Customization
- Migration
- IVPs
- Customer-specific installation tasks

7.6.3 DB2 Volume and Catalog Structure

In DB2 SystemPac, you have the option of putting the TARGET and DLIB data sets on several volumes. To reduce the number of volumes needed, you can change the VOLSER value in the dialogs to the same volume name at unload time.

Figure 58 shows an example of the DB2 SystemPac catalog structure.

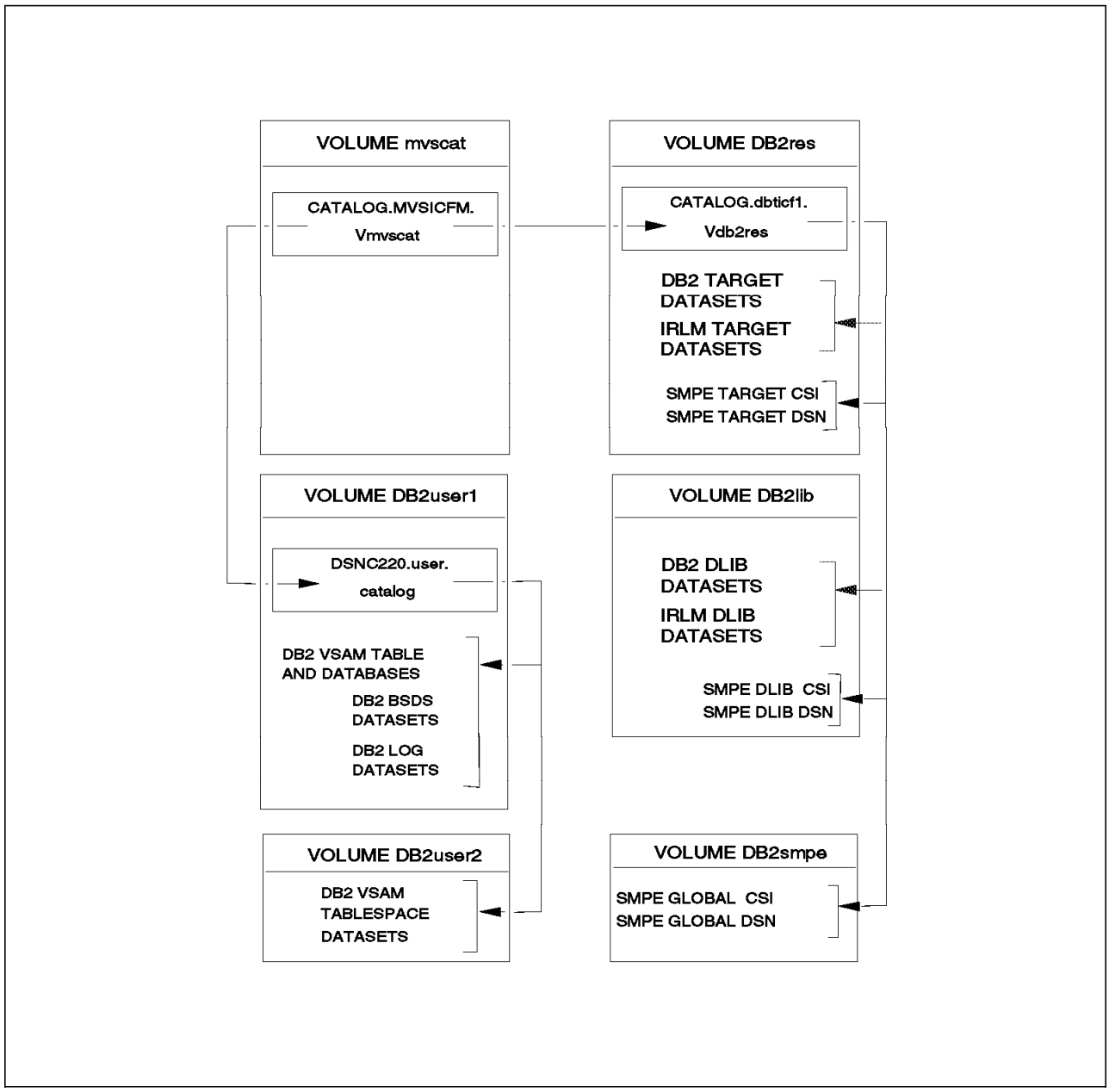


Figure 58. DB2 SystemPac Catalog Structure

During the installation process, you can choose which catalog structure is to be used for the DB2 target and distribution libraries (including IRLM, if any). Using a specific user catalog, as shown in Figure 58, is the recommended way for cataloging the DB2 environment, as this keeps the subsystem's catalog entries together.

There is no need to catalog all the DB2 data sets in the master catalog, because only those libraries that contain modules that are concatenated in the LPALSTxx member of SYS1.PARMLIB have to be cataloged in the master catalog. *IBM DataBase 2 Administration Guide* describes the following information:

- DSNnnn.DSNLINK contains modules that you *must* place in the link list because they are loaded at subsystem initialization during IPL. DSNnnn.DSNLINK has the following characteristics:

  - Contains initialization code
  - Is shareable by multiple subsystems and releases of DB2
  - Is APF-authorized

- DSNnnn.DSNLOAD contains modules that you *can* place in the MVS link list. DSNnnn.DSNLOAD has the following characteristics:
  - Is the main load module repository
  - Is shareable by multiple subsystems at same release level
  - Allows only DB2 to modify code
  - Holds default exits
  - Is APF-authorized
- DSNnnn.DSNEXIT contains modules that you *can* place in the link list. DSNnnn.DSNEXIT has the following characteristics:
  - Holds ZPARMs, DECP, and user-written exits
  - Is modified by user
  - Is APF-authorized

Libraries DSNnnn.DSNLINK, DSNnnn.DSNLOAD, and DSNnnn.DSNEXIT are separate to allow users who are supporting two levels of DB2 to access modules for either level by using STEPLIB and JOBLIB statements.

You may still want to have the DSNnnn.DSNLOAD data set listed once in the LNKLSTxx to remove the need for STEPLIB statements. In this case, you may actually put the DSNnnn.DSNLOAD and DSNnnn.DSNLINK into one library. If you do not place DSNnnn.DSNLOAD in the LNKLSTxx member of SYS1.PARMLIB, you must provide JOBLIB or STEPLIB statements for those procedures and JOBs which use the modules in this data set. The installation and migration jobs provided with DB2 contain the necessary JOBLIB or STEPLIB statements.

The function that solves the problem when you have to catalog several data sets on one system with the same name, is called System-Specific Aliases (SSAs). These aliases are used to locate any data set, system wide, even if it has the same name, without requiring a JOBCAT or STEPCAT. See Section 3.5.2, “System-Specific Aliases” on page 63 for a description of SSAs.

To replace or install the DB2 product set in the maintenance environment, follow the DB2 ServerPac or DB2 SystemPac installation process.

7.6.4 DB2 SMP/E Zone Structure

The DB2 ServerPac and DB2 SystemPac installation processes provide for implementation of DB2 with its own global, target, and distribution zones. This default DB2 ServerPac and DB2 SystemPac installation process defines a separate global zone for each offering that is installed. Figure 59 shows this structure using the DB2 SystemPac or DB2 ServerPac.

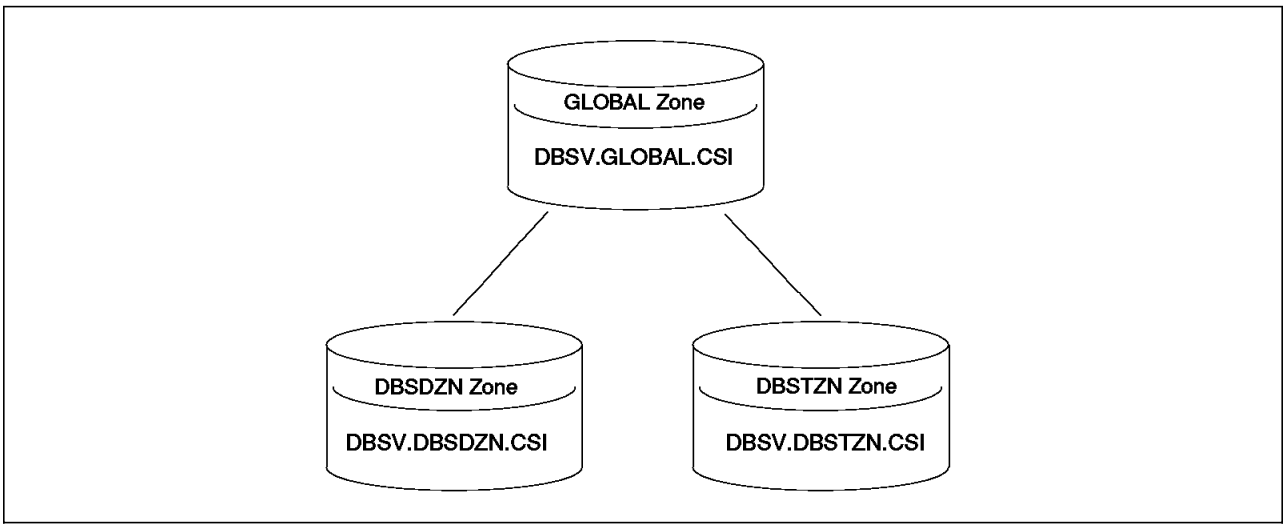


Figure 59. SMP Zone Structure for DB2 SystemPac

Each zone is represented by a single CSI. The data set and zone names are sample names. The names of the zones and of the CSIs can be changed easily by entering the new names in the installation dialog panel.

Using a single global zone approach, as one of the alternatives, requires modifications to the ServerPac or SystemPac implementation process. This is because theses CustomPac installation processes, by default, always define a unique global zone for each subsystem. Whether you use single or separate global zones, each of the subsystems is implemented with its own target and distribution zones (target zone and distribution zone are SREL-dependent).

As discussed earlier, you may modify the ServerPac and SystemPac installation process to comply with the single global zone structure. If you do so, you have to use unique target zone and distribution zone names, making sure they are not defined in the single global zone to describe other target and distribution zones. Refer to Section 7.2.2, “Single Global Zone” on page 155 for further information. Since some jobs could corrupt the existing SMP/E system if not properly tailored, be careful prior to submitting them. It is suggested that you substitute your own tailored jobs for the jobs supplied with DB2 CustomPac, since modifications in supplied jobs are not saved. Refer to *DB2 SystemPac Installation guide* and *ServerPac Installation Guide* regarding this modification.

Figure 60 depicts the single global structure for the DB2 maintenance environment.

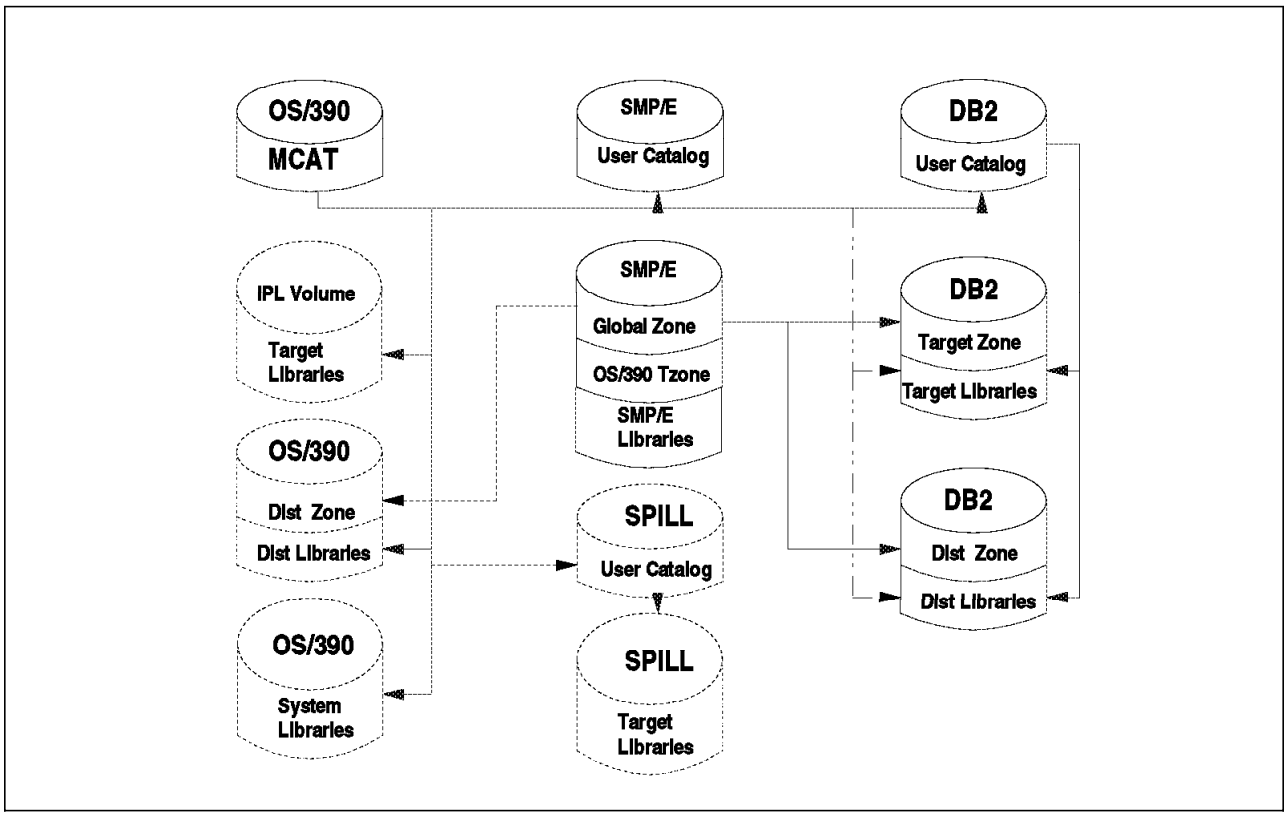


Figure 60. Single Global Zone Structure for the DB2 Maintenance Environment

The DB2 SMP/E related data sets can be cataloged either in the DB2 user catalog or in the SMP/E user catalog. Unique, meaningful qualifiers should be used.

If the default installation procedure is used, the target and distribution data sets will be cataloged in the master catalog. Some modifications will have to be made during installation in order to catalog the target and distribution data sets in the DB2 catalog. Refer to Section 7.6.3, “DB2 Volume and Catalog Structure” on page 166 for further information.

As discussed earlier, DB2 and IMS can share not only the global zone, but also the target zone and the distribution zone. Together they are known as the DataBase System (DBS) product set. The following section describes the possible DBS SMP/E zone structure combinations. The advantages and disadvantages of each structure will also be discussed.

7.6.4.1 DBS Single Zone SMP/E Structure

Figure 61 depicts the logical SMP/E structure shared by IMS and DB2.

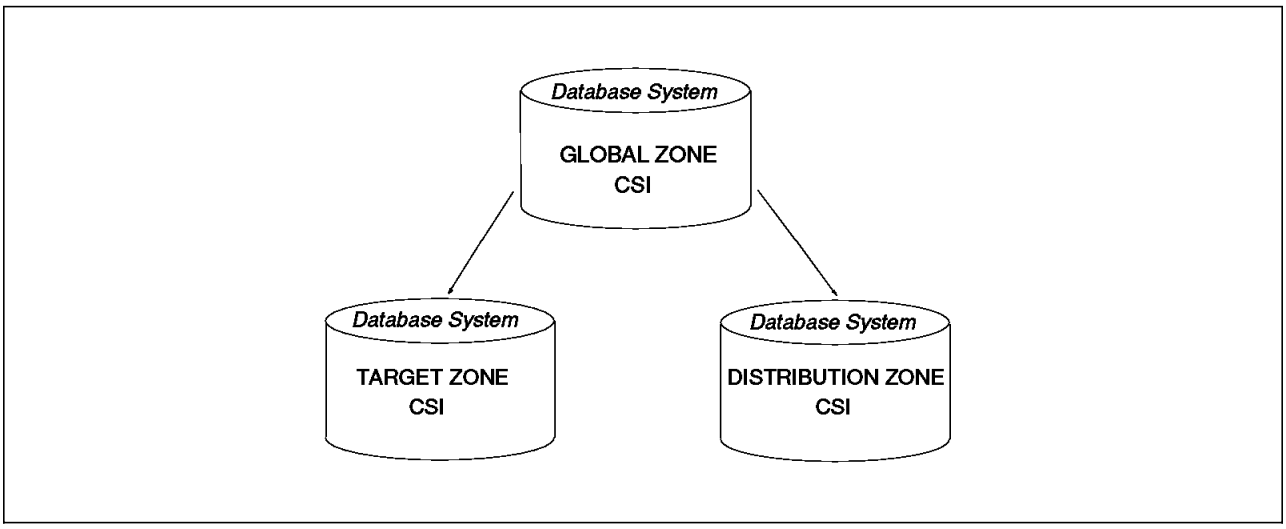


Figure 61. CBIPO DBS (DB2 and IMS) Using the Single Zone SMP/E Structure

This approach is good if you maintain the database area from a single source at your installation. Any dependencies between DB2 and IMS are also easily managed, since they are maintained from the same SMP/E data sets. However, you can run into problems if you wish to totally replace one of the database subsystems without replacing the other. In this instance, you would have to use a CBPDO rather than a SystemPac or ServerPac to replace the upgrading subsystem.

It is recommended that you run separate IRLMs, one for each DB2 system, and one for IMS (refer to *IBM Database 2 Administration Guide*). If you use the same zones for IMS and DB2, you will be limited to using the same DDDEFs for both. The alternative is to edit every maintenance job to point the IRLM maintenance to different data sets. If DB2 and IMS operationally share a single IRLM, a reduction in isolation levels and available lock space may result.

There is a cautionary note if you plan to use IMSGENS. If DB2 and IMS share the same SMP/E data sets, you must ACCEPT or re-APPLY DB2 corrective service to these data sets to allow IMS SYSGENS. Refer to *IBM Database 2 Administration Guide* for details. These problems may occur because the IMSGEN process uses the IMS and DB2 distribution libraries (GENLIBs, LOAD, DBSOURCE, DCSOURCE, DSNALOAD, DSNAMACS, and DSNAHELP). The target zone is re-created by copying the distribution zone, so service on DB2 or any product that was applied but not accepted is not carried to the new target library. This information is therefore lost. To rectify this situation, you have to re-APPLY any DB2 maintenance that in not in ACCEPT status prior to running the IMSGEN. The maintenance would have to have been retained in the SMPPTS.

7.6.4.2 DBS Separate Target and Distribution Zones SMP/E Structure

Figure 62 depicts the logical SMP/E structure shared by DB2 and IMS.

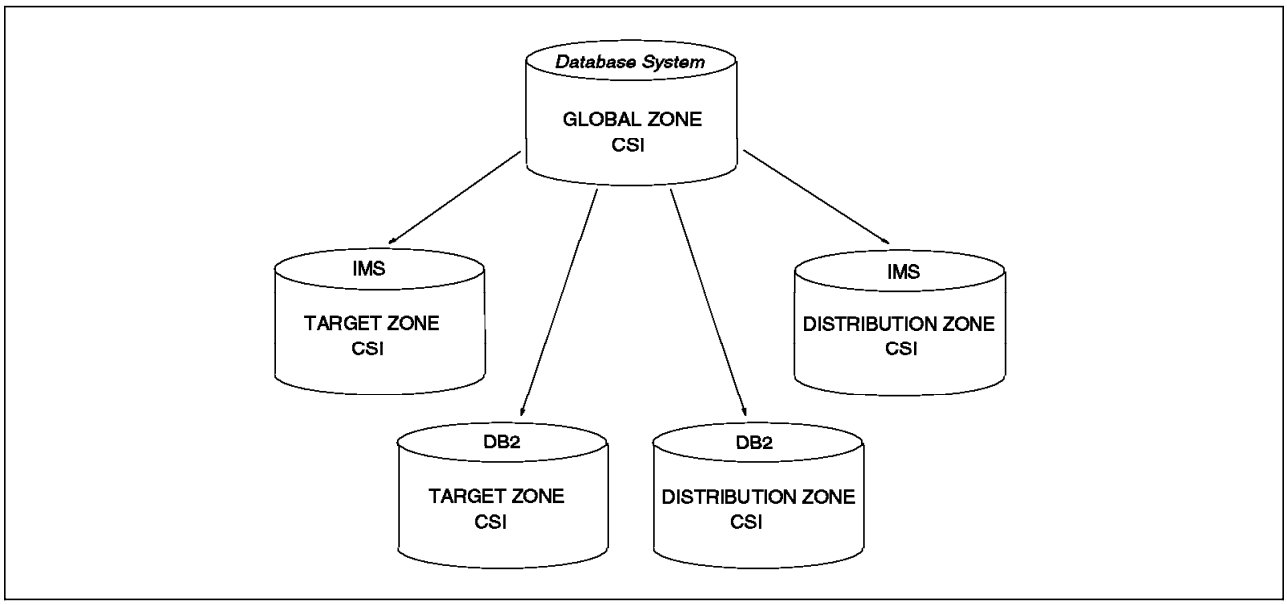


Figure 62. DBS (DB2 and IMS) Using Separate Target and Distribution Zones

In this SMP/E structure, IMS and DB2 are sharing the global zone. Updating one subsystem does not affect the status of the other subsystem. This enables you to use ServicePac, SystemPac, CBPDO, or ServerPac as the mechanism for maintaining the subsystem environment. The discussion in Section 7.6.3, "DB2 Volume and Catalog Structure" on page 166 can be used in conjunction with this SMP/E structure.

Running two occurrences of IRLM is easily managed in this environment. The DDDEFs for the IRLM feature can point to different data sets for IMS and DB2. By keeping the IMS IRLM separate from the DB2 IRLM, you can set the DB2 IRLM parameters independent of the IMS parameters. This allows you to tailor the parameters to the needs of the different subsystems.

7.6.4.3 DBS Completely Separate Zones SMP/E Structure

Figure 63 depicts the logical SMP/E structure used by IMS and DB2. The recommendation is that you maintain separate global zones, and put entries for the target and DLIB zones into a dummy global zone (ZONEINDEX). The advantages of doing this are discussed in Section 7.2.1, "Separate Global Zones" on page 153. Although you may have some extra work initially creating this SMP/E structure, it will be easier to maintain, and offers increased flexibility when installing new or multiple releases of DB2 and IMS.

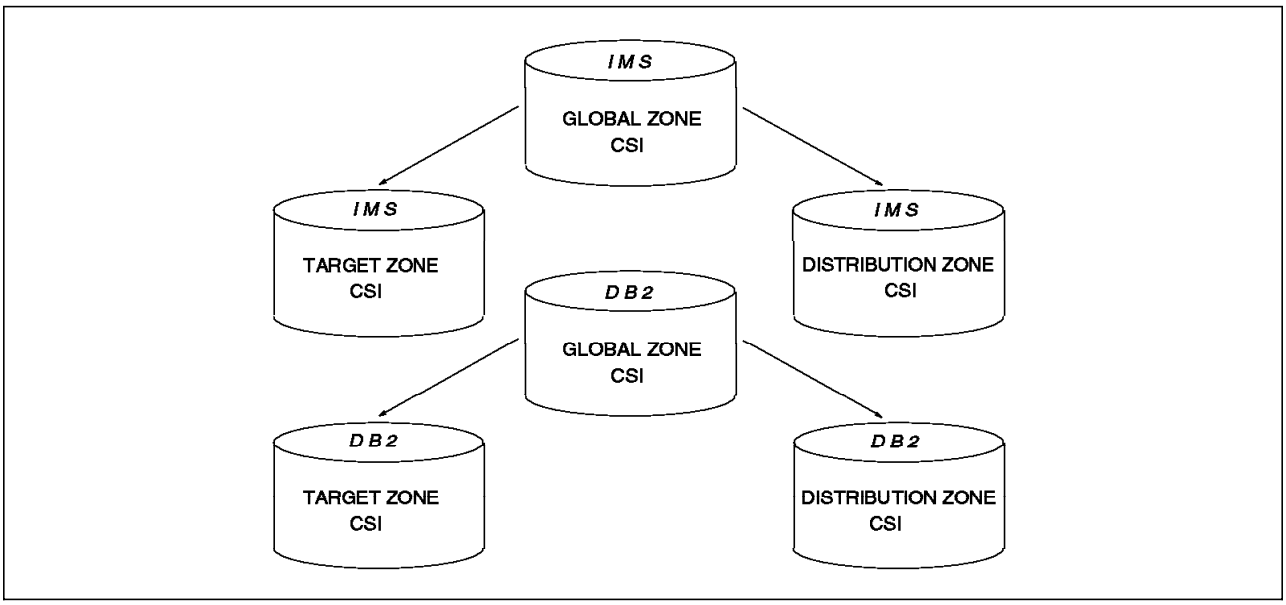


Figure 63. DBS (DB2 and IMS) Using Separate Zones SMP/E Structure

7.6.5 DB2 Installation and Post-Installation Tasks

This section describes the DB2 Installation process in detail. The other program products that come with the package, such as QMF, DB2PM, and so on will be discussed briefly. Refer to the related product installation documents for further details about the installation.

Both DB2 SystemPac and ServerPac is installed using the installation dialogs that are delivered with each offering. The DB2 SystemPac and ServerPac installation dialog specifically provides almost everything you need for a successful installation of DB2. You may choose to use the information provided online rather than the hardcopy supplied in the box. The Installation section, which is made up of the Package Specific Install, Product Specific Install, and Post Installation and Customization is quite easy to follow. Just make sure that you perform the variable tailoring (and job customization, if any) correctly. At this point in the installation you should have:

- DB2 user catalogs and their aliases created
- DB2 SMP/E related data sets (CSIs and all non-VSAM data sets) established
- DB2 distribution libraries allocated and loaded
- System (target) libraries allocated
- The logon procedure, ISPDB2T (copied from the default ISPF PROC and updated)
- DB2 SMP/E procedure (DSNTIJSM) copied to SYS1.PROCLIB
- SYS1.PARMLIB members LNKLSTxx, SCHEDxx, IEFSSNxx, and IEAAPFxx updated
- DSNTINST CLIST created and run
- IVP jobs installed
- VSAM and non-VSAM data sets and databases for DB2 defined and initialized

- DB2 initialization parameter module DSNZPARM generated
- Sample authorization exits built in DSNnnn.DSNEXIT
- SMFPRMxx member of SYS1.PARMLIB updated (for statistical and accounting purposes)
- DB2 connected to TSO
- Subsystem security using external security (RACF) established (optional)
- DB2 connected to IMS (optional)
- Temporary work files and DB2 objects defined and bound
- DB2 product-related information (PGMDIR, PSPINFO and HOLD information) loaded

To make this newly installed DB2 ready to be verified using the IVP jobs, do a system IPL. You have to IPL the MVS system to make all changes in SYS1.PARMLIB effective. Be sure to include the DB2 PARMLIB suffix (ID) in the IPL parameter request. After a successful IPL you can start the DB2 subsystem using the START command. Refer to *DB2 SystemPac Installation Guide* or *DB2 ServerPac Installation Guide* for further details.

7.6.6 DB2 Installation Verification Process

The installation verification procedure for DB2 consists of seven phases: one cleanup phase that drops sample objects and six verification phases. Certain phases of the verification procedure might not apply to the environment in which your DB2 system operates, so you might not perform all phases.

Phase 0 of the installation verification procedure is to be run only if you want to remove all the previous verification processing so that you can begin the verification procedure again. Phases 1-3 test the TSO and batch environments. Phase 4 is for IMS users only, and Phase 5 is for CICS users only. Phase 6 sets up the sample tables for remote-unit-of-work processing.

A brief description of each phase and the jobs provided in the installation dialog is given below. For a complete description of the installation and verification procedures and the sample applications that are provided by DB2, refer to *IBM Database 2 Administration Guide*.

- Phase 0: Delete the Sample Objects

  Phase 0 consists of one job, DSNTEJ0, that frees all plans, drops all objects, and deletes data sets so that Phase 1 can be run again. Even when DSNTEJ0 runs successfully, some of the FREE, DROP, and DELETE commands often fail because the object was not created earlier. You can ignore these errors even though they might generate return codes of 8 or 12.

- Phase 1: Create Load Sample Tables

  Phase 1 consists of two jobs: DSNTEJ1 and DSNTEJP. DSNTEJ1 prepares and invokes program DSNTIAD, which creates objects during the verification procedure. DSNTEJ1P prepares and invokes program DSNTEP2, which lists the contents of the sample tables. DSNTEJ1 must be run before any other sample jobs. DSNTEJ1P can be run only if PL/I is installed on your system. Only step PH01PS02 in job DSNTEJ1P can be run using the SPUFI facility of DB2I.

- Phase 2: Test the Batch Environment

This phase consists of several jobs. These jobs test the program preparation procedures for Assembler(DSNTEJ2A), COBOL(DSNTEJ2C), C(DSNTEJ2D), FORTRAN(DSNTEJ2F), and PL/I(DSNTEJ2P).

- Phase 3: Test SPUFI, Remote Unit of Work, Dynamic SQL, and TSO.

  Phase 3 allows you to test SPUFI, test remote unit of work, test Dynamic SQL statements, run the sample phone application in TSO, and bind packages at the local and remote locations.

  Phase 3 jobs install the ISPF/CAF sample application. This sample consists of an Assembler or COBOL call attachment facility (CAF) interface, a connection manager program, the phone application, and the remote unit of work application. Job DSNTEJ1 prepares the assembler interface and job DSNTEJ3C prepares the COBOL interface. Job DSNTEJ3C and job DSNTEJ3P prepare the COBOL and PL/I versions of the connection manager program. The remote unit of work application is written in COBOL.

- Phase 4: Test the IMS environment

  Jobs DSNTEJ4C and DSNTEJ4P install sample IMS transactions for IMS, COBOL, and PL/I. These jobs perform the following functions:

  – Precompile, compile, and link-edit the IMS online applications
  – Bind the IMS online applications
  – Create the message format service (MFS) panels for the online application
  – Run the required PSBGEN and ACBGEN

- Phase 5: Test the CICS environment

  Phase 5 consists of two jobs, DSNTEJ5C and DSNTEJ5P. Job DSNTEJ5C installs the sample application transactions in COBOL and prepares the organization application. Job DSNTEJ5P installs the transactions in PL/I and prepares the organization, project, and phone applications. Both jobs perform the following functions:

  – Compile and link-edit the CICS online applications
  – Bind the CICS online applications
  – Create the BMS maps for the online applications

- Phase 6: Access Data at a Remote Site

  This is an optional phase. During this phase, data at a remote site will be accessed using Remote Unit of Work. The Remote Unit of Work organization application scenario is executed as part of Phase 6. Before this can be run, job DSNTEJ6 must be run.

The output of the install verification steps is described in the *IBM Database 2 Administration Guide*.

7.7 IMS Subsystem Replacement

IMS and DB2, while separate products, share the same SREL name (P115) and are commonly known as the Data Base System (DBS) feature. As they share the same SREL, they can also share the global zone, the target zone, and the distribution zone. This section will discuss IMS, and will also make references to DB2 where applicable. DB2 is discussed in section 7.6, “DB2 Subsystem Replacement” on page 165.

To replace a IMS subsystem, use one of the following:

- IMS ServerPac
- IMS SystemPac

The term “replacement” does not just mean to replace the existing subsystem; it could also mean a new IMS installation.

7.7.1 IMS ServerPac

IMS ServerPac is a package which is free once you have an OS/390 license. You need a license for the IMS products contained in the ServerPac as well. It is available in a dump-by-dataset format and provides a system that is pre-tested with OS/390 during System Integration Test.

IMS ServerPac use installation dialogs to install the package. The dialogs simplify the task of installing and customizing the package. The ServerPac installation dialog includes the following package-related sections:

- Installation
- Customization
- Migration
- IVPs
- Customer-specific installation tasks

Using the supplied installation dialogs, you can tailor the install to your existing IMS environment. This flexibility makes it unnecessary to reorganize the new subsystem after SystemPac installation.

IMS ServerPac can be used to replace an existing IMS subsystem or to install a IMS subsystem for the first time.

7.7.2 IMS SystemPac

IMS SystemPac is a fee-based software package consisting of installed products for a ready-to-run IMS subsystem. It is available in a Full-Volume-Copy or Dump-by-dataset format which is device-independent.

The package has been customized in response to information provided to IBM or by using product supplied defaults. IMS SystemPac can be used to replace an existing IMS subsystem or to install a IMS subsystem for the first time.

IMS SystemPac uses installation dialogs to install the package. The dialog simplifies the task of installing and customizing the package. The SystemPac installation dialog includes the following package-related sections:

- Installation
- Customization
- Migration
- IVPs
- Customer specific installation tasks

IMS ServerPac and SystemPac use System Specific Aliases (SSAs) in order to define unique names for all data sets, and to ensure that all data sets are cataloged. See section 3.5.2, “System-Specific Aliases” on page 63 for a description of SSAs.

To replace an IMS product set, or install one for the first time in the maintenance environment, the IMS SystemPac or ServerPac installation process is recommended.

The method that you use for subsequent system upgrades depends on how your maintenance environment is set up. As discussed in section 7.6, “DB2 Subsystem Replacement” on page 165, using ServerPac or SystemPac you have the option of putting the IMS and DB2 products in one SMP/E environment (DBS). Therefore, you must decide whether to order separate ServerPacs for the IMS install and the DB2 install, or to order the two on the same ServerPac.

The issues that have to be considered are discussed in detail in Section 7.6.4, “DB2 SMP/E Zone Structure” on page 168. Refer to that section for more detailed information.

7.7.3 IMS Volume and Catalog Structure

Figure 64 shows an example of the IMS catalog structure.

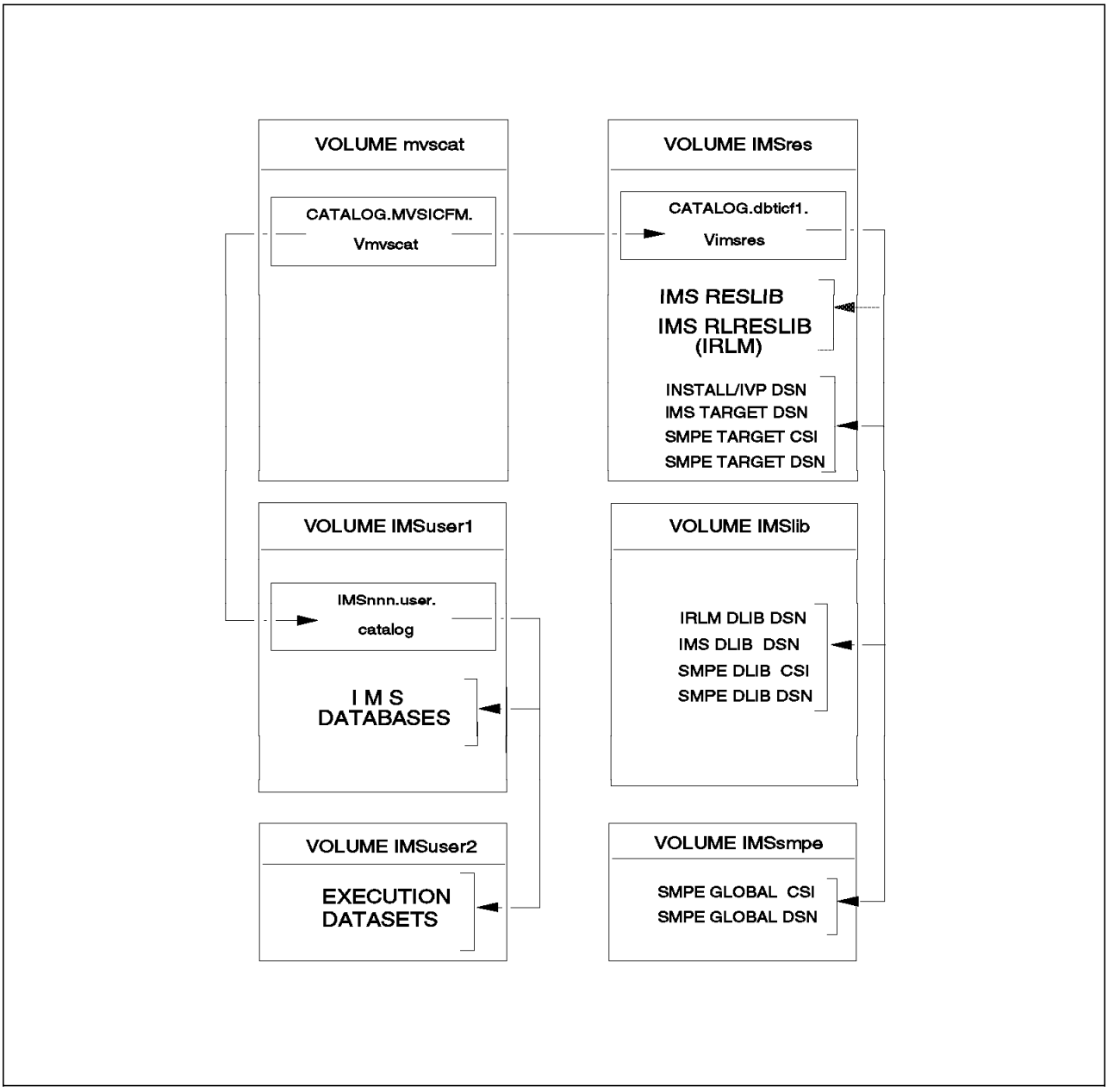


Figure 64. IMS Catalog Structure

Some of the libraries are concatenated with SYS1.LINKLIB such as IMS.RESLIB, and IMS.RLRESLIB.

The concept of System-Specific Aliases (SSAs) has introduced a function to handle the situation when you have to catalog several data sets with the same name on the same system. These aliases are used to locate any data set, systemwide, even if it has the same name, without requiring a JOBCAT or a STEPCAT. See section 3.5.2, “System-Specific Aliases” on page 63 for a description of SSAs.

To replace or install the IMS product set in the maintenance environment, follow the IMS installation process.

7.7.4 IMS SMP/E Zone Structure

The IMS ServerPac and SystemPac installation process provides for the implementation of IMS with its own global, target, and distribution zones. Figure 65 shows this structure.

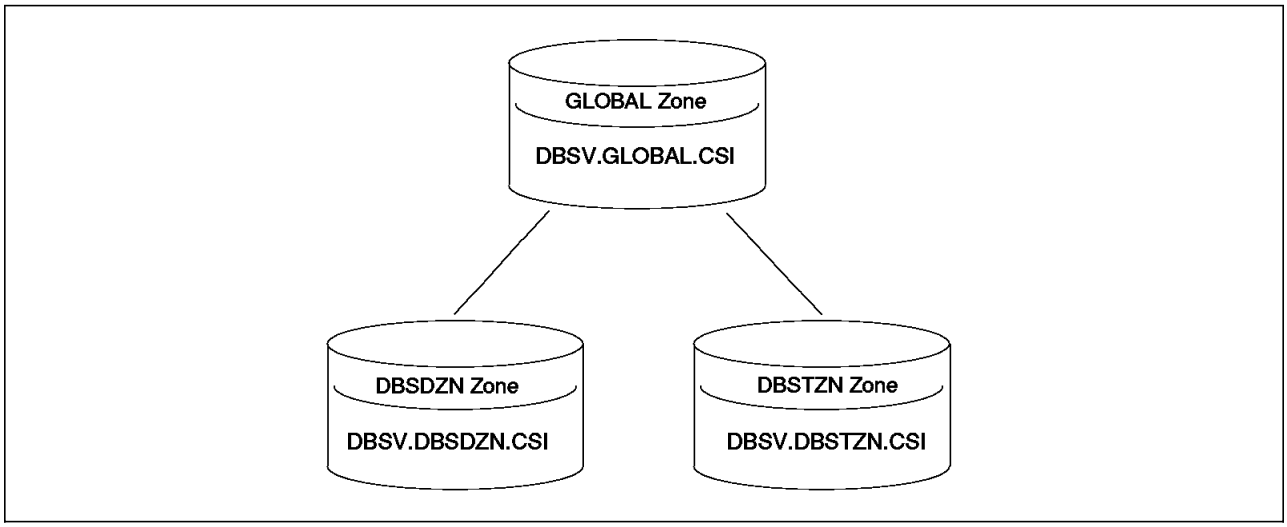


Figure 65. SMP Zone Structure for IMS ServerPac or SystemPac

Using a single global zone approach in which you have one global zone for all subsystems (DB2, IMS and CICS share the same global zone) as one of the alternatives requires modifications to the ServerPac and SystemPac implementation process. This is because the IMS ServerPac or SystemPac installation process, by default, always defines a unique global zone for each subsystem. Whether you use single or separate global zones, each of the subsystems will be implemented with its own target and distribution zones (target zone and distribution zone are SREL-dependent).

As discussed earlier, you may modify the SystemPac and ServerPac installation process to comply with the single global zone structure. If you do so, you have to use unique target zone and distribution zone names. Refer to section 7.2.2, “Single Global Zone” on page 155 for further information. Since some jobs could corrupt the existing SMP/E system if not properly tailored, be careful prior to submitting them. Refer to *ServerPac Installation Guide* or *SystemPac Installation Guide* regarding this modification.

Figure 66 depicts the single global structure for the IMS maintenance environment.

The IMS SMP/E related data sets can be cataloged either in the IMS user catalog or in the SMP/E user catalog. Unique, meaningful qualifiers should be used.

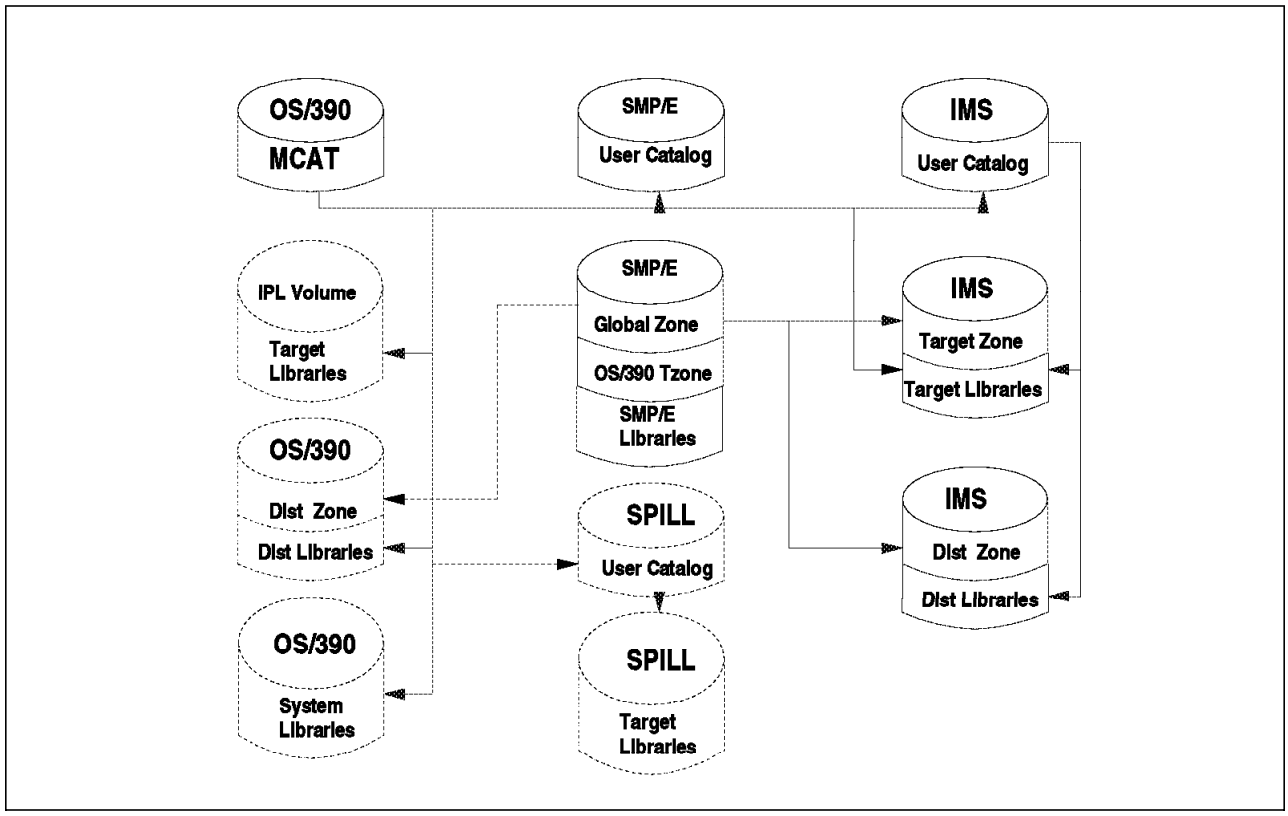


Figure 66. Single Global Zone Structure for the IMS Maintenance Environment

7.7.5 Replacing IMS in the Maintenance Environment

The *OS/390 ServerPac IMS Installation Guide* contains the detailed job descriptions and should be used in conjunction with this section.

When the installation is completed, the DBS product set is included as follows in the maintenance environment:

- IMS user catalog and its aliases created
- IMS SMP/E related data sets (CSIs and all non-VSAM data sets) established
- IMS distribution libraries allocated and loaded
- System (target) libraries allocated
- INSTALIB and INSTATBL data sets are allocated
- INSTALL/IVP dialog is loaded

At this stage, some of the target IMS data sets are still empty. They will be filled during the INSTALL/IVP stage.

What is left to make this new installed IMS ready to be verified using the IVP jobs is a system IPL. You have to IPL the OS/390 system to make all changes effective in this installation process, specifying CLPA or MLPA=xx, or both.

Remember to authorize IMS.RESLIB by adding it to the APF authorization list. For details, refer to *OS/390 Initialization and Tuning Reference*, SC28-1752.

7.7.6 IMS INSTALL/IVP Dialog

The rest of the IMS install is completed by using the IMS INSTALL/IVP dialogs. The INSTALL/IVP dialog is driven from a set of ISPF tables, which contain information about the variables, jobs, tasks, and sequence of presentation you will have to perform for your installation.

Refer to the *IMS/ESA Installation Volume 1 Installation and Verification*, SC26-8023 for details on how to initiate the INSTALL/IVP dialog. It gives a step-by-step approach on how to use the panels.

The dialog is broken down into different and distinct phases. These phases must be run in sequence. In order for a sequence to run, the previous phases must be flagged as being complete. The following session covers the main phases related to the IMS installation, giving a brief description of each phase and the jobs provided in the installation dialog:

1. Variable Gathering Phase

   This phase allows you to specify the environment in which you are completing the install of IMS. It defines JOB statements, DASD allocations, high-level qualifiers for IMS data sets, SMP/E environment parameters, and allocations for some IMS data sets. Ensure that you run through each variable in this phase and confirm that it is set according to your environment. This will save you a lot of editing time when you start to submit the jobs. The final panel for the phase confirms whether you have completed this phase.

   Each time a variable is updated, you must reset the flag for this stage, indicating that the stage is complete. You then rerun the subsequent phases. Once the stage is flagged as complete, you can proceed to the next stage.

2. File Tailoring Phase

   This phase uses the variables from the variable gathering phase, plus skeletons from the DFSSLIB data set, to prepare the materials that will be used during execution. As members are completed, they are stored in the INSTALIB data set. File tailoring replaces existing members in INSTALIB. If file tailoring is rerun, usermods made directly to INSTALIB members will be lost.

   The first time that you enter this phase, ensure that you perform the YES option on the File Tailor All Request panel. The variables that you entered in the variable gathering phase update the members in INSTALIB. Selecting YES on this panel takes a long time to run. The panel is updated frequently to reflect the table items that are being processed.

   Rather than editing each job from the file tailoring phase, select the YES option, and proceed to the execution phase. To do this, you have to indicate that the file tailoring phase has been completed.

3. Execution Phase

   These jobs were prepared using the variables that you entered in the variable phase. In the file tailoring phase, these jobs were updated as members residing in INSTALIB. The panels for the execution phase provide you with a step-by-step approach for completing the IMS install and verifying that the installation is complete. Unlike the previous two stages, this stage does not provide an automatic update. You must ensure that each job runs successfully before proceeding to the next item. Beside each member is an

indication of what you have to initiate (this is called a “title”). Titles are divided into:

- NOTE. An informational message, detailing what the jobs that you are about to run are designed to do.
- JOB. Members that have to be submitted in order to perform the updates.
- TASK. Activities that you must perform that are not supported by the INSTALL/IVP jobs.
- XMPL. Jobs that cannot be submitted through the INSTALL/IVP panels. In many instances, XMPLs are provided to help you in performing the required tasks.

  If you wanted the function provided by this XMPL, you have to run it outside the panels. This could be accomplished by running the job directly from a copy of the member in INSTALIB. Any updates that you make using this method should be documented.
- CLIST. For CLISTs that have been provided.

The jobs that you run will depend on the features of IMS you are installing. However, the following steps give you an indication of the kinds of jobs that may have to be run:

- Installation/IVP preparation steps.

  These steps includes the dialog set-up and offline dump formatter.
- DLIB build steps.

  Many of the jobs that would have to be run at this stage were run during the ServerPac or SystemPac install for IMS.
- System definition steps (IMS generation).

  This is where the IMS generation jobs are run. The IVP process also starts with these steps. Once these steps have completed, the installation of IMS is also complete and the following tasks have been accomplished:

  – IMS executable modules have been placed into IMS.RESLIB.
  – SMP/E has been notified of the content of the IMS system through the SMP/E JCLIN process.
  – IMS cataloged procedures have been placed into IMS.PROCLIB.
  – At this time, IMS service can be installed using the normal RECEIVE/APPLY/ACCEPT operation, and preparation of the IMS application may begin.
- IMS to OS/390 and VTAM interface step.

  Some tasks have to be performed in order to establish the interfaces between IMS and OS/390 and between IMS and VTAM. In this section, ensure that you tailor all the jobs to complement your own environment.

When all these steps are completed, the installation of IMS is also completed. Some follow-on steps must be executed depending on the components of IMS that you are installing. The panels will detail the jobs that pertain to the environment that you have set up. These steps are documented in *IMS/ESA Installation Volume 1 Installation and Verification*, SC28-1752.

The different job steps are initiated based on the options you selected on the Installation Environment Options panel.

7.8 NCP Subsystem Replacement

The NCP installation process is quite simple and straightforward. The NCP generated module does not run on an MVS system; it runs in a Communication Control Unit, such as a D/T3745 and therefore NCP and MVS operations are virtually separate.

To replace a NCP subsystem, use one of the following:

- NCP ServerPac
- NCP SystemPac

7.8.1 NCP ServerPac

NCP ServerPac is a package that is free once you have an OS/390 license. You need a license for the NCP product contained in the ServerPac as well. It is available in a dump-by-dataset format and provides a system that is pre-tested with OS/390 during System Integration Test.

NCP ServerPac use installation dialogs to install the package. The dialogs simplify the task of installing and customizing the package. The ServerPac installation dialog includes the following package-related sections:

- Installation
- Customization
- Migration
- IVPs
- Customer-specific installation tasks

Using the supplied installation dialogs, you can tailor the install to your existing NCP environment. This flexibility makes it unnecessary to reorganize the new subsystem after SystemPac installation.

NCP ServerPac can be used to replace an existing NCP subsystem or to install a NCP subsystem for the first time.

7.8.2 NCP SystemPac

NCP SystemPac is a fee-based software package consisting of installed products for a ready-to-run NCP subsystem. It is available in a Full-Volume-Copy or Dump-by-dataset format which is device-independent.

The package has been customized in response to information provided to IBM or by using product supplied defaults. NCP SystemPac can be used to replace an existing NCP subsystem or to install a NCP subsystem for the first time.

NCP SystemPac uses installation dialogs to install the package. The dialog simplifies the task of installing and customizing the package. The SystemPac installation dialog includes the following package-related sections:

- Installation
- Customization
- Migration
- IVPs
- Customer-specific installation tasks

7.8.3 NCP Volume and Catalog Structure

In the NCP ServerPac or SystemPac, you have the option of putting the TARGET and DLIB data sets on several volumes. To reduce the number of volumes needed, you can change the VOLSER value variable in the dialogs to use the same volume name at unload time.

Figure 67 shows an example of the NCP SystemPac catalog structure.

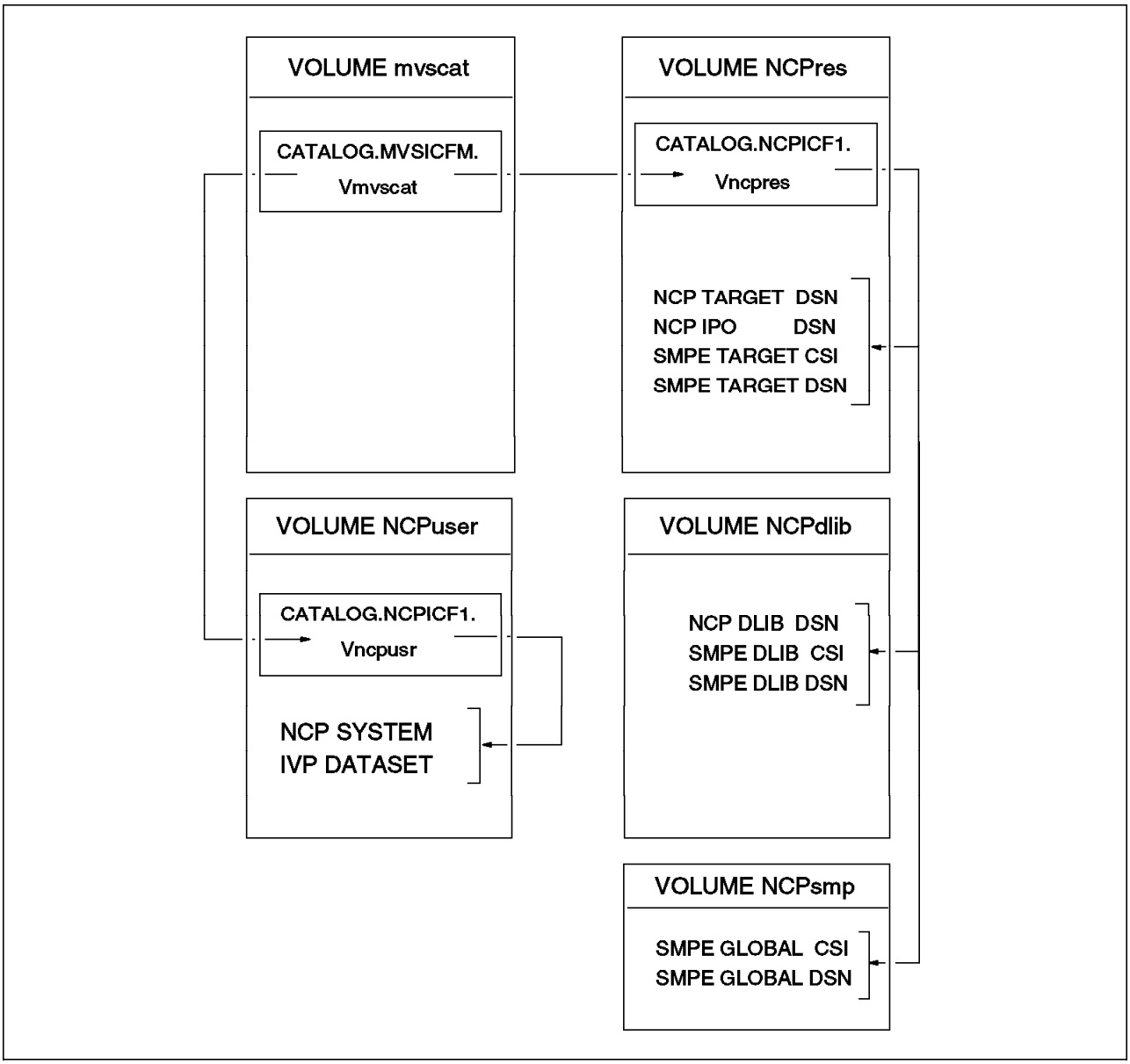


Figure 67. NCP SystemPac Catalog Structure

None of the NCP modules are loaded during the system IPL; therefore, all NCP libraries are cataloged in their own user catalog.

A function to solve the situation when you have to catalog several data sets with the same name on the same system is the concept of System-Specific Aliases (SSAs). These aliases are used to locate any data set, systemwide, even if it has the same name, without it requiring a JOBCAT or a STEPCAT. This is discussed in more detail in section 3.5.2, “System-Specific Aliases” on page 63.

To replace or install the NCP product set in the maintenance environment, follow the NCP SystemPac or NCP ServerPac installation process.

7.8.4 NCP SMP/E Zone Structure

The NCP SystemPac and NCP ServerPac installation processes provide for the implementation of NCP with its own global, target, and distribution zones. The previously discussed CustomPac installation processes will define a separate global zone for each offering that is installed by default. Figure 68 shows this structure using NCP SystemPac.

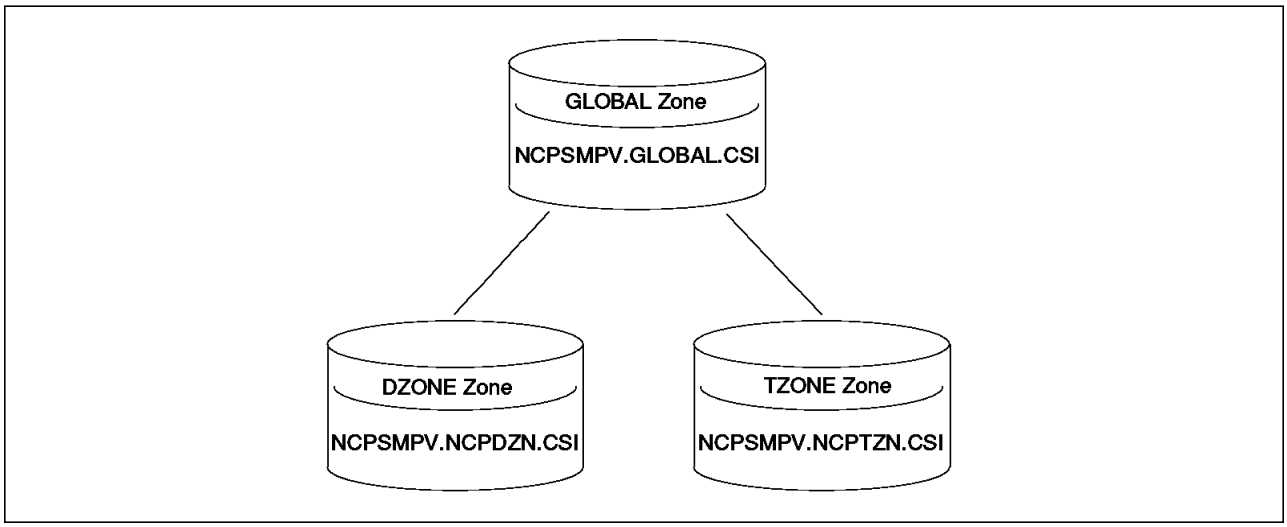


Figure 68. SMP Zone Structure for NCP SystemPac

Each zone is represented by a single CSI. The data set and zone names are sample names. The names of the zones and of the CSIs can be changed easily by entering the new names for the variables in the dialog panel.

Using a single global zone approach as one of the alternatives requires modifications to the SystemPac or ServerPac implementation process because either process, by default, always defines a unique global zone for each subsystem. Whether you use single or separate global zones, each of the subsystems will be implemented with its own target and distribution zones (target zone and distribution zone are SREL-dependent).

As discussed earlier, you may modify the SystemPac or ServerPac installation process to comply with the single global zone structure. If you do so, you have to use unique target zone and distribution zone names, making sure they are not defined in the single global zone to describe other target and distribution zones. Refer to section 7.2.2, “Single Global Zone” on page 155 for further information. Since some jobs could corrupt the existing SMP/E system if not properly tailored, be careful prior to submitting them. It is suggested that you substitute your own tailored jobs for the NCP SystemPac supplied jobs, since modifications in supplied jobs are not saved. Refer to *NCP SystemPac Installation Guide* and *ServerPac Installation Guide* regarding this modification.

Figure 69 depicts the single global structure for the NCP maintenance environment.

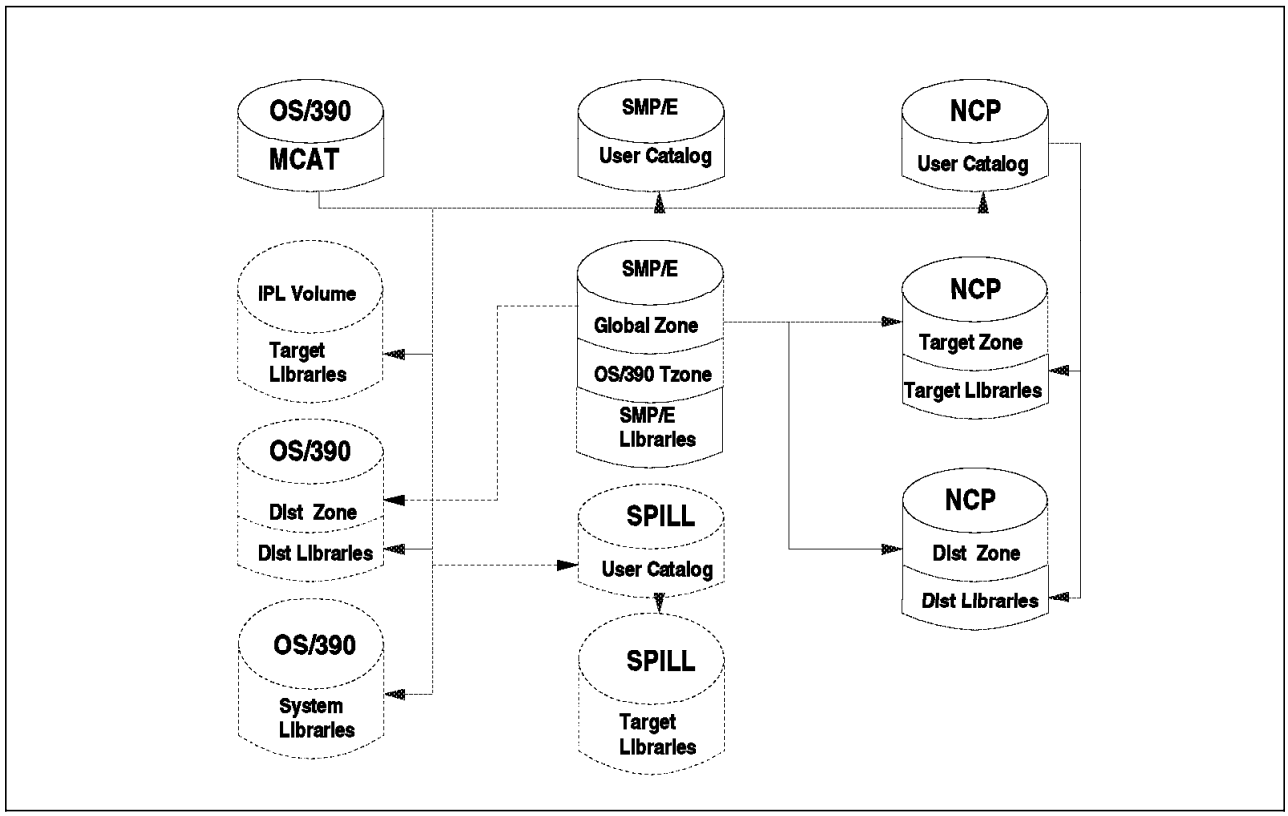


Figure 69. Single Global Zone Structure for NCP Maintenance Environment

The NCP SMP/E related data sets can be cataloged either in the NCP user catalog or in the SMP/E user catalog. Unique, meaningful qualifiers should be used.

7.8.5 NCP Installation and Post-Installation Tasks

The NCP SystemPac and ServerPac is installed using the installation dialogs that are delivered with each offering. The NCP SystemPac and ServerPac installation dialogs provide almost everything you need to successfully install the NCP product. You may choose to use the information provided online rather than using the hardcopy supplied in the box. The Installation section, which is made up of the Package Specific Install, Product Specific Install, and Post-Installation and Customization is quite easy to follow. Just make sure that you perform the variable tailoring (and job customization, if any) correctly. At this point in the installation you should have:

- NCP user catalogs and their aliases created
- NCP SMP/E related data sets (CSIs and all non-VSAM data sets) established
- NCP distribution libraries allocated and loaded
- System (target) libraries allocated
- NCP data sets required to execute the IVP allocated
- External security (RACF) in NCP start parameter updated
- NCP product-related information (PGMDIR, PSPINFO and HOLD information) loaded

NCP installation does need an APF authorized library. Refer to the *NCP SystemPac Installation Guide* or *NCP ServerPac Installation Guide* for detail information.

7.8.6 NCP Installation Verification Process

If you wish to test the NCP feature, a job is provided for some levels of NCP to do an NCP load module generation (for NCP 7.3, no jobs are provided, just the documentation required to create the jobs). This sample source is supplied with the NCP IVP job. There are two kinds of IVP jobs provided:

- HNC5402B for ACF/NCP
- HEP1900B for Emulation Program (EP)

The output of this test NCP generation will probably not be useable in your installation, but the sample definition statements may be helpful when you are customizing NCP. This job may also be used for NCP generation after you have created your customized NCP source.

Refer to *NCP, SSP, and EP Generation and Loading Guide*, SC31-6221 for further information.

7.9 Subsystem Cloning

Cloning a system is the process of making a duplicate copy of the base system. For the following discussions, the purpose of cloning a system will be to provide one or several target production operational levels, from either an upgraded maintenance level or a production maintenance level.

A subsystem clone has its data sets cataloged in separate user catalogs and resides on different DASD volumes. They may or may not exist on the same system image. First, decide whether SMP/E is required in the production operational environment. If all the subsystems are cloned from a production maintenance level or an upgraded maintenance level, then there is no need to clone the SMP/E data sets. Maintenance should always be performed on the production maintenance level and copied to the production operational levels. If you choose to use SMP/E to clone the subsystems, you will have to create a new set of target libraries, and a new target zone pointed to by the single master global CSI. Such a structure is illustrated in Figure 70, using as an example the DB2 and IMS subsystems.

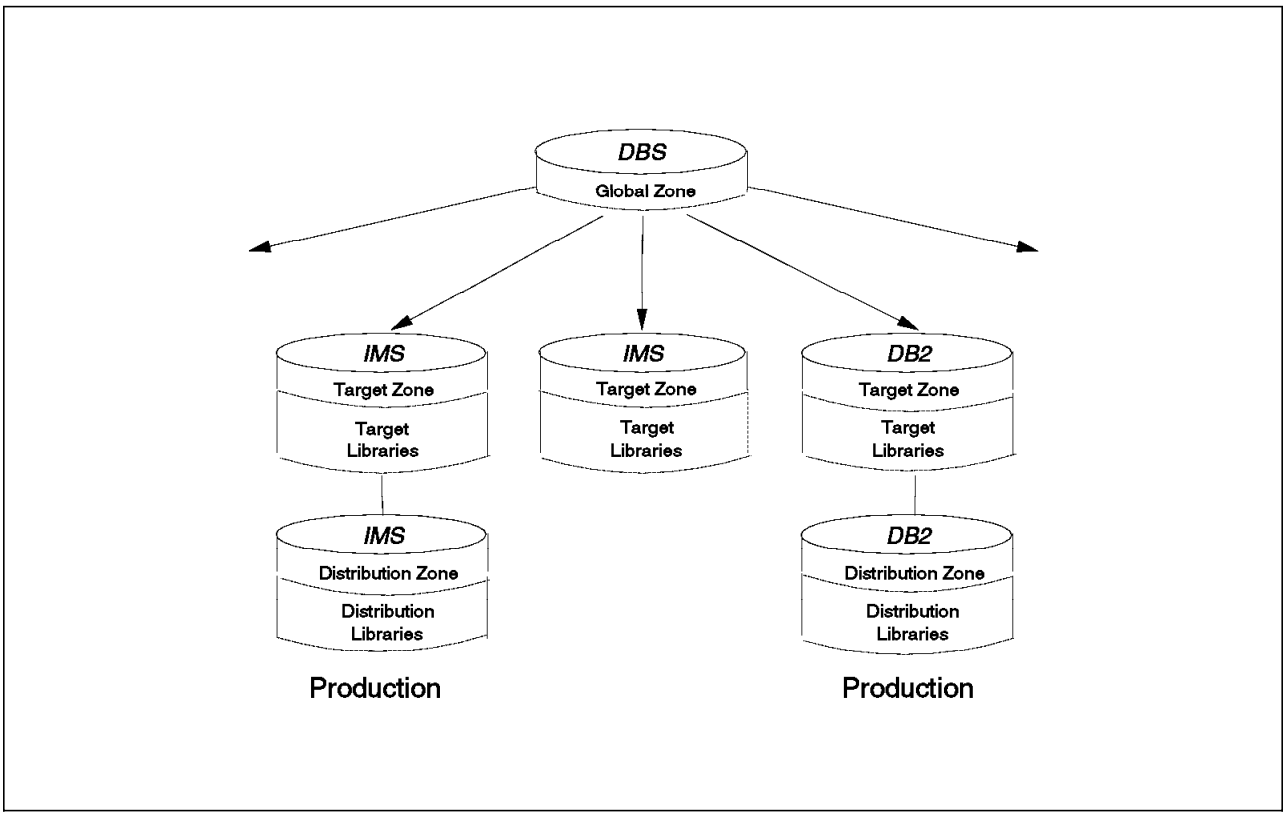


Figure 70. SMP/E Structure for Subsystem Cloning

The steps to create this environment are listed as follows.

1. Decide what the subsystem naming conventions will be.
2. Allocate a user catalog for each subsystem environment that you intend to clone.
3. Perform new RACF definitions; for example, new data sets profiles, new started task names, or new resource classes.
4. Define aliases in the master catalog related to the new user catalog.
5. Allocate new target data sets and libraries.
6. Copy all the target data sets and libraries to the new production operational level.
7. Update VTAM with the new production operational level APPLIDs.
8. Update PARMLIB definitions.
9. Update the OS/390 libraries if required (SVCs, for example).

If you intend to use SMP/E, you will also have to do the following:

1. Allocate and copy your SMP/E data sets (STS, MTS, SCDS, and LOG).
2. Define a new target CSI.
3. Prime this CSI using GIMZPOOL.
4. In the global zone, add an entry for the new target zone in the ZONEINDEX.
5. Using ZONECOPY, copy the source target CSI to the new target CSI.

6. Using ZONEEDIT, change all the DDDEF entries in the target zone to point to the new data set names.

This cloning process results in two copies of every data set, with each copy having its own unique name. For example, DB2.DSNLOAD resides on the production operational level, and an identical data set, DB2.MAINT.DSNLOAD resides on the production maintenance level. Further maintenance will be applied to the production maintenance level.

7.9.1 Cloning DB2

Once you have cloned the DB2 target libraries, you must revisit the customization and IVP steps for the DB2 install. DB2 uses a CLIST called DSNTINST, which can run under native TSO or under TSO/ISPF. This is discussed in more detail in section 7.6.6, “DB2 Installation Verification Process” on page 174. The verification process performs post SMP/E installation and customization activities such as:

- Installs DB2
- Connects DB2 to TSO, IMS/ESA, or CICS/ESA
- Prepares DB2 for use
- Verifies the installation process

Ensure that when you execute the DSNCLIST member, you are executing the cloned environment. In this case, we have to change the PREFIX parameter value in the INSTALL/IVP dialog panel to the new DB2 name. Failure to do so could mean that you overwrite the existing values. Refer to *DB2 for MVS/ESA Installation Guide* , SC26-3456 for a full description of the DB2 INSTALL/IVP dialogs, and to *DB2 for MVS/ESA Administration Guide*, SC26-3265 for other DB2-related installation and customization topics.

7.9.2 Cloning IMS/ESA

Before attempting to clone an IMS/ESA system, make sure that all maintenance has been SMP/E accepted. If an IMS system definition is executed when maintenance is in APPLY status, the maintenance might be regressed.

The IMS/ESA installation requires that the DLIB libraries and some target libraries are cloned. This is because the IMS/ESA installation process builds the majority of the target libraries from the DLIB libraries, using the IMS System Definition Process (IMSGEN). This process is detailed in *IMS/ESA Installation Volume 2 System Definition and Tailoring*, SC26-8024. The steps required to clone IMS/ESA are a combination of the steps to clone the target libraries, and certain steps performed by the IMS/ESA INSTALL/IVP dialogs. These steps are discussed in detail in Section 7.7.6, “IMS INSTALL/IVP Dialog” on page 180.

The following target libraries will require copying prior to invoking the INSTALL/IVP dialogs:

- INSTALIB
- INSTATBL
- DFSCLST
- DFSEXEC
- DFSMLIB

- DFSPLIB
- DFSSLIB
- DFSTLIB
- DFSRTRM

The INSTALL/IVP process consists of jobs, tasks, and supporting materials necessary to install IMS/ESA, and then verify the installation. It is divided into two major functions, INSTALL and IVP. You can use the INSTALL function to:

- Allocate the SMP/E data sets (if required)
- Allocate the Distribution Libraries (DLIB)s
- Allocate the target libraries
- Initialize the SMP/E data sets (if required)
- Copy the DLIBs
- Process IMS/ESA pre-generation service

You can use the IVP function to:

- Perform the IMSGEN
- Establish an interface between OS/390 and VTAM
- Prepare and execute the applications and systems verification routines

Refer to *IMS/ESA Installation Volume 1 Installation and Verification*, SC26-8023 for a full description of the IMS/ESA INSTALL/IVP dialogs; to *IMS/ESA Customization Guide System*, SC26-8020: to *IMS/ESA System Administration Guide: DB*, SC26-8012; to *IMS/ESA System Administration Guide: TM*, SC26-8014; to *IMS/ESA System Administration Guide: System*, SC26-8013 for other IMS/ESA-related installation and customization topics.

7.9.3 Cloning CICS/ESA

There are no specific additional tasks required for cloning CICS/ESA subsystems libraries. Additional customization that may be required is documented in *CICS/ESA System Definition Guide*, SC33-1164 and in *CICS/ESA Customization Guide*, SC33-1165.

7.9.4 Cloning NCP

There are no specific additional tasks required for cloning NCP subsystems libraries. Additional customization that may be required is documented in *NCP, SSP, EP Generation and Loading Guide*, SC31-6221.

Appendix A. Electronic Software Support Offerings

This appendix describes electronic facilities provided by IBM to aid you in managing and maintaining your software environment. Confirm with your marketing representative whether any of the following offerings are available in your country. The terms and conditions under which these facilities are offered vary among countries.

A.1 IBMLink and DIALIBM

IBMlink is an electronic interface between you and IBM. It provides access to open (available to all customers), discretionary (available to selected customers), and fee applications.

The applications that exist online today include:

- Product announcement information
- Ordering information
- Servicelink (see below)
- Mailing facilities for electronic communication between you and for instance, your IBM Marketing Team
- Education class schedules
- Publications ordering

This interface is available in some parts of the world as DIALIBM. Functionally there is no difference between DIALIBM and IBMLink.

A.2 ServiceLink

ServiceLink is an application that runs under IBMLink or DIALIBM, allowing access to IBM service information online.

ServiceLink has several components that are described below. It must be stressed that not all of the listed functions are available in all countries and the usage of some of these is controlled.

ServiceLink allows you to electronically do the following without calling the IBM Support Center:

- Search for service and support information
- View product installation information and maintenance information
- Order maintenance
- Electronically report problems
- Monitor the status of open Authorized Program Analysis Reports (APARs), Program Temporary Fixes (PTFs), and service orders

The applications available through Servicelink are:

- Service Information Search (SIS)

  Service Information Search (SIS) lets you search for and retrieve service and support information to help you resolve software problems and questions.

SIS lets you use both free-form and assisted search arguments, and lets you easily refine your search argument to locate the most applicable information. You can select which libraries you would like to search and this may include non-defect libraries (including information on problems with ISV products).

You can use search arguments to look at APAR and PTF information, or to search through a number of technical databases. These databases contain technical hints and tips from user experiences and answers to users' technical questions.

- Preventive Service Planning

  Preventive Service Planning (PSP) lets you see software product installation information. The information is organized into upgrades and subsets that the IBM Support Center normally mails to you as printouts.

- Service Request and Delivery (SRD)

  Service Request and Delivery (SRD) lets you order corrective service for OS/390 products yourself, without calling the IBM Support Center. It also lets you check on the status of your orders.

  The corrective service consists of Program Temporary Fixes (PTFs). You can order an individual fix, several fixes and their requisites in the form of a PTF package, or a fix and only the requisites that apply to the service level of your system, in the form of a PTF package. In some countries it is possible to order a service-only PDO within SRD.

  When a defect is identified, and a fix is available, you may electronically order that specific PTF. You should check with your marketing representative for the electronic options available in your country. Additional software or hardware requirements may need to be met for this.

- Electronic Technical Response (ETR)

  Electronic Technical Response (ETR) lets you electronically report a problem to IBM and to receive responses electronically. You can also request that the IBM Support Center (ISC) call you with their response. Your problem report generates an ETR record. This record is updated by IBM to provide a solution, an answer, or to request more information from you. You are notified that an update to the record has taken place. ETR may also be used for raising non-defect questions (according to the terms and conditions in your country).

- APAR Status Tracking (AST)

  APAR Status Tracking (AST) lets you select Authorized Program Analysis Reports (APARs) or Program Temporary Fixes (PTFs) to track so that you are notified automatically when IBM changes them. APARs that are created because of one of your Electronic Technical Response (ETR) records are automatically added to your tracking list.

  AST lists the APARs and PTFs that you are tracking, and the status of each. The record will show whether the APAR or PTF is:

  Open
  Closed
  Reopened
  In transit (where APAR information is being sent to IBM)
  Fix-tested (where the APAR resolution is being tested)

  You will be notified when the status of the APAR or PTF changes.

- View Program Listings (VPL)

  VPL enables you to view Program Temporary Fix (PTF) listings that were formerly distributed on microfiche. VPL allows viewing of IBM PTF source code listings of their respective products.

  Note: You must be a licensed user to use this feature. This feature requires that you enter the customer number that owns the product license. In some countries this application is available outside Servicelink.

- Product Cross Reference (PCR)

  PCR provides you with a command that displays information relative to software product name, component ID, and FMID cross-reference information.

- Electronic Service Call (ECS)

  ECS allows you to report suspected hardware problems electronically, track the status of the problem, and communicate updated information directly to your CE′s portable terminal (PT).

- Automatic Software Alert Process (ASAP)

  Automatic Software Alert Process (ASAP), a restricted application, lets you receive automatic notifications when critical information about your MVS product changes. The notifications include information about high-impact and pervasive (HIPER) APARs, PTFs in error, zaps in error, and the corresponding fixes. You list the products that you want to be notified about in your ASAP profile.

  ASAP lists the products that you have requested to be notified about, and the corresponding APARs and PTFs. This list is called your “APAR notification list.”

- Remote Viewing Support Capability (RSV-XCEL)

  Remote screen viewing provides you with the capability to control interactive, online sessions that enable multiple users to view the data being presented on the controlling user′s screen.

  RSV-XCEL allows the IBM Support Center and yourself to simultaneously view traces, dumps, record structures, and so on. The appropriate files can then be transferred from you to IBM, or from IBM to you. It also allows you to share data with other personnel at your site, whether they are at the same location or a remote location.

- Remote System Viewing In A PC Environment (RSVPC)

  RSVPC is a PC DOS-to-DOS application that allows an IBM Support Representative to call a PC at your site, using a dial-up phone line, and view the data presented on your terminal.

- Premium Response

  Where available, requesting Premium Response means that IBM guarantee to respond within a specified time frame.

- Management Status Reports

  Each month, a usage report will be sent to your site detailing all usage of the Servicelink facilities used. This will include age of the reported problems and the number of outstanding calls at the time of the report. This is not available in all countries.

A.3 Online Maintenance Information System (OMIS)

The Online Maintenance Information System (OMIS) is also referred to as Enhanced Corrective Service, or Packaged Corrective Service.

For every SystemPac, ProductPac, FunctionPac, ServicePac, or every Selective Follow on Service update shipped from the Software Service Manufacturing Center, a profile is created, or updated from the customer′s SMP/E CSI. This profile lists all the FMIDS and PTFs applied to the system, and is kept permanently online. The profile number corresponds to the order number of an update and is written in the Installation Guide for the relevant update. The profile number should be the number of the last update that is installed on the system that is being referenced.

An OMIS update provides service built against the profile kept at the Software Service Manufacturing Center. An update contains not only the specific PTFs ordered, but also all requisites for these PTFs, which are tailored to the system.

When a call is placed with the IBM Software Support Center, a fix may be identified or a PSR may want more detailed information of the FMIDs or PTFs that are installed. A PSR signs on to the Software Service Manufacturing Center′s OMIS System where the profile number is entered. The OMIS profile for this customer can be browsed for further information on what is installed. IBM′s pre/coreq checking utilities can also be used against the profile to determine the exact number of fixes that have to be applied. The fixes can be ordered there and then, since there is an option for shipment. The profile is updated to reflect the shipment. The resulting update contains the selected service, installation jobs, and an installation guide.

As only the profile, and not the CSI, is kept online, selected service does not undergo an SMP/E APPLY CHECK.

If applying products or service outside of an update, a fresh copy of the CSI should be sent to the Software Service Manufacturing Center, where a new profile can be created to keep OMIS in synchronization.

A.3.1 OMIS Features

OMIS is shipped for all PTFs requested as a result of placing a call to the IBM Software Support Center, providing that a current, up-to-date profile is held by the Software Service Manufacturing Center, and that the correct profile number is given to the IBM Software Support Center.

A.4 MVS HealthCheck

MVS HealthCheck is a service available in some countries that provides a comprehensive set of service level reports for MVS, CICS, DB2, IMS, NCP, and related licensed programs. It assists in maintaining software subsystem availability by providing enhanced software service level information and complements the MVS CustomPacs.

The information supplied in the MVS HealthCheck is based on the data collected from your existing SMP/E Consolidated Software Inventory (CSI) data set. This data set, which contains a history of the software products and service applied to a system, is used by IBM to compare your software levels with the IBM-defined ideal service level. A hardcopy report is returned that contains the latest available information and is customized to, and uniquely relevant to, the supplied CSI.

MVS HealthCheck not only determines the top-of-chain PTFs, but works out the entire set of PTFs that would be needed to get the system to the IBM ideal service level. Often, this is in the order of 2000 PTFs for a typical MVS CSI. Each of the recommended HIPER PTFs has its cover letter printed at the back of the report.

Appendix B. SMPEUTIL

B.1 Description

SMPEUTIL is a collection of utility programs that can be used in the MVS SMP/E environment. SMPEUTIL is a follow-on to the SMPEHRPT tool, which has been available in some countries as an offering.

B.2 SMPEUTIL Functions

SMPEUTIL consists of the following functions:

- REPORT ERRSYSMODS (called the “SMARTMVS report program”)
- REPORT ZONEDEFINITIONS (called the “ZoneChecker”)
- GENERATE SUBSET (called the “GENLITE process”)

B.2.1 REPORT ERRSYSMODS

The REPORT ERRSYSMODS function produces a “health” report on any SMP/E-managed OS/390 system or subsystem. This report is based on the service level of that system and the current IBM knowledge of PTF and product quality, as expressed by SMP/E HOLD statements. The report shows those SYSMODS that are found to be in error (PE), but are integrated in the system without the fixing SYSMOD.

SMP/E Release 6 implements the main part of this function with the SMP/E REPORT ERRSYSMODS command. See *SMP/E Reference* for more information on the SMP/E REPORT ERRSYSMODS command. The REPORT ERRSYSMODS function, shipped as part of the SMPEUTIL utility program, goes one step further than the SMP/E REPORT ERRSYSMODS command, in that it supports report data from other offerings, such as SMARTMVS and MVSFMIDS. See section B.2.1.1, “Related Offerings.” The data provided by these offerings include the notification of highly pervasive APARS (HIPERS).

Note: If you are using standard HOLDDATA, (not SMARTMVS enriched) the SMP/E Release 6 REPORT ERRSYSMODS command gives you the same results as the SMPEUTIL REPORT ERRSYSMODS function.

B.2.1.1 Related Offerings

Other offerings that may be of interest to the SMPEUTIL user are the two described below. These offerings are *not* available in all countries and you should contact your IBM representative for availability information.

- SMARTMVS
  - The SMARTMVS package includes a regular delivery of up-to-date, data enriched, SMP/E HOLDDATA for all products supported by the normal IBM preventive service process (that is, PUT/CBPDO service supported products). SMARTMVS data is enriched HOLDDATA that utilizes the COMMENT operand of the ++HOLD statement to describe HIPER information. Additionally, a SMARTMVS user may subscribe to weekly updates of HOLDDATA. These weekly updates are distributed by the Software Production Center (SPC) and the Software Service Competence

Center (SSCC) in Copenhagen. They are also available via a subscription to the SMARTMVS PACKAGE on MVSTOOLS.

SMARTMVS data is *not* a prerequisite for using SMPEUTIL, but the REPORT ERRSYSMODS output report will be of more value to the user when SMARTMVS data is also used.

- MVSFMIDS
  - MVSFMIDS is a separate package that provides separate titles for FMIDS. SMPEUTIL REPORT ERRSYSMODS can then use these product titles to further enhance its various reports. However, this data is not required to run REPORT ERRSYSMODS.

B.2.2 REPORT ZONEDEFINITIONS

The REPORT ZONEDEFINITIONS program was developed to verify that only relevant service was being RECEIVED into a customer′s global zone.

Also, selected data from all zones in a specific global zone environment is cross-checked to verify that inconsistencies or counter-productive definitions do not exist.

In general, the REPORT ZONEDEFINITIONS function attempts to verify that data is correctly and completely defined for entries such as:

- ZONE entries themselves
- DDDEFs
- ZONESETs
- FMIDSETs
- OPTIONS entries
- FMIDs
- SRELs

B.2.3 GENERATE SUBSET

The GENERATE SUBSET or GENLITE program was developed in 1989 to show that it was possible to create a “new system” containing only a subset of the “from system.” The name comes from the fact that this process is “lighter” in content than the normal SMP/E GENERATE process.

The objective of the GENLITE process is to generate a system or subsystem from a given system. The newly created system or subsystem will contain *only* a subset of the products (FMIDs) included in the original (source) system. The GENERATE SUBSET function can be used as part of the traditional cloning process. The process is not unlike the IPOGEN process, which utilizes the SMP/E GENERATE command.

Appendix C. BUILDMCS Command

The BUILDMCS command provides customers with a more automated and less error-prone process for propagating products from one set of target and distribution zones and associated libraries to another set. This command generates the MCS and JCLIN required to reinstall the specified FMIDs.

C.1 Overview

The BUILDMCS command provides facilities to help copy a product from one SMP/E environment and install it into another. The BUILDMCS command is not intended to be used with all products; it is intended to be used for products that have no intersections with other products. These intersections come in two forms:

1. **Shared Load Modules**

   A shared load module is any load module that contains modules from more than one product. If the product to be copied supplies modules that reside in load modules along with modules from other products, then that product has shared load modules.

2. **Common Elements**

   A common element is any element with the same name and type that is supplied by more than one product. One product may take ownership of the element from another product using the VERSION operand.

If a product has either of the above types of intersections with another product, the BUILDMCS output may be incorrect because SMP/E does not have enough information in the zone entries to correctly identify a corresponding superseding function SYSMOD.

Following is a list of items to watch out for when using the BUILDMCS command because not enough information resides in the zone entries.

Element Versioning

If the original ++FUNCTION or any PTF for the product to be copied supplied elements using the VERSION operand on the element MCS, the VERSION operand will not be included on the element MCS created by the BUILDMCS command. The version information is not saved in the zone entries during APPLY or ACCEPT processing, and therefore is not available for the BUILDMCS command.

Macros Causing Assemblies

If the original ++FUNCTION or any PTFs for the product to be copied supplied macros using the ASSEM or PREFIX operands on the ++MAC MCS, these operands will not be included by the BUILDMCS command. This information is not saved in the zone entries during APPLY or ACCEPT processing, and therefore is not available for the BUILDMCS command.

Move, Rename, and Delete MCS

If the original ++FUNCTION or any PTFs for the product to be copied supplied ++MOVE, ++RENAME, or ++DELETE MCS, these MCS will not be included in the MCS created by the BUILDMCS command.

Load Module Definition

It is possible to have more than one product supply modules that reside in a single load module. Using the zone entry information, SMP/E cannot determine which product supplied the JCLIN link edit step to define the load module. Therefore, it is possible the BUILDMCS command will create a JCLIN link edit step to define a load module even though the product to be copied did not originally supply the JCLIN or define the load module.

For more information, see *OS/390 Release 2 SMP/E User's Guide*.

Appendix D. Sample JCL for Cloning

This appendix contains the flow and the related sample JCL to build an OS/390 clone. The initial environment is formed by the maintenance SYSRES volumes (OS3RS1 and OS3RS2), the SMP/E CSI and related data sets that map the target data sets (on volume SMPEV1), and the current production system master catalog (MCAT.OS3RSA.VOS3CAT on volume OS3CAT). The environment being created includes the new SYSRES volumes (OS3RSC and OS3RSD), the SMP/E CSI and related data sets that map the new target data sets (on volume TOTSY1), and the new production system master catalog (MCAT.OS3RSC.VOS3CAT on volume OS3CAT). Figure 71 illustrates the volume layout.

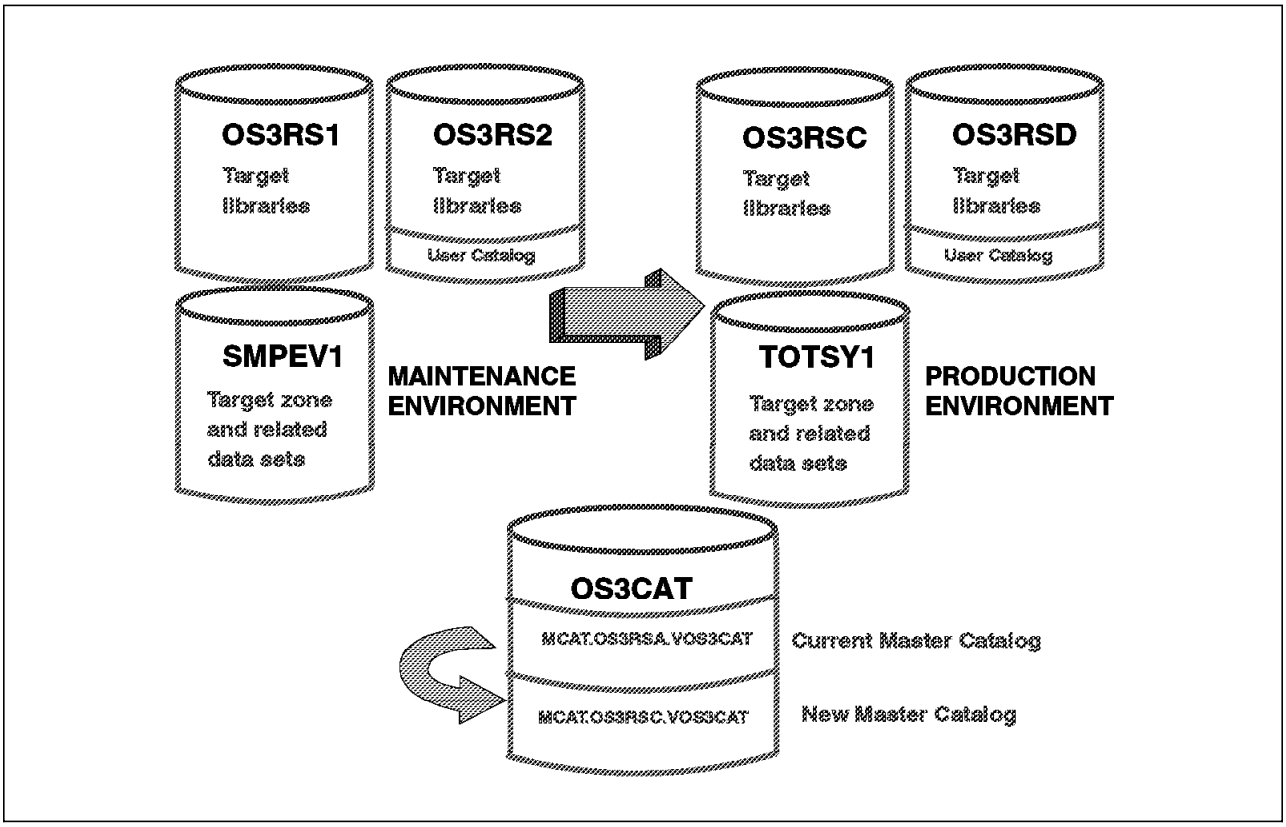


Figure 71. Volume Layout

The following text describes to jobs that are used to create the new production environment:

1. Initialize the new target volumes (OS3RSC and OS3RSD).

```
//INIT1 EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
         INIT -
           UNITADDRESS(0CFD) -
           DEVTYP(3390) -
           VOLID(OS3RSC) -
           VTOC(1113,0,90) -
           INDEX(1110,0,45) -
           PURGE -
```

```
    NOVERIFY
  INIT -
    UNITADDRESS(0CFE) -
    DEVTYP(3390) -
    VOLID(OS3RSD) -
    VTOC(1113,0,90) -
    INDEX(1110,0,45) -
    PURGE -
    NOVERIFY
```

2. Delete the old versions of the SMP/E data sets.

   As we reuse the target volumes, we must delete the old SMP/E data sets that were associated with a previous version of the target volumes.

```
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
   DELETE SMPE.MVST10C.SMPLTS
   DELETE SMPE.MVST10C.SMPMTS
   DELETE SMPE.MVST10C.SMPSCDS
   DELETE SMPE.MVST10C.SMPSTS
   DELETE SMPE.MVST10C.SMPTLOG
   DELETE SMPE.MVST10C.SMPTLOGA
   DELETE SMPE.MVST20C.SMPLTS
   DELETE SMPE.MVST20C.SMPMTS
   DELETE SMPE.MVST20C.SMPSCDS
   DELETE SMPE.MVST20C.SMPSTS
   DELETE SMPE.MVST20C.SMPTLOG
   DELETE SMPE.MVST20C.SMPTLOGA
   DELETE SMPE.OS3RSC.CSI
/*
```

3. Copy the contents of the maintenance SYSRES and spill volumes to the new target volumes.

   First allocate a new user catalog on the spill volume. The data sets on the SYSRES volume (OS3RS1) are copied without any catalog activity, as these data sets are cataloged indirectly (DEVT(0000) VOLUMES(******)). The data sets on the spill volume (OS3RS2) are copied and cataloged in the user catalog on the spill volume.

```
//COPY1    EXEC PGM=IDCAMS
//FILE     DD DISP=SHR,UNIT=3390,VOL=SER=OS3RSD
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    EXPORT 'CATALOG.TOTICF1.VOS3RSD' DISCONNECT
    SET LASTCC=0
    SET MAXCC=0
    DEFINE -
      USERCATALOG                    -
            (NAME (CATALOG.TOTICF1.VOS3RSD) -
            VOLUME(OS3RSD)          -
            ICFCATALOG                -
            STRNO(8)                  -
            CYLINDERS(10 5 )          -
            SHAREOPTIONS(3 4))        -
      DATA(BUFND(8)                   -
            NAME (CATALOG.TOTICF1.VOS3RSD.DATA) -
            CYLINDERS(10 5)           -
            CONTROLINTERVALSIZE(4096) -
            FREESPACE(20 20)          -
```

```
            RECORDSIZE(100 32400))    -
      INDEX(BUFNI(7)                  -
            NAME (CATALOG.TOTICF1.VOS3RSD.INDEX) -
            TRACKS(4)                 -
            CONTROLINTERVALSIZE(4096) -
            IMBED                     -
            REPLICATE)
/*
//STEP2    EXEC PGM=ADRDSSU,REGION=0M,COND=(0,LT)
//SYSPRINT DD   SYSOUT=*
//RES1IN   DD   UNIT=3390,VOL=SER=OS3RS1,DISP=SHR
//RES2IN   DD   UNIT=3390,VOL=SER=OS3RS2,DISP=SHR
//RES1OUT  DD   UNIT=3390,VOL=SER=OS3RSC,DISP=SHR
//RES2OUT  DD   UNIT=3390,VOL=SER=OS3RSD,DISP=SHR
//SYSIN    DD   *
 PARALLEL
 COPY INDD(RES1IN) OUTDD(RES1OUT) -
     DS(EXCLUDE(SYS1.VTOCIX.* -
               SYS1.VVDS.* -
               )) -
     TOLERATE(ENQF) SHARE ALLEXCP ALLDATA(*) CANCELERROR
 COPY INDD(RES2IN) OUTDD(RES2OUT) -
     DS(EXCLUDE(SYS1.VTOCIX.* -
               SYS1.VVDS.* -
               CATALOG.** -
               )) -
     TOLERATE(ENQF) SHARE ALLEXCP ALLDATA(*) CANCELERROR -
     RECATALOG(CATALOG.TOTICF1.VOS3RSD)
/*
```

4. Write the IPL text on the new SYSRES volume.

```
//IPLTEXT PROC VOL=,UNIT=3390
//DSF     EXEC PGM=ICKDSF,REGION=1M
//SYSPRINT DD  SYSOUT=*
//IPLVOL   DD  DISP=SHR,VOL=SER=&VOL,UNIT=&UNIT
//IPLTEXT  DD  DSN=SYS1.SAMPLIB(IPLRECS),
//             DISP=SHR,UNIT=&UNIT,VOL=SER=&VOL
//         DD  DSN=SYS1.SAMPLIB(IEAIPL00),
//             DISP=SHR,UNIT=&UNIT,VOL=SER=&VOL
//         PEND
//STEP1    EXEC IPLTEXT,VOL=OS3RSC,UNIT=3390
//SYSIN    DD  *
   REFORMAT  DDNAME(IPLVOL)                                    -
             IPLDD(IPLTEXT)                                    -
             NOVERIFY                                          -
             BOOTSTRAP
/*
```

5. Copy the SMP/E target zone CSI and related data sets to new volume TOTSY1.

   These data sets will map the new target libraries on the target volumes.

```
//SMPCOPY  EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SMPEVOL  DD   UNIT=3390,VOL=SER=OS3SY1,DISP=SHR
//SYSIN    DD   *
 COPY OUTDD(SMPEVOL) -
     DS(INCLUDE(SMPE.MVST100.**)) -
     RENAMEU(SMPE.MVST100.**,SMPE.MVST10C.**)  -
```

```
    CATALOG -
    TOLERATE(ENQF) SHARE ALLEXCP ALLDATA(*) CANCELERROR
 COPY OUTDD(SMPEVOL) -
    DS(INCLUDE(SMPE.MVST200.**)) -
    RENAMEU(SMPE.MVST200.**,SMPE.MVST20C.**)  -
    CATALOG -
    TOLERATE(ENQF) SHARE ALLEXCP ALLDATA(*) CANCELERROR
 COPY OUTDD(SMPEVOL) -
    DS(INCLUDE(SMPE.OSR2TZN.**)) -
    RENAMEU(SMPE.OSR2TZN.**,SMPE.OS3RSC.**)  -
    CATALOG -
    TOLERATE(ENQF) SHARE ALLEXCP ALLDATA(*) CANCELERROR
/*
```

6. Edit the new SMP/E CSI to reflect the new target libraries on volumes OS3RSC and OS3RSD.

   The job:

   - Deletes the old version of the zone index
   - Renames the copied zone index to the new name
   - Updates all of the VOLSERs in the target zone
   - Updates the SMP/E target zone data set names

```
//SMPEDIT  EXEC PGM=GIMSMP,TIME=1440,REGION=0M
//SMPCSI   DD DISP=SHR,DSN=SMPE.GLOBAL.CSI
//SYSPRINT DD SYSOUT=*
//SMPRPT   DD SYSOUT=*
//SMPOUT   DD SYSOUT=*
//SMPCNTL  DD *
  SET BDY(GLOBAL) .
  UCLIN.
  DEL GLOBALZONE ZONEINDEX((MVST10C)) .
  DEL GLOBALZONE ZONEINDEX((MVST20C)) .
  ENDUCL.
  ZONERENAME(MVST100)
    TO(MVST10C)
    OPTIONS(OPTMVST)
    RELATED(MVSD100)
   NEWDATASET(SMPE.OS3RSC.CSI) .
  ZONERENAME(MVST200)
    TO(MVST20C)
    OPTIONS(OPTMVST)
    RELATED(MVSD200)
   NEWDATASET(SMPE.OS3RSC.CSI) .
  SET        BDY (MVST10C) .
  ZONEEDIT   DDDEF.
      CHANGE  VOLUME(OS3RS1,OS3RSC) .
      CHANGE  VOLUME(OS3RS2,OS3RSD) .
  ENDZONEEDIT .
  UCLIN.
        REP DDDEF(SMPLTS)  DATASET(SMPE.MVST10C.SMPLTS) .
        REP DDDEF(SMPMTS)  DATASET(SMPE.MVST10C.SMPMTS) .
        REP DDDEF(SMPSCDS) DATASET(SMPE.MVST10C.SMPSCDS) .
        REP DDDEF(SMPSTS)  DATASET(SMPE.MVST10C.SMPSTS) .
        REP DDDEF(SMPLOG)  DATASET(SMPE.MVST10C.SMPTLOG) .
        REP DDDEF(SMPLOGA) DATASET(SMPE.MVST10C.SMPTLOGA) .
  ENDUCL.
  SET        BDY (MVST20C) .
  ZONEEDIT   DDDEF.
      CHANGE  VOLUME(OS3RS1,OS3RSC) .
```

```
      CHANGE  VOLUME(OS3RS2,OS3RSD) .
  ENDZONEEDIT .
  UCLIN.
       REP DDDEF(SMPLTS)  DATASET(SMPE.MVST1OC.SMPLTS) .
       REP DDDEF(SMPMTS)  DATASET(SMPE.MVST1OC.SMPMTS) .
       REP DDDEF(SMPSCDS) DATASET(SMPE.MVST1OC.SMPSCDS) .
       REP DDDEF(SMPSTS)  DATASET(SMPE.MVST1OC.SMPSTS) .
       REP DDDEF(SMPLOG)  DATASET(SMPE.MVST1OC.SMPTLOG) .
       REP DDDEF(SMPLOGA) DATASET(SMPE.MVST1OC.SMPTLOGA) .
  ENDUCL.
/*
```

7. Allocate the new production master catalog.

```
//MCAT1 EXEC PGM=IDCAMS
//FILE     DD DISP=SHR,UNIT=3390,VOL=SER=OS3CAT
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DELETE MCAT.OS3RSC.VOS3CAT UCAT RECOVERY
    SET LASTCC=0
    SET MAXCC=0
    DEFINE -
      USERCATALOG                    -
            (NAME (MCAT.OS3RSC.VOS3CAT) -
            VOLUME(OS3CAT)          -
            ICFCATALOG                -
            STRNO(8)                  -
            CYLINDERS(10 5 )          -
            SHAREOPTIONS(3 4))        -
      DATA(BUFND(8)                   -
            NAME (MCAT.OS3RSC.VOS3CAT.DATA) -
            CYLINDERS(10 5)           -
            CONTROLINTERVALSIZE(4096) -
            FREESPACE(20 20)          -
            RECORDSIZE(100 32400))    -
      INDEX(BUFNI(7)                  -
            NAME (MCAT.OS3RSC.VOS3CAT.INDEX) -
            TRACKS(4)                 -
            CONTROLINTERVALSIZE(4096) -
            IMBED                     -
            REPLICATE)
```

8. Run the MCNVTCAT utility to create the control statements used to build the new master catalog.

```
//DELET1  EXEC PGM=IDCAMS,COND=(4000,LT)
//*-------------------------------------------------------------------
//*  DELETE DATA SET DSN=SYSPOK.OT0000014.CONVERT.
//*  STEP CREATE5 WILL ALLOCATE THIS DATA SET.
//*-------------------------------------------------------------------
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DELETE SYS2.CLONE.BUILDCAT NONVSAM
  SET LASTCC=0
  SET MAXCC=0
/*
//NOTOK   EXEC PGM=CPPMAXRC,COND=((0,GE,DELET1),(4000,LT))
//*
//ASSEM2  EXEC PGM=ASMA90,
//            PARM='DECK,NOOBJ',COND=(4000,LT)
//*-------------------------------------------------------------------
```

```
//* ASSEMBLE INPUT PARAMETERS
//*--------------------------------------------------------------------
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DSN=IPO1.SAMPLIB,
//            DISP=SHR
//SYSPUNCH DD DSN=&CARD,DISP=(,PASS),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
//SYSIN    DD *
  PUNCH ' LISTCAT ALL OUTFILE(OUTFILE) -'
  PUNCH '   CAT(MCAT.OS3RSA.VOS3CAT)'
&PERIOD.END     END
/*
//NOTOK   EXEC PGM=CPPMAXRC,COND=((0,GE,ASSEM2),(4000,LT))
//*
//LIST3   EXEC PGM=IDCAMS,COND=(4000,LT)
//*--------------------------------------------------------------------
//* LIST VSAM SOURCE CATALOG TO BE CONVERTED
//*--------------------------------------------------------------------
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=&CARD,DISP=(OLD,PASS)
//OUTFILE  DD DSN=&OUT,DISP=(,PASS),
//            UNIT=SYSDA,
//            SPACE=(CYL,(3,3)),
//            DCB=(RECFM=VBA,LRECL=125,BLKSIZE=625)
/*
//NOTOK   EXEC PGM=CPPMAXRC,COND=((4,GE,LIST3),(4000,LT))
//*
//GENER4  EXEC PGM=MCNVTCAT,
//             PARM='MCAT.OS3RSC.VOS3CAT',
//             COND=(4000,LT)
//*--------------------------------------------------------------------
//* EXECUTE MCNVTCAT(CBIPO-SUPPLIED PROGRAM)
//* TO GENERATE IDCAMS CONTROL CARDS FROM LISTCAT OUTPUT.
//*--------------------------------------------------------------------
//STEPLIB  DD DSN=SYSPOK.OT000014.LOADLIB,
//            DISP=SHR
//SYSEMSG  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//LIST     DD DSN=&OUT,DISP=(OLD,PASS)
//CONTR1   DD DSN=&CARD1,DISP=(NEW,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
//CONTR2   DD DSN=&CARD2,DISP=(NEW,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
//CONTR3   DD DSN=&CARD3,DISP=(NEW,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
//CONTR4   DD DSN=&CARD4,DISP=(NEW,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
//CONTR5   DD DSN=&CARD5,DISP=(NEW,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
```

```
//CONTR6   DD DSN=&CARD6,DISP=(NEW,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
//CONTR7   DD DSN=&CARD7,DISP=(NEW,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
//CONTRS   DD DSN=&OUTDS,DISP=(NEW,PASS),
//            UNIT=SYSDA,
//            SPACE=(CYL,(1,1))
/*
//NOTOK   EXEC PGM=CPPMAXRC,COND=((0,GE,GENER4),(4000,LT))
//*
//CREATE5 EXEC PGM=IEBGENER,COND=(4000,LT)
//*-------------------------------------------------------------------
//* CREATE A PARTITIONED DATA SET FROM GENER4 OUTPUT.
//*-------------------------------------------------------------------
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=&OUTDS,DISP=(OLD,PASS)
//SYSUT2   DD DSN=SYS2.CLONE.BUILDCAT,
//            DISP=(NEW,CATLG),
//            VOL=SER=TOTTS4,UNIT=3390,
//            DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//            SPACE=(CYL,(1,1,5)),LABEL=EXPDT=00000
//SYSIN    DD *
  GENERATE MAXNAME=7,MAXFLDS=7,MAXGPS=7
        MEMBER  NAME=(ALIAS)                          1
ALIAS   RECORD  IDENT=(5,'ALIAS',4),FIELD=(80)        2
        MEMBER  NAME=(IMPORT)                         3
IMPORT  RECORD  IDENT=(6,'IMPORT',4),FIELD=(80)       4
        MEMBER  NAME=(NONVSAM)                        5
NONVSAM RECORD  IDENT=(7,'NONVSAM',4),FIELD=(80)      6
        MEMBER  NAME=(SYSCTLG)                        7
SYSCTLG RECORD  IDENT=(7,'SYSCTLG',4),FIELD=(80)      8
        MEMBER  NAME=(SYS1)                           9
SYS1    RECORD  IDENT=(4,'SYS1',4),FIELD=(80)        10
        MEMBER  NAME=(GDG)                           11
GDG     RECORD  IDENT=(3,'GDG',4),FIELD=(80)         12
        MEMBER  NAME=(MISC)                          13
MISC    RECORD  IDENT=(4,'MISC',4),FIELD=(80)        14
/*
//NOTOK   EXEC PGM=CPPMAXRC,COND=((0,GE,CREATE5),(4000,LT))
//*
```

9. Create the DEFINE statements required to recatalog any VSAM objects in the the new master catalog. Appendix E, “Recataloging VSAM Objects” describes this process.

10. Build the new master catalog.

```
//MCAT3    EXEC  PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DISP=SHR,DSN=SYS2.CLONE.BUILDCAT(IMPORT)
//         DD  DISP=SHR,DSN=SYS2.CLONE.BUILDCAT(NONVSAM)
//         DD  DISP=SHR,DSN=SYS2.CLONE.BUILDCAT(RECAT)
//         DD  DISP=SHR,DSN=SYS2.CLONE.BUILDCAT(SYS1)
```

Appendix E. Recataloging VSAM Objects

This appendix contains two EXECs that create IDCAMS commands to be used to recatalog the VSAM objects from an existing master catalog into a new master catalog. We've tested these EXECs with PAGESPACEs, STGINDEX data sets, SMF data sets, the APPC data sets SYS1.APPCTP and SYS1.APPCSI, and a few other VSAM clusters, including those used to define a SMS environment. These EXECs are distributed on an as-is basis. They have not been subjected to any formal testing. The REXX coding probably is not the best, but it works!

The output from the CATCVTM job creates a member named MISC in the output data set. All of the lines appear as comments. Remove the lines for data sets that need not be recreated (if any), remove the line for the current master catalog's self-defining record, remove any other comment lines, and strip the comment delimiters from the remaining lines. Surround the data set names to be defined with a leading and trailing quote. Save this data set. Here is an example of the first few lines of our data set:

```
'PAGE.SC42.COMMON'
'PAGE.SC42.LOCAL1'
'PAGE.SC42.LOCAL2'
```

Run the RECATM EXEC. The EXEC expects three parameters:

1. The name of the sequential data set or PDS(E) member containing the names of the VSAM objects to be defined
2. The name of the sequential data set or PDS(E) member to receive the definitions that the EXEC will create
3. The name of the new master catalog

For example:

```
READY
recatm 'sys2.clone.buildcat(misc)' 'sys2.clone.buildcat(recat)'
       'MCAT.OS3RSC.VOS3CAT'
READY
```

Execute the IDCAMS statements in the output data set to create the required definitions in the new master catalog. Here is an example of the definitions produced for the three data sets shown above:

```
DEFINE PAGESPACE (NAME('PAGE.SC42.COMMON') -
                  VOLUMES(MVS004) -
                  RECATALOG -
                      ) -
        CATALOG('MCAT.OS3RSC.VOS3CAT')

DEFINE PAGESPACE (NAME('PAGE.SC42.LOCAL1') -
                  VOLUMES(MVS004) -
                  RECATALOG -
                      ) -
        CATALOG('MCAT.OS3RSC.VOS3CAT')

DEFINE PAGESPACE (NAME('PAGE.SC42.LOCAL2') -
                  VOLUMES(MVS005) -
                  RECATALOG -
                      ) -
```

```
        CATALOG('MCAT.OS3RSC.VOS3CAT')
```

The RECATM EXEC

```
/* REXX */
parse arg in_dsn out_dsn catname .
"alloc f(recatm) da("||in_dsn||") shr reuse"
"execio * diskr recatm (stem in_line."
"alloc f(recato) da("||out_dsn||") old reuse"
"execio 0 diskw recato (open"
do i = 1 to in_line.0
  parse var in_line.i dsn .
  if left(dsn,2) /= '/*' then
    "%recatx" dsn catname
end
"execio * diskr recatm (finis"
"execio 0 diskw recato (finis"
"free f(recatm,recato)"
exit 0
```

The RECATX EXEC (called by RECATM)

```
/* REXX */
parse arg dsn catname
address TSO
x = outtrap('LINE.','*','NOCONCAT')
"listc ent("||dsn||") all"
call do_inits
do i = 1 to line.0
  parse var line.i field1 the_rest
  select
    when field1 = 'CLUSTER' | field1 = 'PAGESPACE' then
      do
        object_type = field1
        parse var the_rest . object_name
      end
    when field1 = 'DATA' then
      parse var the_rest . data_name
    when field1 = 'INDEX' then
      parse var the_rest . index_name
    when left(field1,18) = 'VOLSER------------' then
      do
        parse var field1 'VOLSER------------' volume_name
        if pos(volume_name,volume_list) = 0 then
          if volume_list = '' then
            volume_list = volume_name
          else
            volume_list = volume_list||","||volume_name
      end
    when pos(' INDEXED ', line.i) /= 0 then
      attr = 'INDEXED'
    when pos(' NONINDEXED ', line.i) /= 0 then
      attr = 'NONINDEXED'
    when pos(' LINEAR ', line.i) /= 0 then
      attr = 'LINEAR'
    when pos(' NUMBERED ', line.i) /= 0 then
      attr = 'NUMBERED'
    otherwise
      nop
  end
```

```
end

/* NONINDEXED can not be specified for a PAGESPACE */
if object_type = 'PAGESPACE' then
  attr = ''

oline.1 = "  DEFINE" object_type "(NAME('"||object_name||"') - "
oline.2 = "                     VOLUMES("||volume_list||") -"
oline.3 = "                     RECATALOG -"
oline.4 = "                      " attr " ) -"
oline.5 = "          CATALOG("||catname||")"
oline.6 = " "
oline.0 = 6
"execio * diskw recato (stem oline."

/* call debug */

x = outtrap('off')
exit 0

do_inits:

object_type = ''
object_name = ''
data_name = ''
index_name = ''
volume_list = ''
attr = ''
return

debug:

say object_type
say object_name
if data_name /= '' then
  say data_name
if index_name /= '' then
  say index_name
say volume_list
select
  when indexed then say "INDEXED"
  when nonindexed then say "NONINDEXED"
  when linear then say "LINEAR"
  when numbered then say "NUMBERED"
  otherwise say "Bug in indexed/nonindexed/linear/numbered."
end
say catname
say ' '

return
```

Appendix F. Special Notices

This publication is intended to help system programmers involved with the installation and maintenance of an OS/390 operating environment. The information in this publication is not intended as the specification of any programming interfaces that are provided by the OS/390 product. See the PUBLICATIONS section of the IBM Programming Announcement for the OS/390 product for more information about what publications are considered to be product documentation.

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 program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM
BookManager
BookMaster
C++/MVS
CBIPO
CBPDO
CICS
CICS/ESA
CICS/MVS
CICSPlex
Common User Access
DATABASE 2
DB2
DFSMS
DFSMS/MVS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
Enterprise System/3090
Enterprise System/9000
Enterprise Systems Architecture/390
Enterprise Systems Connection Architecture
ES/3090
ES/9000
ESA/390
ESCON
FunctionPac
GDDM
Hardware Configuration Definition
IBM
IBMLink
IMS
IMS/ESA
Language Environment
MVS/ESA
NetView
OpenEdition
Parallel Sysplex
PR/SM
Print Services Facility
ProductPac
PS/2
PSF
RACF *
S/390
ServicePac
SolutionPac
SupportPac
Sysplex Timer
System/390
SystemPac
VTAM

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo
are trademarks or registered trademarks of Microsoft Corporation.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other trademarks are trademarks of their respective companies.

Appendix G. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook. A publication whose order number begins with the prefix **LY** is available to IBM-licensed customers only.

G.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 233.

- *MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit*, SG24-4529
- *MVS/ESA HCD and Dynamic I/O Reconfiguration Primer*, SG24-4037

G.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection (available soon)	SBOF-7250	SK2T-8042

G.3 Other Publications

- OS/390 BDT

Short Title	Title	Order Number
OS/390 BDT Overview	*OS/390 V1R2.0 BDT Overview*	GC28-1741
OS/390 BDT Installation	*OS/390 V1R2.0 BDT Installation*	SC28-1742
OS/390 BDT File-to-File Transaction Guide	*OS/390 V1R2.0 BDT File-to-File Transaction Guide*	SC28-1743
OS/390 BDT Commands	*OS/390 V1R2.0 BDT Commands*	SC28-1744
OS/390 BDT Messages and Codes	*OS/390 V1R2.0 BDT Messages and Codes*	SC28-1745

- BookManager

Short Title	Title	Order Number
IBM BookManager READ/MVS and BUILD/MVS: General Information	*IBM BookManager READ/MVS and BUILD/MVS: General Information*	GC38-2832
IBM BookManager READ/MVS: Getting Started	*IBM BookManager READ/MVS: Getting Started*	SC38-2033
IBM BookManager READ/MVS: Displaying Online Books	*IBM BookManager READ/MVS: Displaying Online Books*	SC38-2034
IBM BookManager READ/MVS: Installation Planning and Customization	*IBM BookManager READ/MVS: Installation Planning and Customization*	SC38-2035
IBM BookManager BUILD/MVS: Preparing Online Books	*IBM BookManager BUILD/MVS: Preparing Online Books*	SC38-2036
IBM BookManager BUILD/MVS: Installation Planning and Customization	*IBM BookManager BUILD/MVS: Installation Planning and Customization*	SC38-2037
Writing Softcopy for BookManager	*Writing Softcopy for BookManager*	SC22-0011

- OS/390 OpenEdition

Short Title	Title	Order Number
OS/390 OpenEdition MVS Introduction	*OS/390 OpenEdition MVS Introduction*	GC28-1889
OS/390 OpenEdition MVS POSIX.1 Conformance Document	*OS/390 OpenEdition MVS POSIX.1 Conformance Document*	GC28-1895
OS/390 OpenEdition MVS POSIX.2 Conformance Document	*OS/390 OpenEdition MVS POSIX.2 Conformance Document*	GC28-1896
OS/390 OpenEdition MVS User's Guide	*OS/390 OpenEdition MVS User's Guide*	SC28-1891
OS/390 OpenEdition MVS Command Reference	*OS/390 OpenEdition MVS Command Reference*	SC28-1892
OS/390 OpenEdition MVS Programming Tools	*OS/390 OpenEdition MVS Programming Tools*	SC28-1904
OS/390 OpenEdition MVS Messages and Codes	*OS/390 OpenEdition MVS Messages and Codes*	SC28-1908
OS/390 OpenEdition MVS Programming: Assembler Callable Services Reference	*OS/390 OpenEdition MVS Programming: Assembler Callable Services Reference*	SC28-1899
OS/390 OpenEdition MVS Planning	*OS/390 OpenEdition MVS Planning*	SC28-1890
OS/390 OpenEdition MVS File System Interface Reference	*OS/390 OpenEdition MVS File System Interface Reference*	SC28-1909
OS/390 OpenEdition MVS Using REXX and OpenEdition MVS	*OS/390 OpenEdition MVS Using REXX and OpenEdition MVS*	SC28-1905
C/MVS Library Reference: OpenEdition MVS Curses	*C/MVS Library Reference: OpenEdition MVS Curses*	SC23-3876
OS/390 OpenEdition MVS XPG4 Conformance Document	*OS/390 OpenEdition MVS XPG4 Conformance Document*	GC28-1897
C/C++ for MVS Library Reference	*C/C++ for MVS Library Reference*	SC23-3881
OS/390 OpenEdition MVS Communications Server Guide	*OS/390 OpenEdition MVS Communications Server Guide*	SC28-1906

- I/O Configuration Management

Short Title	Title	Order Number
OS/390 HCD Planning	*OS/390 Hardware Configuration Definition Planning*	GC28-1750
OS/390 HCD User's Guide	*OS/390 HCD User's Guide*	SC28-1848
HCD Messages	*OS/390 HCD Messages*	GC28-1849
HCD Scenarios	*OS/390 HCD Scenarios*	SC28-1850
HCD Reference Summary	*OS/390 HCD Reference Summary*	SX22-0043

- C/C++ Language

Short Title	Title	Order Number
OS/390 C/C++ Open Class Library User's Guide	*OS/390 Release 2 C/C++ Open Class Library User's Guide*	SC09-2363
OS/390 Class Library Reference	*OS/390 Release 2 C/C++ Class Library Reference*	SC09-2364
OS/390 C/C++ Database Access Class Library Utility Guide	*OS/ 390 Release 2 C/C++ Database Access Class Library Utility Guide*	SC09-2132
OS/390 C/C++ Language Reference	*OS/390 Release 2 C/C++ Language Reference*	SC09-2360
OS/390 C/C++ Compiler and Run-time Migration Guide	*OS/390 Release 2 C/C++ Compiler and Run-time Migration Guide*	SC09-2359
OS/390 C/C++ Master Index	*OS/390 Release 2 C/C++ Master Index*	SC28-2367
OS/390 C/C++ Programming Guide	*OS/390 Release 2 C/C++ Programming Guide*	SC09-2362
OS/390 C/C++ Reference Summary	*OS/390 Release 2 C/C++ Reference Summary*	SX09-1313
OS/390 C/C++ User's Guide	*OS/390 Release 2 C/C++ User's Guide*	SC09-2361
Debug Tool User's Guide and Reference	*OS/390 Release 2 C/C++ Debug Tool User's Guide and Reference*	SC09-2137
OS/390 C/C++ SOM-Enabled Class Library User's Guide and Reference	*OS/390 Release 2 C/C++ SOM-Enabled Class Library User's Guide and Reference*	SC09-2366
OS/390 C/C++ Run-Time Library Reference	*OS/390 Release 2 C/C++ Run-Time Library Reference*	SC28-1663

- RMF

Short Title	Title	Order Number
RMF Messages and Codes	*OS/390 RMF Messages and Codes*	GC28-1948
RMF Performance Management Guide	*OS/390 RMF Performance Management Guide*	SC28-1951
RMF User's Guide	*OS/390 RMF User's Guide*	SC28-1949
RMF Diagnosis Guide	*OS/390 RMF Diagnosis Guide*	LY28-1132
RMF Report Analysis	*OS/390 RMF Report Analysis*	SC28-1950
RMF Reference Summary	*OS/390 RMF Reference Summary*	SX22-0044

Short Title	Title	Order Number
RMF Programmer's Guide	*OS/390 RMF Programmer's Guide*	SC28-1952
RMF Data Areas	*OS/390 RMF Data Areas*	LY28-1134

- OpenEdition DCE

Short Title	Title	Order Number
OpenEdition DCE Application Development Guide: Introduction and Style	*OS/390 OpenEdition DCE Application Development Guide: Introduction and Style*	SC28-1587
OpenEdition DCE Application Development Guide: Core Components	*OS/390 OpenEdition DCE Application Development Guide: Core Components*	SC28-1588
OpenEdition DCE Application Development Guide: Directory Services	*OS/390 OpenEdition DCE Application Development Guide: Directory Services*	SC28-1589
OpenEdition DCE Command Reference	*OS/390 OpenEdition DCE Command Reference*	SC28-1585
OpenEdition DCE Administration Guide	*OS/390 OpenEdition DCE Administration Guide*	SC28-1584
OpenEdition DCE Application Development Reference	*OS/390 OpenEdition DCE Application Development Reference*	SC28-1590
DCE: Understanding the Concepts	*Distributed Computing Environment: Understanding the Concepts*	GC09-1478
OpenEdition DCE Configuring and Getting Started	*OS/390 OpenEdition DCE Configuring and Getting Started*	SC28-1583
OpenEdition DCE Introduction	*OS/390 OpenEdition DCE Introduction*	GC28-1581
OpenEdition DCE Messages and Codes	*OS/390 OpenEdition DCE Messages and Codes*	SC28-1591
OpenEdition DCE User's Guide	*OS/390 OpenEdition DCE User's Guide*	SC28-1586
OpenEdition DCE Security Server Overview	*OS/390 OpenEdition DCE Server Overview*	GC28-1938
OpenEdition DCE Application Support Supplementary Information	*Application Support Supplementary Information*	LY28-1100
OpenEdition DCE Application Support Programming Guide	*OpenEdition DCE Application Support for MVS/ESA Programming Guide*	SC09-1530
OpenEdition DCE Application Support Licensed Program Specifications	*OpenEdition DCE Application Support for MVS/ESA Licensed Program Specifications*	GC09-1701
OpenEdition DCE Application Support Configuration and Administration Guide	*OpenEdition DCE Application Support for MVS/ESA Configuration and Administration Guide*	SC09-1659

- LANRES, LAN Server

Short Title	Title	Order Number
OS/390 IBM LAN Server for MVS Guide	*OS/390 IBM LAN Server for MVS Guide*	SC28-1731
OS/390 IBM LAN Server for MVS Configuration Files and Commands	*OS/390 IBM LAN Server for MVS Configuration Files and Commands*	SC28-1732
OS/390 IBM LAN Server for MVS Installation Guide	*OS/390 IBM LAN Server for MVS Installation Guide*	GC28-1733
OS/390 LANRES Installation Guide	*OS/390 LAN Resource Extension and Services Installation Guide*	GC28-1736

Short Title	Title	Order Number
OS/390 LANRES Configuration Files and Commands	*OS/390 LAN Resource Extension and Services Configuration Files and Commands*	SC28-1735

- Multi-System Configuration Management

Short Title	Title	Order Number
OS/390 Parallel Sysplex Overview	*OS/390 Parallel Sysplex Overview: An Introduction*	GC28-1860
OS/390 Parallel Sysplex Systems Management	*OS/390 Parallel Sysplex Systems Management*	GC28-1861
OS/390 Parallel Sysplex Hardware and Software Migr	*OS/390 Parallel Sysplex Hardware and Software Migr*	GC28-1862
OS/390 Parallel Sysplex Application Migration	*OS/390 Parallel Sysplex Application Migration*	GC28-1863
OS/390 V1R2.0 MVS Setting Up a Sysplex	*OS/390 MVS Setting Up a Sysplex*	GC28-1779
OS/390 V1R2.0 MVS Sysplex Services Guide	*OS/390 MVS Programming: Sysplex Services Guide*	GC28-1771
OS/390 V1R2.0 MVS Sysplex Services Reference	*OS/390 MVS Programming: Sysplex Services Reference*	GC28-1772

- OS/390 Operating System

Short Title	Title	Order Number
OS/390 V1R2.0 MVS Auth Assembler Services Reference ALE-DYN	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 1 (ALESERV-DYNALLO)*	GC28-1764
OS/390 V1R2.0 MVS Auth Assembler Services Reference ENF-ITT	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 2 (ENFREQ-ITTFMTB)*	GC28-1765
OS/390 V1R2.0 MVS Auth Assembler Services Reference LLA-SDU	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 3 (LLACOPY-SDUMPX)*	GC28-1766
OS/390 V1R2.0 MVS Auth Assembler Services Reference SET-WTO	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 4 (SETFRR-WTOR)*	GC28-1767
OS/390 V1R2.0 MVS Extended Addressability Guide	*OS/390 MVS Programming: Extended Addressability Guide*	GC28-1769
OS/390 V1R2.0 MVS Assembler Services Guide	*OS/390 MVS Programming: Assembler Services Guide*	GC28-1762
OS/390 V1R2.0 MVS Assembler Services Reference	*OS/390 MVS Programming: Assembler Services Reference*	GC28-1910
OS/390 V1R2.0 MVS Auth Assembler Services Guide	*OS/390 MVS Programming: Authorized Assembler Services Guide*	GC28-1763
Introducing OS/390	*Introducing OS/390*	GC28-1725
OS/390 V1R2.0 MVS JCL User's Guide	*OS/390 MVS JCL User's Guide*	GC28-1758
OS/390 V1R2.0 MVS JCL Reference	*OS/390 MVS JCL Reference*	GC28-1757

Short Title	Title	Order Number
MVS OLTEP	*MVS OLTEP*	GC28-1448
MVS Remote Workstation Generation	*MVS Remote Workstation Generation*	GC28-1433
OS/390 V1R2.0 MVS Callable Services for HLL	*OS/390 MVS Programming: Callable Services for High-Level Languages*	GC28-1768
OS/390 V1R2.0 MVS Writing TPs for APPC/MVS	*OS/390 MVS Programming: Writing Transaction Programs for APPC/MVS*	GC28-1775
OS/390 V1R2.0 MVS Planning: APPC/MVS Management	*OS/390 MVS Planning: APPC/MVS Management*	GC28-1807
OS/390 V1R2.0 MVS IPCS Commands	*OS/390 MVS Interactive Problem Control System (IPCS) Commands*	GC28-1754
OS/390 V1R2.0 MVS IPCS User's Guide	*OS/390 MVS Interactive Problem Control System (IPCS) User's Guide*	GC28-1756
OS/390 V1R2.0 MVS IPCS Customization	*OS/390 MVS Interactive Problem Control System (IPCS) Customization*	GC28-1755
OS/390 V1R2.0 MVS Data Areas, Vol 1 (ABEP-DALT)	*OS/390 MVS Data Areas, Volume 1 (ABEP-DALT)*	LY28-1164
OS/390 V1R2.0 MVS Data Areas, Vol 2 (DCCB-ITTCTE)	*OS/390 MVS Data Areas, Volume 2 (DCCB-ITTCTE)*	LY28-1165
OS/390 V1R2.0 MVS Data Areas, Vol 3 (IVT-RCWK)	*OS/390 MVS Data Areas, Volume 3 (IVT-RCWK)*	LY28-1166
OS/390 V1R2.0 MVS Data Areas, Vol 4 (RD-SRRA)	*OS/390 MVS Data Areas, Volume 4 (RD-SRRA)*	LY28-1167
OS/390 V1R2.0 MVS Data Areas, Vol 5 (SSAG-XTLST)	*OS/390 MVS Data Areas, Volume 5 (SSAG-XTLST)*	LY28-1168
OS/390 V1R2.0 MVS Initialization and Tuning Guide	*OS/390 MVS Initialization and Tuning Guide*	SC28-1751
OS/390 V1R2.0 MVS Initialization and Tuning Reference	*OS/390 MVS Initialization and Tuning Reference*	SC28-1752
OS/390 V1R2.0 MVS Installation Exits	*OS/390 MVS Installation Exits*	SC28-1753
OS/390 V1R2.0 MVS Using the Functional Subsystem Interface	*OS/390 MVS Using the Functional Subsystem Interface*	SC28-1911
OS/390 V1R2.0 MVS Conversion Notebook	*OS/390 MVS Conversion Notebook*	GC28-1747
OS/390 V1R2.0 MVS Using the Subsystem Interface	*OS/390 MVS Using the Subsystem Interface*	SC28-1789
OS/390 V1R2.0 MVS System Commands Summary	*OS/390 MVS System Commands Summary*	GX22-0040
MVS Planning: Security	*MVS Planning: Security*	GC28-1439
OS/390 Up and Running!	*OS/390 Up and Running!*	GC28-1726
OS/390 V1R2.0 MVS System Commands	*OS/390 MVS System Commands*	GC28-1781
OS/390 V1R2.0 MVS System Management Facilities (SMF)	*OS/390 MVS System Management Facilities (SMF)*	GC28-1783
OS/390 V1R2.0 MVS Planning: Operations	*OS/390 MVS Planning: Operations*	GC28-1760
OS/390 V1R2.0 MVS Planning: Global Resource Serialization	*OS/390 MVS Planning: Global Resource Serialization*	GC28-1759
OS/390 V1R2.0 MVS System Data Set Definition	*OS/390 MVS System Data Set Definition*	GC28-1782

Short Title	Title	Order Number
OS/390 V1R2.0 MVS Device Validation Support	*OS/390 MVS Device Validation Support*	GC28-1748
OS/390 V1R2.0 MVS System Messages, Vol 1 (ABA-ASA)	*OS/390 MVS System Messages, Volume 1 (ABA-ASA)*	GC28-1784
OS/390 V1R2.0 MVS System Messages, Vol 2 (ASB-EWX)	*OS/390 MVS System Messages, Volume 2 (ASB-EWX)*	GC28-1785
OS/390 V1R2.0 MVS System Messages, Vol 3 (GDE-IEB)	*OS/390 MVS System Messages, Volume 3 (GDE-IEB)*	GC28-1786
OS/390 V1R2.0 MVS System Messages, Vol 4 (IEC-IFD)	*OS/390 MVS System Messages, Volume 4 (IEC-IFD)*	GC28-1787
OS/390 V1R2.0 MVS System Messages, Vol 5 (IGD-IZP)	*OS/390 MVS System Messages, Volume 5 (IGD-IZP)*	GC28-1788
OS/390 V1R2.0 MVS Dump Output Messages	*OS/390 MVS Dump Output Messages*	GC28-1749
OS/390 V1R2.0 MVS System Codes	*OS/390 MVS System Codes*	GC28-1780
OS/390 V1R2.0 MVS Routing and Descriptor Codes	*OS/390 MVS Routing and Descriptor Codes*	GC28-1778
OS/390 V1R2.0 MVS Recovery and Reconfiguration Guide	*OS/390 MVS Recovery and Reconfiguration Guide*	GC28-1777
OS/390 V1R2.0 MVS JES Common Coupling Services	*OS/390 MVS Programming: JES Common Coupling Services*	GC28-1770
OS/390 V1R2.0 MVS IPCS Commands Summary	*OS/390 MVS Interactive Problem Control System (IPCS) Commands Summary*	GX22-0039
MVS Batch Local Shared Resources	*MVS Programming: Batch Local Shared Resources Subsystem Guide*	GC28-1469
MVS Hiperbatch Guide	*MVS Programming: Hiperbatch Guide*	GC28-1470
OS/390 V1R2.0 MVS Writing Servers for APPC/MVS	*OS/390 MVS Programming: Writing Servers for APPC/MVS*	GC28-1774
OS/390 V1R2.0 MVS Writing Transaction Schedulers for APPC/MVS	*OS/390 MVS Programming: Writing Transaction Schedulers for APPC/MVS*	GC28-1776
OS/390 V1R2.0 MVS APPC/MVS Handbook for OS/2	*OS/390 MVS APPC/MVS Handbook for the OS/2 System Administrator*	GC28-1746
NJE Formats and Protocols	*Network Job Entry Formats and Protocols*	SC23-0070
OS/390 V1R2.0 MVS Product Registration	*OS/390 MVS Programming: Product Registration*	GC28-1729
OS/390 V1R2.0 MVS Product Management	*OS/390 MVS Product Management*	GC28-1730
OS/390 Product Management	*OS/390 MVS Programming: Product Management*	GC28-1730
OS/390 V1R2.0 MVS Diagnosis: Procedures	*OS/390 MVS Diagnosis: Procedures*	LY28-1082
OS/390 V1R2.0 MVS Diagnosis: Tools and Service Aids	*OS/390 MVS Diagnosis: Tools and Service Aids*	LY28-1085
OS/390 V1R2.0 MVS Diagnosis: Reference	*OS/390 MVS Diagnosis: Reference*	LY28-1084
OS/390 V1R2.0 MVS Planning: Workload Management	*OS/390 MVS Planning: Workload Management*	GC28-1761
OS/390 V1R2.0 MVS Workload Management Services	*OS/390 MVS Programming: Workload Management Services*	GC28-1773
OS/390 Information Roadmap	*OS/390 Information Roadmap*	GC28-1727

Short Title	Title	Order Number
OS/390 V1R2.0 MVS Diagnosis: Tools and Service Aids	*OS/390 MVS Diagnosis: Tools and Service Aids*	LY28-1845
OS/390 V1R2.0 MVS Diagnosis: Procedures	*OS/390 MVS Diagnosis: Procedures*	LY28-1844
OS/390 V1R2.0 MVS Diagnosis: Reference	*OS/390 MVS Diagnosis: Reference*	LY28-1872
OS/390 Introduction and Release Guide	*OS/390 Introduction and Release Guide*	GC28-1725
SystemView for MVS Up and Running!	*SystemView for MVS Up and Running!*	GC28-1241
OS/390 Up and Running!	*OS/390 Up and Running!*	GC28-1726
OS/390 Information Roadmap	*OS/390 Information Roadmap*	GC28-1727
OS/390 Summary of Message Changes	*OS/390 Summary of Message Changes*	GC28-1499

- SMP/E and Installation Manuals

Short Title	Title	Order Number
ServerPac Introduction to CustomPac Installation Dialogs	*MVS Custom-Built Offerings IBM ServerPac for OS/390 Introduction to CustomPac Installation Dialogs*	SC28-1244
MVS Packaging Rules	*Standard Packaging Rules for MVS-Based Products*	SC23-3695
OS/390 SMP/E Diagnosis Guide	*OS/390 System Modification Program Extended Diagnosis Guide*	SC28-1737
OS/390 SMP/E Messages and Codes	*OS/390 System Modification Program Extended Messages and Codes*	SC28-1738
OS/390 SMP/E User's Guide	*OS/390 System Modification Program Extended User's Guide*	SC28-1740
OS/390 SMP/E User's Guide	*OS/390 System Modification Program Extended User's Guide*	SC28-1740
OS/390 SMP/E Command Reference	*OS/390 System Modification Program Extended Command Reference*	SC28-1805
OS/390 SMP/E Reference	*OS/390 System Modification Program Extended Reference*	SC28-1806

- SOMobjects Environment

Short Title	Title	Order Number
OS/390 SOMobjects Reference Summary	*OS/390 SOMobjects Reference Summary*	SC28-1856
OS/390 SOMobjects Class Library Reference	*OS/390 SOMobjects Class Library Reference*	SC28-1858
OS/390 SOMobjects User's Guide	*OS/390 SOMobjects User's Guide*	GC28-1859
OS/390 SOMobjects Planning Guide	*OS/390 SOMobjects Planning Guide*	GC28-1857
OS/390 SOMobjects Service Classes Programmer's Guide	*OS/390 SOMobjects Service Classes Programmer's Guide*	GC28-1809
OS/390 SOMobjects Service Classes Class Library Reference	*OS/390 SOMobjects Service Classes Class Library Reference*	GC28-1810

- TCP/IP

Short Title	Title	Order Number
TCP/IP for MVS: Application Programming Interface Reference	*TCP/IP for MVS: Application Programming Interface Reference*	SC31-7187
TCP/IP for MVS: CICS TCP/IP Socket Interface Guide and Reference	*TCP/IP for MVS: CICS TCP/IP Socket Interface Guide and Reference*	SC31-7131
TCP/IP for MVS: Diagnosis Guide	*TCP/IP for MVS: Diagnosis Guide*	LY43-0105
TCP/IP Version 3 for OpenEdition MVS: Applications Feature Guide	*TCP/IP Version 3 for OpenEdition MVS: Applications Feature Guide*	SC31-8069
TCP/IP for MVS: IMS TCP/IP Application Development Guide and Reference	*TCP/IP for MVS: IMS TCP/IP Application Development Guide and Reference*	SC31-7186
TCP/IP for MVS: Messages and Codes	*TCP/IP for MVS: Messages and Codes*	SC31-7132
TCP/IP for MVS: Planning and Migration Guide	*TCP/IP for MVS: Planning and Migration Guide*	SC31-7189
TCP/IP for MVS: Offloading TCP/IP Processing	*TCP/IP for MVS: Offloading TCP/IP Processing*	SC31-7133
TCP/IP: Performance Tuning Guide	*TCP/IP: Performance Tuning Guide*	SC31-7188
TCP/IP for MVS: Customization and Administration Guide	*TCP/IP for MVS: Customization and Administration Guide*	SC31-7134
TCP/IP for MVS: Network Print Facility	*TCP/IP for MVS: Network Print Facility*	SC31-8074
TCP/IP for MVS: Programmer's Reference	*TCP/IP for MVS: Programmer's Reference*	SC31-7135
TCP/IP for MVS: User's Guide	*TCP/IP for MVS: User's Guide*	SC31-7136

- JES2 Subsystem

Short Title	Title	Order Number
OS/390 JES2 Messages	*OS/390 JES2 Messages*	GC28-1796
OS/390 JES2 Commands	*OS/390 JES2 Commands*	GC28-1790
OS/390 JES2 Initialization and Tuning Guide	*OS/390 JES2 Initialization and Tuning Guide*	SC28-1791
OS/390 JES2 Initialization and Tuning Reference	*OS/390 JES2 Initialization and Tuning Reference*	SC28-1792
OS/390 JES2 Diagnosis	*OS/390 JES2 Diagnosis*	LY28-1086
OS/390 JES2 Introduction	*OS/390 JES2 Introduction*	GC28-1794
OS/390 JES2 Program Reference	*OS/390 JES2 Program Reference*	LY28-1088
OS/390 JES2 Installation Exits	*OS/390 JES2 Installation Exits*	SC28-1793
OS/390 JES2 Macros	*OS/390 JES2 Macros*	SC28-1795
OS/390 JES2 Data Areas, Vol 1 $APT-$ERAPL	*OS/390 JES2 Data Areas, Volume 1 $APT-$ERAPL*	LY28-1096
OS/390 JES2 Data Areas, Vol 2 $FCLWORK-$OUTWORK	*OS/390 JES2 Data Areas, Volume 2 $FCLWORK-$OUTWORK*	LY28-1097
OS/390 JES2 Data Areas, Vol 3 $PADDR-$XRQ	*OS/390 JES2 Data Areas, Volume 3 $PADDR-$XRQ*	LY28-1098
OS/390 JES2 Migration Notebook	*OS/390 JES2 Migration Notebook*	GC28-1797

- JES3 Subsystem

Short Title	Title	Order Number
OS/390 JES3 Conversion Notebook	*OS/390 JES3 Conversion Notebook*	GC28-1799
OS/390 JES3 Diagnosis	*OS/390 JES3 Diagnosis*	LY28-1090
OS/390 JES3 Initialization and Tuning Guide	*OS/390 JES3 Initialization and Tuning Guide*	SC28-1802
OS/390 JES3 Introduction	*OS/390 JES3 Introduction*	GC28-1808
OS/390 JES3 Initialization and Tuning Reference	*OS/390 JES3 Initialization and Tuning Reference*	SC28-1803
OS/390 JES3 Customization	*OS/390 JES3 Customization*	LY28-1089
OS/390 JES3 Messages	*OS/390 JES3 Messages*	GC28-1804
OS/390 JES3 Commands	*OS/390 JES3 Commands*	GC28-1798
OS/390 JES3 Diagnosis Reference	*OS/390 JES3 Diagnosis Reference*	LY28-1092
OS/390 JES3 Data Areas, Volume 1	*OS/390 JES3 Data Areas, Volume 1*	LY28-1179
OS/390 JES3 Data Areas, Volume 2	*OS/390 JES3 Data Areas, Volume 2*	LY28-1180
OS/390 JES3 Data Areas, Volume 3	*OS/390 JES3 Data Areas, Volume 3*	LY28-1181
OS/390 JES3 Data Areas, Volume 4	*OS/390 JES3 Data Areas, Volume 4*	LY28-1182

- DFSORT

Short Title	Title	Order Number
DFSORT Getting Started	*Getting Started with DFSORT*	SC26-4109
DFSORT Installation and Customization	*DFSORT Installation and Customization*	SC33-4034
DFSORT Application Programming Guide	*DFSORT Application Programming Guide*	SC33-4035
DFSORT Messages, Codes and Diagnosis Guide	*DFSORT Messages, Codes and Diagnosis Guide*	SC26-7050
DFSORT Licensed Program Specifications	*DFSORT Licensed Program Specifications*	GC33-4032
DFSORT Tuning Guide	*DFSORT Tuning Guide*	SC26-3111
DFSORT Panels Guide	*DFSORT Panels Guide*	GC26-7037

- ICKDSF

Short Title	Title	Order Number
ICKDSF R16 Refresh, User's Guide	*Device Support Facilities User's Guide*	GC35-0033

- Storage Environment

Short Title	Title	Order Number
DFSMS/MVS V1R3 General Information	*DFSMS/MVS Version 1 Release 3 General Information*	GC26-4900
DFSMS/MVS V1R3 Licensed Program Specifications	*DFSMS/MVS Version 1 Release 3 Licensed Program Specifications*	GC26-4903

Short Title	Title	Order Number
DFSMS/MVS V1R3 Master Index	*DFSMS/MVS Version 1 Release 3 Master Index*	GC26-4904
DFSMS/MVS V1R3 Planning for Installation	*DFSMS/MVS Version 1 Release 3 Planning for Installation*	SC26-4919
DFSMS/MVS V1R3 Installation Exits	*DFSMS/MVS Version 1 Release 3 Installation Exits*	SC26-4908
DFSMS/MVS V1R3 Managing Catalogs	*DFSMS/MVS Version 1 Release 3 Managing Catalogs*	SC26-4914
DFSMS/MVS V1R1 Checkpoint/Restart	*DFSMS/MVS Version 1 Release 1 Checkpoint/Restart*	SC26-4907
DFSMS/MVS V1R3 Using Data Sets	*DFSMS/MVS Version 1 Release 3 Using Data Sets*	SC26-4922
DFSMS/MVS V1R3 Macro Instructions for Data Sets	*DFSMS/MVS Version 1 Release 3 Macro Instructions for Data Sets*	SC26-4913
DFSMS/MVS V1R3 DFSMSdfp Advanced Services	*DFSMS/MVS Version 1 Release 3 DFSMSdfp Advanced Services*	SC26-4921
DFSMS/MVS V1R3 Access Method Services for ICF	*DFSMS/MVS Version 1 Release 3 Access Method Services for the Integrated Catalog Facility*	SC26-4906
DFSMS/MVS V1R3 Access Method Services for VSAM	*DFSMS/MVS Version 1 Release 3 Access Method Services for VSAM Catalogs*	SC26-4905
DFSMS/MVS V1R3 DFSMSdfp Storage Administration Reference	*DFSMS/MVS Version 1 Release 3 DFSMSdfp Storage Administration Reference*	SC26-4920
DFSMS/MVS V1R3 Using ISMF	*DFSMS/MVS Version 1 Release 3 Using the Interactive Storage Management Facility*	SC26-4911
DFSMS/MVS V1R3 Utilities	*DFSMS/MVS Version 1 Release 3 Utilities*	SC26-4926
DFSMS/MVS V1R3 DFM/MVS Guide and Reference	*DFSMS/MVS Version 1 Release 3 Distributed FileManager/MVS Guide and Reference*	SC26-4915
DFSMS/MVS V1R3 Program Management	*DFSMS/MVS Version 1 Release 3 Program Management*	SC26-4916
DFSMS/MVS V1R3 DFSMSdfp Diagnosis Guide	*DFSMS/MVS Version 1 Release 3 DFSMSdfp Diagnosis Guide*	LY27-9605
DFSMS/MVS V1R3 DFSMSdfp Diagnosis Reference	*DFSMS/MVS Version 1 Release 3 DFSMSdfp Diagnosis Reference*	LY27-9606
DFSMS/MVS V1R2 Using Magnetic Tapes	*DFSMS/MVS Version 1 Release 2 Using Magnetic Tapes*	SC26-4923
DFSMS/MVS V1R2 Using the Volume Mount Analyzer	*DFSMS/MVS Version 1 Release 2 Using the Volume Mount Analyzer*	SC26-4925
DFSMS/MVS V1R3 Implementing System-Managed Storage	*DFSMS/MVS Version 1 Release 3 Implementing System-Managed Storage*	SC26-3123
MVS/ESA SML: Leading a Storage Administration Group	*MVS/ESA Storage Management Library: Leading a Storage Administration Group*	SC26-3126
MVS/ESA SML: Managing Storage Groups	*MVS/ESA Storage Management Library: Managing Storage Groups*	SC26-3125
MVS/ESA SML: Managing Data	*MVS/ESA Storage Management Library: Managing Data*	SC26-3124

Short Title	Title	Order Number
DFSMS/MVS V1R2 OAM Application Programmer's Reference	*DFSMS/MVS Version 1 Release 2 Object Access Method Application Programmer's Reference*	SC26-4917
DFSMS/MVS V1R3 OAM Planning, Installation, and Administration Guide for Object Support	*DFSMS/MVS Version 1 Release 3 Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*	SC26-4918
DFSMS/MVS V1R3 OAM Planning, Installation, and Storage Administration Guide for Tape Libraries	*DFSMS/MVS Version 1 Release 3 Object Access Method Planning, Installation, and Storage Administration for Tape Libraries*	SC26-3051
DFSMS/MVS V1R3 DFSMSrmm Guide and Reference	*DFSMS/MVS Version 1 Release 3 DFSMSrmm Guide and Reference*	SC26-4931
DFSMS/MVS V1R3 DFSMSrmm Implementation and Customization Guide	*DFSMS/MVS Version 1 Release 3 DFSMSrmm Implementation and Customization Guide*	SC26-4932
DFSMS/MVS V1R3 DFSMSrmm Diagnosis Guide	*DFSMS/MVS Version 1 Release 3 DFSMSrmm Diagnosis Guide*	LY27-9615
DFSMS/MVS V1R3 DFSMShsm Managing Your Own Data	*DFSMS/MVS Version 1 Release 3 DFSMShsm Managing Your Own Data*	SH21-1077
DFSMS/MVS V1R3 DFSMShsm Storage Administration Guide	*DFSMS/MVS Version 1 Release 3 DFSMShsm Storage Administration Guide*	SH21-1076
DFSMS/MVS V1R3 DFSMShsm Storage Administration Reference Summary	*DFSMS/MVS Version 1 Release 3 DFSMShsm Storage Administration Reference Summary*	SX26-3808
DFSMS/MVS V1R3 Managing Data Availability	*DFSMS/MVS Version 1 Release 3 Managing Data Availability*	SC26-4928
DFSMS/MVS V1R3 DFSMShsm Implementation and Customization Guide	*DFSMS/MVS Version 1 Release 3 DFSMShsm Implementation and Customization Guide*	SH21-1078
DFSMS/MVS V1R3 DFSMShsm Diagnosis Guide	*DFSMS/MVS Version 1 Release 3 DFSMShsm Diagnosis Guide*	LY27-9607
DFSMS/MVS V1R3 DFSMShsm Diagnosis Reference	*DFSMS/MVS Version 1 Release 3 DFSMShsm Diagnosis Reference*	LY27-9608
DFSMS/MVS V1R3 DFSMShsm Storage Administration Reference	*DFSMS/MVS Version 1 Release 3 DFSMShsm Storage Administration Reference*	SH21-1075
DFSMS/MVS V1R3 DFSMSdss Storage Administration Guide	*DFSMS/MVS Version 1 Release 3 DFSMSdss Storage Administration Guide*	SC26-4930
DFSMS/MVS V1R3 DFSMSdss Storage Administration Reference	*DFSMS/MVS Version 1 Release 3 DFSMSdss Storage Administration Reference*	SC26-4929
DFSMS/MVS V1R3 DFSMSdss Diagnosis Guide	*DFSMS/MVS Version 1 Release 3 DFSMSdss Diagnosis Guide*	LY27-9609
Stand-Alone Services Overview	*Data Facility Data Set Services Version 2 Release 5 and DFSMS/MVS Version 1 Stand-Alone Services Overview*	SC26-0185
Remote Copy Administrator's Guide	*Remote Copy Administrator's Guide*	SC35-0169
DFSMS/MVS Network File System User's Guide	*DFSMS/MVS Network File System User's Guide*	SC26-7028
DFSMS/MVS Network File System Customization and Operation	*DFSMS/MVS Network File System Customization and Operation*	SC26-7029
DFSMS/MVS V1R2 Network File System Performance Tuning Guide	*DFSMS/MVS Version 1 Release 2 Network File System Performance Tuning Guide*	SC26-7019

- Security Server

Short Title	Title	Order Number
OS/390 Security Server (RACF) General User's Guide	*OS/390 Security Server (RACF) General User's Guide*	SC28-1917
OS/390 Security Server (RACF) System Programmer's Guide	*OS/390 Security Server (RACF) System Programmer's Guide*	SC28-1913
OS/390 Security Server (RACF) Macros and Interfaces	*OS/390 Security Server (RACF) Macros and Interfaces*	SC28-1914
OS/390 Security Server (RACF) Command Language Reference	*OS/390 Security Server (RACF) Command Language Reference*	SC28-1919
OS/390 Security Server (RACF) General Information Manual	*OS/390 Security Server (RACF) General Information Manual*	GC28-1912
OS/390 Security Server (RACF) Messages and Codes	*OS/390 Security Server (RACF) Messages and Codes*	SC28-1918
OS/390 Security Server (RACF) Security Administrator's Guide	*OS/390 Security Server (RACF) Security Administrator's Guide*	SC28-1915
OS/390 Security Server (RACF) Auditor's Guide	*OS/390 Security Server (RACF) Auditor's Guide*	SC28-1916
OS/390 Security Server (RACF) Diagnosis Guide	*OS/390 Security Server (RACF) Diagnosis Guide*	LY27-2639
OS/390 Security Server (RACF) Data Areas	*OS/390 Security Server (RACF) Data Areas*	LY27-2640
OS/390 Security Server External Security Interface (RACROUTE) Macro Reference	*OS/390 Security Server External Security Interface (RACROUTE) Macro Reference*	GC28-1922
OS/390 Security Server (RACF) Callable Services	*OS/390 Security Server (RACF) Callable Services*	GC28-1921
OS/390 Security Server (RACF) Planning: Installation and Migration	*OS/390 Security Server (RACF) Planning: Installation and Migration*	GC28-1920
OS/390 Security Server (RACF) Support for: MVS OpenEdition DCE, SOMobjects for MVS, and SystemView forMVS	*OS/390 Security Server (RACF) Support for: MVS OpenEdition DCE, SOMobjects for MVS, and SystemView forMVS*	GC28-1924
OS/390 Security Server (OpenEdition DCE Security Server) Overview	*OS/390 Security Server (OpenEdition DCE Security Server) Overview*	GC28-1938

- EREP

Short Title	Title	Order Number
EREP V3R5 User's Guide	*Environmental Record Editing and Printing Program Version 3 Release 5 User's Guide*	GC35-0151
EREP V3R5 Reference	*Environmental Record Editing and Printing Program Version 3 Release 5 Reference*	GC35-0152

- TSO/E

Short Title	Title	Order Number
TSO/E Administration	*OS/390 TSO/E Administration*	SC28-1966
TSO/E REXX Reference	*OS/390 TSO/E REXX Reference*	SC28-1975
TSO/E General Information	*OS/390 TSO/E General Information*	GC28-1964
TSO/E Customization	*OS/390 TSO/E Customization*	SC28-1965
TSO/E Programming Guide	*OS/390 TSO/E Programming Guide*	SC28-1970
TSO/E Programming Services	*OS/390 TSO/E Programming Services*	SC28-1971
TSO/E CLISTs	*OS/390 TSO/E CLISTs*	SC28-1973
TSO/E Guide to SRPI	*OS/390 TSO/E Guide to the Server-Requester Programming Interface*	SC28-1976
TSO/E User's Guide	*OS/390 TSO/E User's Guide*	SC28-1968
TSO/E REXX User's Guide	*OS/390 TSO/E REXX User's Guide*	SC28-1974
TSO/E System Programming Command Reference	*OS/390 TSO/E System Programming Command Reference*	SC28-1972
TSO/E Command Reference	*OS/390 TSO/E Command Reference*	SC28-1969
TSO/E Diagnosis: Data Areas	*OS/390 TSO/E System Diagnosis: Data Areas*	LY28-1138
TSO/E Messages	*OS/390 TSO/E Messages*	GC28-1978
TSO/E Primer	*OS/390 TSO/E Primer*	GC28-1967

- Open System Adapter

Short Title	Title	Order Number
OSA Planning	*Planning for the System/390 Open Systems Adapter Feature*	GC23-3870
OSA/SF User's Guide	*OS/390 Open Systems Adapter Support Facility User's Guide*	SC28-1855
OSA/SF Messages and Codes	*OS/390 Open Systems Adapter Support Facility Messages and Codes*	SC28-1923

- OpenEdition DCE DFS

Short Title	Title	Order Number
DFS Administration Guide and Reference	*OS/390 OpenEdition DCE DFS Administration Guide and Reference*	SC28-1720
DFS Configuring and Getting Started	*OS/390 OpenEdition DCE DFS Configuring and Getting Started*	SC28-1722
Introducing OS/390 OE DCE DFS	*Introducing OS/390 OpenEdition DCE DFS*	SC28-1723
DFS Messages and Codes	*OS/390 OpenEdition DCE DFS Messages and Codes*	SC28-1724

- ISPF

Short Title	Title	Order Number
OS/390 V1R2.0 ISPF Planning and Customizing	*OS/390 V1R2.0 ISPF Planning and Customizing*	SC28-1298
OS/390 V1R2.0 ISPF Getting Started	*OS/390 V1R2.0 ISPF Getting Started*	SC28-1294
OS/390 V1R2.0 ISPF Dialog Tag Language Guide and Reference	*OS/390 V1R2.0 ISPF DTL Guide*	SC28-1219
OS/390 V1R2.0 ISPF User's Guide	*OS/390 V1R2.0 ISPF User's Guide*	SC28-1239
OS/390 V1R2.0 ISPF Dialog Developer's Guide and Reference	*OS/390 V1R2.0 ISPF Dialog Developer's Guide*	SC28-1273
OS/390 V1R2.0 ISPF Edit and Edit Macros	*OS/390 V1R2.0 ISPF Edit and Edit Macros*	SC28-1312
OS/390 V1R2.0 ISPF Examples	*OS/390 V1R2.0 ISPF Examples*	SC28-1282
OS/390 V1R2.0 ISPF Services Guide	*OS/390 V1R2.0 ISPF Services Guide*	SC28-1272
OS/390 V1R2.0 ISPF Library Management Facility	*OS/390 V1R2.0 ISPF Library Management Facility*	SC28-1317
OS/390 V1R2.0 ISPF Software Configuration and Library Manager Project Manager's Guide	*OS/390 V1R2.0 ISPF SCLM Project Manager's Guide*	SC28-1319
OS/390 V1R2.0 ISPF Software Configuration and Library Manager Developer's Guide	*OS/390 V1R2.0 ISPF SCLM Developer's Guide*	SC28-1318
OS/390 V1R2.0 ISPF Software Configuration and Library Manager Reference	*OS/390 V1R2.0 ISPF SCLM Reference*	SC28-1320
OS/390 V1R2.0 ISPF Messages and Codes	*OS/390 V1R2.0 ISPF Messages and Codes*	SC28-1326
OS/390 V1R2.0 ISPF Reference Summary	*OS/390 V1R2.0 ISPF Reference Summary*	SC28-1308

- VTAM

Short Title	Title	Order Number
VTAM Installation and Migration Guide	*VTAM Installation and Migration Guide*	GC31-6547
VTAM Release Guide	*VTAM Release Guide*	GC31-6555
VTAM Network Implementation Guide	*VTAM Network Implementation Guide*	SC31-6548
VTAM Resource Definition Reference	*VTAM Resource Definition Reference*	SC31-6552
VTAM Resource Definition Samples	*VTAM Resource Definition Samples*	SC31-6554
VTAM Customization	*VTAM Customization*	LY43-0068
VTAM Operation	*VTAM Operation*	SC31-6549
VTAM Messages and Codes	*VTAM Messages and Codes*	SC31-6546
VTAM Programming	*VTAM Programming*	SC31-6550
VTAM Programming for LU 6.2	*VTAM Programming for LU 6.2*	SC31-6551
VTAM CMIP Services and Topology Agent Programming Guide	*VTAM CMIP Services and Topology Agent Programming Guide*	SC31-6544
VTAM Diagnosis	*VTAM Diagnosis*	LY43-0069
VTAM Data Areas for MVS/ESA	*VTAM Data Areas for MVS/ESA*	LY43-0071

Short Title	Title	Order Number
APPC Application Suite User's Guide	*APPC Application Suite User's Guide*	SC31-6532
APPC Application Suite Administration	*APPC Application Suite Administration*	SC31-6533
APPC Application Suite Programming	*APPC Application Suite Programming*	SC31-6534
VTAM AnyNet Feature: Guide to Sockets over SNA	*VTAM AnyNet Feature: Guide to Sockets over SNA*	SC31-6559
VTAM AnyNet Feature: Guide to SNA over TCP/IP	*VTAM AnyNet Feature: Guide to SNA over TCP/IP*	SC31-6560
VTAM Glossary	*VTAM Glossary*	GC31-6556
Planning for NetView, NCP, and VTAM		SC31-8063
Planning for Integrated Networks		SC31-8062
VTAM Licensed Program Specifications		GC31-6553
VTAM Program Directory		

- Language Environment

Short Title	Title	Order Number
OS/390 Language Environment Debugging Guide and Run-Time Messages	*OS/390 Language Environment Debugging Guide and Run-Time Messages*	SC28-1942
OS/390 Language Environment Programming Guide	*OS/390 Language Environment Programming Guide*	SC28-1939
OS/390 Language Environment Programming Reference	*OS/390 Language Environment Programming Reference*	SC28-1940
OS/390 Language Environment Writing Interlanguage Communication Applications	*OS/390 Language Environment Writing Interlanguage Communication Applications*	SC28-1943
OS/390 Language Environment Installation and Customization	*OS/390 Language Environment Installation and Customization*	SC28-1941
OS/390 Language Environment Run-Time Migration Guide	*OS/390 Language Environment Run-Time Migration Guide*	SC28-1944
OS/390 Language Environment Concepts Guide	*OS/390 Language Environment Concepts Guide*	GC28-1945

- Other Products

Short Title	Title	Order Number
CICS Family: General Information	*CICS Family: General Information*	GC33-0155
CICS for MVS/ESA Migration Guide	*CICS for MVS/ESA Migration Guide*	GC33-1571
CICS for MVS/ESA Installation Guide	*CICS for MVS/ESA Installation Guide*	SC33-1681
CICS for MVS/ESA System Definition Guide	*CICS for MVS/ESA System Definition Guide*	SC33-1682
CICS for MVS/ESA Customization Guide	*CICS for MVS/ESA Customization Guide*	SC33-1683
CICS for MVS/ESA Resource Definition Guide	*CICS for MVS/ESA Resource Definition Guide*	SC33-1684
DB2 Installation Guide	*DB2 for MVS/ESA Installation Guide*	SC26-3456

Short Title	Title	Order Number
DB2 Admin Guide	*DB2 for MVS/ESA Administration Guide*	SC26-3265
IMS/ESA Installation Volume 1	*IMS/ESA Installation Volume 1 Installation and Verification*	SC26-8023
IMS/ESA Installation Volume 2	*IMS/ESA Installation Volume 2 System Definition and Tailoring*	SC26-8024
IMS/ESA Customization Guide	*IMS/ESA Customization Guide*	SC26-8020
IMS/ESA Data Base	*IMS/ESA System Administration Guide: Data Base*	SC26-8012
IMS/ESA Transaction Manager	*IMS/ESA System Administration Guide: Transaction Manager*	SC26-8014
IMS/ESA System	*IMS/ESA System Administration Guide: System*	SC26-8013
NCP, SSP, EP Generation and Loading Guide	*NCP, SSP, EP Generation and Loading Guide*	SC31-6221

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **ITSO4USA category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269	
Canada	1-403-267-4455	
Outside North America	(+45) 48 14 2207	(long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number _____
- Credit card number _____

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Index

E

F

G

H

I

J

L

M

N

O

P

R

S

T

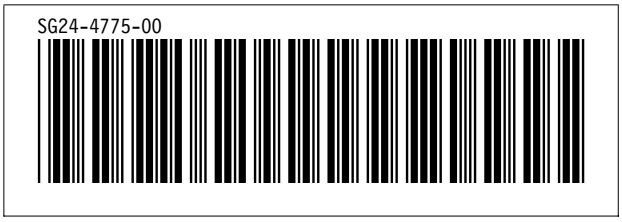
U

V

W

Z

Printed in U.S.A.



Artwork Definitions

Table Definitions

Figures

		141	47	140
R5IPSYS	R5IPROP	146	48	146
R5IPSPI	R5IPROP	147	49	147
R5HRGDA	R5H	151	50	151
R5HMES1	R5H	154	51	153, 160
R5HMES2	R5H	155	52	155, 160
R5HLVL3	R5H	156	53	156
R5HOES1	R5H	158	54	157, 158, 159
R5HCIC1	R5HCICS	162	55	161
R5HCIC2	R5HCICS	163	56	163
R5HCX3	R5HCICS	164	57	163
R5HDB21	R5HDB2	167	58	166, 167
R5HDB22	R5HDB2	169	59	168
R5HDB3	R5HDB2	170	60	169
R5HDB24	R5HDB2	171	61	170
R5HDB25	R5HDB2	172	62	171
R5HDB26	R5HDB2	173	63	172
R5HIMS1	R5HIMS	177	64	177
R5HIMS2	R5HIMS	178	65	178
R5HIM3	R5HIMS	179	66	178
R5HNCP1	R5HNCP	183	67	183
R5HNCP2	R5HNCP	184	68	184
R5HNC3	R5HNCP	185	69	184
R5HCLON	R5HCLONE	187	70	186
R5FLOW1	R5XFLOW	201	71	201

Headings

id	File	Page	References
R5A	R5A	1	Chapter 1, Introduction vii
R5AWHY1	R5A	1	1.1, Why Read This Book?
R5AWISS	R5A	1	1.2, Issues 8
R5ASMDF	R5A	2	1.3, Software Management Defined 7, 9
R50S390	R5A	4	1.4, What Makes Up OS/390? 102
R5EUP4	R5A	4	1.4.1, OS/390 Elements and Features
R5ASDEL	R5A	5	1.5, OS/390 Software Delivery Offerings 79
R5ARGDE	R5A	6	1.6, Reader′s Guide
R5ALIM	R5A	7	1.7, Some Limitations of This Book
R5ASSU	R5A	8	1.8, Assumptions
R5ACONC	R5A	8	1.9, Read On
R5C	R5C	9	Chapter 2, Maintenance Philosophy vii, 3, 89
R5CWHY	R5CMNT	10	2.1, Why Manage Software?
R5CWHYA	R5CMNT	10	2.1.1, Management Cycle
R5CWHYR	R5CMNT	10	2.1.2, The Need for Change 17
R5CCURR	R5CMNT	12	2.2, How Current Should I Be?
R5CRSK2	R5CMNT	13	2.2.1, The Risk of Not Changing
R5CRSK1	R5CMNT	13	2.2.2, The Risk of Changing
R5CRSK3	R5CMNT	15	2.2.3, The Minimum Risk Point
R5CRSK4	R5CMNT	15	2.2.3.1, No Risk
R5CRSK5	R5CMNT	15	2.2.3.2, What If Everyone Stays Behind?
R5CRSK6	R5CMNT	17	2.2.3.3, Minimum Risk Point Determination
R5CSTRA	R5CMNT	17	2.3, Approach for Keeping Current 10, 12, 16, 76, 90
R5CSTR1	R5CMNT	17	2.3.1, Today′s Strategy
R5CRATE	R5CMNT	18	2.3.2, Rate of Change
R5CDYNA	R5CMNT	19	2.3.3, Dynamic Versus Static
R5EM	R5CMGNT	19	2.4, Installation Strategy
R5ESM	R5CMGSW	20	2.4.1, How Should I Manage My Software?
R5EUOR	R5CMGSW	21	2.5, Upgrade or Replace? 17, 19
R5EREPM	R5CREPM	23	2.5.1, Replacement Methods 5, 75, 76
R5CSPAC	R5CREPM	24	2.5.1.2, SystemPac 25
R5EREPT	R5CREPM	25	2.5.2, Replacement Tasks
R5EUPGM	R5CUPGM	26	2.5.3, Upgrade Methods 5, 75
R5COPT	R5CUPGM	28	2.5.3.5, Optional Feature Activation 24

Index Entries

Revisions

id	File	Page	References
RC13102	?	?	? 197, 197
RC14102	?	?	? 197, 198

Tables

id	File	Page	References
R5UPG	R5CMGSW	23	1 23
R5ERT	R5CREPM	25	2 25
R5UPG1	R5CUPGM	32	3 32
R5PLAT	R5CIS	35	4 35
R5ETU1	R5EUPGS	95	5 94
DCEPROC	R5EDCE	119	6 118

Processing Options

Runtime values:

Document fileid	SG244775 SCRIPT
Document type	USERDOC
Document style	REDBOOK
Profile	EDFPRF40
Service Level	0022
SCRIPT/VS Release	4.0.0
Date	96.10.30
Time	20:35:28
Device	3820A
Number of Passes	3
Index	YES
SYSVAR D	Y
SYSVAR G	INLINE
SYSVAR H	Y
SYSVAR P	DRAFTM
SYSVAR S	O
SYSVAR T	R
SYSVAR X	Y

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	YES
DVCF conditions file	(none)
DVCF value 1	(none)
DVCF value 2	(none)
DVCF value 3	(none)
DVCF value 4	(none)
DVCF value 5	(none)
DVCF value 6	(none)
DVCF value 7	(none)
DVCF value 8	(none)
DVCF value 9	(none)
Explode	NO
Figure list on new page	YES
Figure/table number separation	YES
Folio-by-chapter	NO
Head 0 body text	Part

Head 1 body text	Chapter
Head 1 appendix text	Appendix
Hyphenation	NO
Justification	NO
Language	ENGL
Keyboard	395
Layout	OFF
Leader dots	YES
Master index	(none)
Partial TOC (maximum level)	4
Partial TOC (new page after)	INLINE
Print example id's	NO
Print cross reference page numbers	YES
Process value	DRAFTM
Punctuation move characters	.,
Read cross-reference file	(none)
Running heading/footing rule	NONE
Show index entries	NO
Table of Contents (maximum level)	3
Table list on new page	YES
Title page (draft) alignment	RIGHT
Write cross-reference file	(none)

Imbed Trace

Page 0	O390SU
Page 0	O390VARS
Page 0	REDB$POK
Page 0	OS3902CT
Page 0	ADM99L03
Page 0	APS99L03
Page 0	BDT99L03
Page 0	BKM99L03
Page 0	BPX99L03
Page 0	CBD99L03
Page 0	CBC99L03
Page 0	DFH99L03
Page 0	ERB99L03
Page 0	EUV99L03
Page 0	EWX99L03
Page 0	E0S99L03
Page 0	E0Z99L03
Page 0	GIM99L03
Page 0	GOT99L03
Page 0	HAL99L03
Page 0	HAS99L03
Page 0	IAT99L03
Page 0	ICE99L03
Page 0	ICK99L03
Page 0	IDA99L03
Page 0	ICH99L03
Page 0	IEA99L03
Page 0	IES99L03
Page 0	IFC99L03
Page 0	IKJ99L03
Page 0	IOA99L03
Page 0	IOE99L03
Page 0	ISP99L03
Page 0	IST99L03
Page 0	LEN99L03
Page 0	SOM99L03
Page i	REDB$ED1
Page i	O390EDNO
Page i	REDB$ED2
Page vii	O390ABST
Page vii	O390ORG
Page vii	O390ACKS
Page viii	REDB$COM
Page viii	O390MAIN
Page viii	O390IND
Page viii	R5A
Page 8	R5C
Page 9	R5CMNT
Page 19	R5CMGNT
Page 20	R5CMGSW
Page 23	R5CREPM
Page 25	R5CUPGM
Page 33	R5CIS
Page 35	R5CCONC
Page 37	R5D
Page 43	R5DLOGAR