

A group of people are gathered around a table, smiling and stacking their hands in a circle in the center. The scene is overlaid with a semi-transparent blue filter. In the top left corner, there are two overlapping circles, one solid purple and one with purple and white horizontal stripes. In the bottom right corner, there is a large white circle containing the 'GL & GROUP' logo.

Elementos básicos y características opcionales de z/OS

Elementos básicos y características opcionales de z/OS

El sistema operativo z/OS consta de elementos básicos y características opcionales.

Los elementos básicos (o simplemente elementos) proporcionan funciones esenciales del sistema operativo.

El programa de control básico (BCP):

Proporciona servicios esenciales del sistema operativo.

El BCP incluye el programa de configuración de E/S (IOCP), el administrador de carga de trabajo (WLM), las instalaciones de administración del sistema (SMF), el núcleo de servicios del sistema z/OS UNIX, el enlazador de administración de programas y otros componentes.

Modelo de información común (CIM):

El CIM es un modelo de datos estándar para describir y acceder a datos de administración de sistemas en entornos heterogéneos.

Permite a los administradores de sistemas escribir aplicaciones que midan los recursos del sistema en una red con diferentes sistemas operativos y hardware. Servidor de comunicaciones

El servidor de comunicaciones:

Los servicios criptográficos proporcionan las siguientes funciones, básicamente: secreto de datos, integridad de datos, identificación personal, firmas digitales y administración de claves criptográficas.

DFSMSdfp:

Proporciona funciones de administración de dispositivos, programas, datos y almacenamiento.

Servicio de archivos distribuidos:

El servicio de archivos distribuidos proporciona, el soporte de servicio de archivos (el cliente y servidor DFS) se encuentra en el nivel OSF 1.2.2. – El sistema de archivos zSeries (zFS). El zFS es un sistema de archivos UNIX jerárquico (HFS).

Los sistemas de archivos zFS contienen archivos y directorios a los que se puede acceder con las API de z/OS file system jerárquico.

Los sistemas de archivos zFS se pueden montar en la jerarquía z/OS UNIX junto con otros tipos de sistemas de archivos locales (o remotos) (como TFS, AUTOMNT y NFS).

Definición de configuración de hardware (HCD)

HCD define tanto la configuración del sistema operativo como la configuración del hardware del procesador para un sistema.

HTTP Server

Es el servidor web

Proporciona un servicio web escalable y de alto rendimiento para aplicaciones críticas de comercio electrónico.

Admite conexiones seguras de capa de sockets seguros (SSL), almacenamiento en caché dinámico mediante el acelerador de caché de respuesta rápida, múltiples direcciones IP, autenticación de proxy y caracteres de doble byte.



Servicios de seguridad integrados

Los servicios de seguridad integrados proporcionan funciones de seguridad básicas para z/OS.

Sus componentes incluyen:

Asignación de identidad empresarial (EIM), que permite asignar la identidad de un usuario en un sistema a la identidad del usuario en otro sistema.

Servicio de autenticación de red, que utiliza el algoritmo DES para el cifrado.

Instalación de productividad del sistema interactivo (ISPF)

ISPF proporciona aspectos del desarrollo de software basado en host.

ISPF tiene cuatro componentes principales: Administrador de diálogos, Instalación de desarrollo de programas, Administrador de bibliotecas y configuración de software y el componente Cliente/Servidor.

Subsistema de entrada de trabajos (JES)

Las instalaciones de z/OS pueden utilizar uno de los dos subsistemas de entrada de trabajos; un subsistema de entrada de trabajos acepta el envío de trabajo para el BCP. – JES2 ejerce un control independiente sobre sus funciones de procesamiento de trabajos.

– JES3 ejerce un control centralizado. JES2 es un elemento base de z/OS; JES3 es una característica opcional.

Language Environment

Language Environment proporciona el entorno de ejecución para programas generados con C, C++, COBOL, Fortran y PL/I.

Network File System (NFS)

NFS actúa como un servidor de archivos para estaciones de trabajo, computadoras personales u otros sistemas autorizados en una red TCP/IP.

System Modification Program Extended (SMP/E)

SMP/E es una herramienta para instalar y mantener software, y para administrar el inventario de software que se ha instalado.

Time Sharing Option/Extensions (TSO/E)

TSO/E permite a los usuarios crear una sesión interactiva con el sistema z/OS.

TSO proporciona una capacidad de inicio de sesión de usuario único y una interfaz de línea de comandos básica para z/OS.

z/OS UNIX System Services (z/OS UNIX)

z/OS UNIX proporciona la interfaz de comandos estándar familiar para los usuarios interactivos de UNIX.

Las características opcionales (o simplemente características) se pueden solicitar con z/OS y proporcionan funciones adicionales del sistema operativo.

Las características opcionales incluyen:

DFSMSdss

DFSMSdss copia y mueve datos para realizar copias de seguridad y recuperación, y para reducir la fragmentación del espacio libre.

DFSMSHsm

DFSMSHsm proporciona administración automatizada del almacenamiento DASD, incluida la administración del espacio para datos inactivos y con poca cantidad de datos, y la administración de la disponibilidad para la pérdida accidental de datos causada por desastres locales y del sitio. DFSMSHsm también le permite hacer un uso eficaz de los medios de cinta.

z/OS UNIX System Services (z/OS UNIX)

DFSORT

DFSORT permite ordenar, fusionar, copiar, generar informes y analizar de forma rápida y sencilla la información de su empresa, así como un manejo versátil de datos a nivel de registro, campo y bit.

Infoprint Server

Infoprint Server le permite imprimir archivos en impresoras z/OS desde cualquier estación de trabajo que tenga acceso TCP/IP.



z/OS UNIX System Services (z/OS UNIX)

Resource Measurement Facility (RMF)

RMF recopila datos sobre el uso de recursos z/OS y proporciona informes en cualquier sistema en un sysplex.

System Display and Search Facility (SDSF)

SDSF le proporciona información para supervisar, gestionar y controlar su sistema z/OS.



z/OS UNIX System Services (z/OS UNIX)

DFSORT

DFSORT permite ordenar, fusionar, copiar, generar informes y analizar de forma rápida y sencilla la información de su empresa, así como un manejo versátil de datos a nivel de registro, campo y bit.

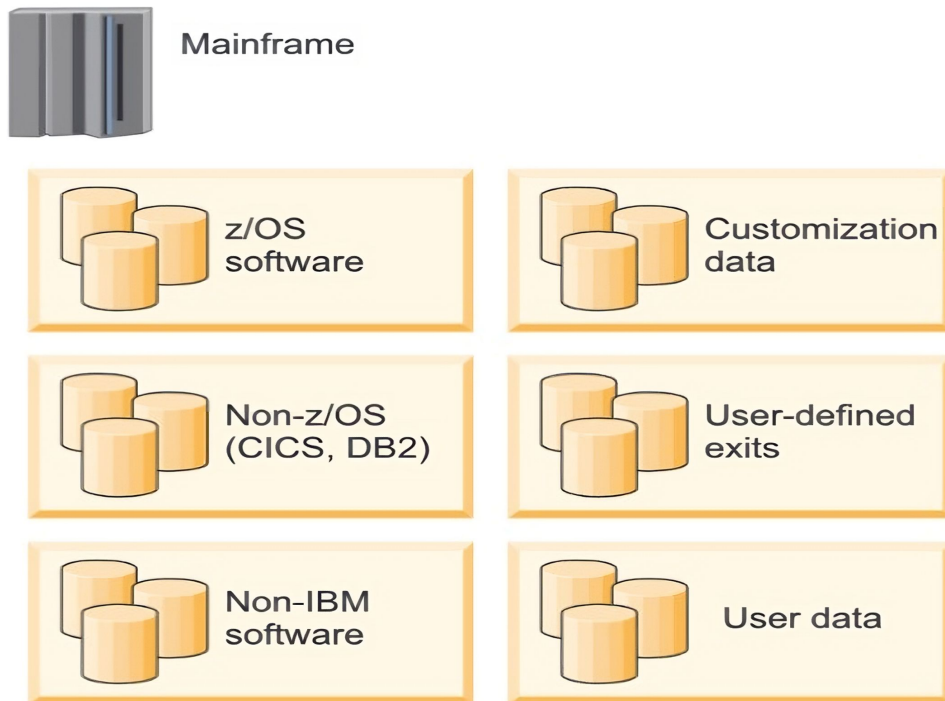
Infoprint Server

Infoprint Server le permite imprimir archivos en impresoras z/OS desde cualquier estación de trabajo que tenga acceso TCP/IP.



Organización básica del software en el Storage

La siguiente figura, ilustra los diferentes tipos de código y datos de software que existen en un sistema y cómo se organizan generalmente en el almacenamiento del sistema:



Organización básica del software en el Storage

El software z/OS, tal como lo suministra IBM, se instala normalmente en una serie de volúmenes de disco conocidos como volúmenes de residencia del sistema (SYSRES). Gran parte de la flexibilidad de z/OS se basa en estos conjuntos SYSRES. Permiten aplicar mantenimiento a un nuevo conjunto que se clona a partir del conjunto de producción mientras el conjunto actual está ejecutando trabajos de producción.

Luego, se puede realizar una breve interrupción para realizar la carga inicial de programa (IPL) desde el nuevo conjunto y se habrá implementado el mantenimiento. Además, el cambio se puede deshacer haciendo la carga inicial de programa (IPL) desde el conjunto anterior. Las correcciones a z/OS se gestionan con un producto llamado SMP/E.

Se utiliza la catalogación indirecta mediante símbolos del sistema para que una biblioteca en particular se catalogue como perteneciente, por ejemplo, al volumen 2 de SYSRES, y el sistema resuelva el nombre de ese volumen en el momento de la carga inicial de programa a partir de los símbolos del sistema.

Organización básica del software en el Storage

Otro software de IBM (como CICS; DB2; IMS, etc.) y productos de software que no son de IBM o de terceros proveedores normalmente se instalan en otro grupo de volúmenes, en lugar de en los volúmenes SYSRES. Los conjuntos SYSRES normalmente son administrados como una entidad por SMP/E, por lo que su contenido suele estar limitado al software z/OS.

El software que no es z/OS se instala en tantos volúmenes como sean necesarios y, por lo tanto, se puede administrar por separado.

Otro grupo de volúmenes está reservado para los datos de personalización, que se refieren a datos como las bibliotecas del sistema z/OS (SYS1.PARMLIB y SYS1.PROCLIB, por ejemplo); el catálogo maestro; el archivo de definición de E/S (IODF); conjuntos de datos de página; spools del subsistema de entrada de trabajos (JES); el directorio /etc ; y otros elementos que son esenciales para el funcionamiento del sistema z/OS.

Organización básica del software en el Storage

También es donde se almacenan los datos de SMP/E para administrar el software. Estos conjuntos de datos no siempre se encuentran en volúmenes DASD separados del software z/OS suministrado por IBM; algunas instalaciones colocan PARMLIB y PROCLIB en el primer SYSRES, otras los colocan en el pack (o volumen) del catálogo maestro o en otro lugar.

Esto es una cuestión de elección y depende de cómo se administren los volúmenes SYSRES. Cada instalación tendrá un método preferido.

En muchos sistemas, algunos de los valores predeterminados proporcionados por IBM no son apropiados, por lo que es necesario modificarlos.

Organización básica del software en el Storage

Las salidas de usuario y las modificaciones de usuario (usermods) se realizan en el código IBM para que se comporte como lo requiere la instalación.

Las modificaciones se administran generalmente mediante SMP/E.

Por último, otro conjunto de volúmenes contiene datos de producción, prueba y usuario; este conjunto es generalmente el grupo más grande de volúmenes de disco.

Este conjunto de volúmenes no es parte de las bibliotecas del sistema, pero se presenta aquí para completar.

A menudo se divide en grupos y se administra mediante el almacenamiento administrado por el sistema (SMS), que puede dirigir los datos a volúmenes administrados de manera adecuada.

Por ejemplo, los datos de producción se pueden colocar en volúmenes que se respaldan diariamente, mientras que los datos del usuario solo se pueden capturar semanalmente y se pueden migrar a cinta después de un breve período de inactividad para liberar espacio en el disco, para más datos.

z/OS system Adaptacion (customization)

La personalización del sistema (también conocida como tailoring) es el proceso general mediante el cual una instalación adapta su sistema operativo.

Los programadores de sistemas planifican y completan minuciosamente los pasos de este proceso, seleccionando las opciones del sistema a través de varios mecanismos diferentes.

Lográndose por ejemplo, a través de los siguientes mecanismos:

z/OS system Adaptacion (customization)

Definición de configuración de hardware (HCD)

Los programadores del sistema utilizan el diálogo de HCD para realizar una variedad de tareas, incluida la definición del sistema operativo y las configuraciones de hardware, la activación de la configuración (es decir, la aplicación de cambios de configuración al sistema) y la consulta o impresión de los datos de configuración.

Initialization-time selections

Selecciones en el momento de la inicialización.

Al inicializar el sistema operativo, los programadores del sistema adaptan el entorno del sistema a través de varias fuentes, incluidas las acciones del operador, los datos de personalización en las bibliotecas del sistema (SYS1.PARMLIB y otros conjuntos de datos de parmlib) y el lenguaje de control de trabajos (JCL) para el subsistema del Master Scheduler.

z/OS system Adaptacion (customization)

Parámetros implícitos del sistema

Varios requisitos del sistema afectan la forma en que funciona el sistema.

Estos requisitos del sistema pueden considerarse como parámetros "implícitos". Implican sentencias DD, data sets, opciones de hardware, etc.

Algunos ejemplos (entre otros) son:

- Sentencias DD SYSABEND, SYSMDUMP y SYSUDUMP.
- Los data sets de System Management Facilities (SMF).
- Adición de nuevos módulos a la concatenación LPAALST.
- Elección del dispositivo en el que residirán los conjuntos de datos de paginación PLPA.
- Definición de page data sets, mediante el comando DEFINE PAGESPACE.

z/OS system libraries

SYS1.PARMLIB

Es una library obligatoria que contiene parámetros de control para todo el sistema. PARMLIB es un conjunto de datos importante en un sistema operativo z/OS y se puede considerar que realiza una función similar a: /etc en un sistema UNIX.

El propósito de PARMLIB es proporcionar muchos parámetros de control en un formato preespecificado en un único conjunto de datos, minimizando así la necesidad de que el operador ingrese parámetros durante el proceso de inicialización (IPL).

SYS1.PARMLIB debe residir en un volumen de acceso directo, que puede ser el volumen de residencia del sistema.

Los parámetros se especifican en miembros proporcionados por IBM y creados por la instalación.

Todos los parámetros y miembros del conjunto de datos SYS1.PARMLIB se describen en: z/OS MVS Initialization and Tuning Reference.

z/OS system libraries

SYS1.LINKLIB

Contiene gran parte del código ejecutable, también conocido como módulos, para los componentes y utilidades de z/OS.

De manera predeterminada, SYS1.LINKLIB es el primer data set en la Lnklist, que es una colección de bibliotecas que contienen código de sistema y de usuario.

Uno de los módulos en SYS1.LINKLIB es MSTJCL00, que contiene las declaraciones iniciales del lenguaje de control de trabajos (JCL) que inician el subsistema del master scheduler.

El master scheduler establece comunicación entre el sistema operativo y el subsistema de entrada de trabajos principal, que es normalmente es JES2 (o JES3).



z/OS system libraries

Link Pack Area (LPA)

La LPA es parte del almacenamiento del ‘área común’ de un address space, y se divide en secciones paginables, fijas y modificadas:

Las bibliotecas especificadas en los miembros SYS1.LPALIB, LPALSTxx o PROGxx de parmlib, se cargan en el LPA paginable (PLPA).

Estas bibliotecas contienen módulos para programas del sistema de solo lectura, junto con cualquier programa de usuario de solo lectura re-entrante seleccionado por una instalación que se puede compartir entre los usuarios del sistema.

Los miembros IEAFIXxx especifican los módulos cargados en el LPA fijo (FLPA). Esta área se debe utilizar solo para módulos que aumentan significativamente el rendimiento cuando son fijos en lugar de paginables.

Los mejores candidatos para FLPA son los módulos que se utilizan con poca frecuencia, pero que son necesarios para una respuesta rápida.

z/OS system libraries

SYS1.PROCLIB

SYS1.PROCLIB es una librería obligatoria, que contiene los procedimientos JCL suministrados por IBM que se utilizan para realizar determinadas funciones del sistema. El JCL puede ser para tareas del sistema o para procesar tareas de programa invocadas por el operador o el programador.

Un procedimiento JCL en SYS1.PROCLIB es el procedimiento catalogado JES2, que define las librerías de procedimientos relacionados con los trabajos (job-related).

SYS1.NUCLEUS

Contiene los módulos supervisores básicos del sistema.

Símbolos del sistema (system symbols) en PARMLIB

Los símbolos del sistema son elementos que permiten que diferentes sistemas z/OS compartan definiciones PARMLIB y al mismo tiempo conserven valores únicos en esas definiciones.

Los símbolos del sistema actúan como variables en un programa; pueden tomar valores diferentes, basados según el 'input' al programa.

Cuando especifica un símbolo del sistema en una definición PARMLIB compartida, el símbolo del sistema actúa como un “marcador de posición” (placeholder).

Cada sistema que comparte la definición reemplaza el símbolo del sistema con un valor único durante la inicialización.

Cada símbolo del sistema tiene un nombre, que comienza con un ampersand (&) y opcionalmente termina con un punto (.), y tiene un texto de sustitución, que es la cadena de caracteres (string) que el sistema sustituye por un símbolo cada vez que aparece.



Símbolos del sistema (system symbols) en PARMLIB

Hay dos tipos de símbolos del sistema:

Dinámico

El texto de sustitución puede cambiar en cualquier momento durante una IPL.

Estático

El texto de sustitución se define en la inicialización del sistema y permanece fijo durante la vida del IPL.

Algunos símbolos están reservados para uso del sistema.



Símbolos del sistema (system symbols) en PARMLIB

Se puede visualizar los símbolos en el sistema ingresando el comando: **D SYMBOLS**.

Ejemplo:

```
----- SDSF OPTION MENU -  
COMMAND INPUT = /D SYMBOLS  
IEA007I STATIC SYSTEM SYMBOL VALUES  
&SYSALVL. = "2"  
&SYSCLONE. = "70"  
&SYSNAME. = "SC70"  
&SYSPLEX. = "SANDBOX"  
&SYSR1. = "Z17RC1"  
&ALLCLST1. = "CANCEL"  
&CMDLIST1. = "70,00"  
&COMMDSN1. = "COMMON"  
&DB2. = "V8"  
&DCEPROC1. = "."  
&DFHSMCMD. = "00"  
&DFHSMHST. = "6"  
&DFHSPRI. = "NO"  
&DFSPROC1. = "."  
&DLIB1. = "Z17DL1"  
&DLIB2. = "Z17DL2"  
&DLIB3. = "Z17DL3"  
&DLIB4. = "Z17DL4"  
&IEFSSNXX. = "R7"  
&IFAPRDX. = "4a"
```

Símbolos del sistema (system symbols) en PARMLIB

El miembro de PARMLIB, IEASYMxx proporciona un único lugar para especificar parámetros del sistema para cada sistema en un entorno multisistema.

IEASYMxx contiene instrucciones (statements) que definen símbolos de sistema estáticos y que especifican miembros PARMLIB de IEASYSxx que contienen parámetros del sistema (statement SYSPARM).

Ejemplo de un miembro IEASYMxx de PARMLIB:

```
SYSDEF      SYSCONE(&SYSNAME(3:2))
SYMDEF(&SYSR2='&SYSR1(1:5).2')
SYMDEF(&SYSR3='&SYSR1(1:5).3')
SYMDEF(&DLIB1='&SYSR1(1:3).DL1')
SYMDEF(&DLIB2='&SYSR1(1:3).DL2')
SYMDEF(&DLIB3='&SYSR1(1:3).DL3')
SYMDEF(&DLIB4='&SYSR1(1:3).DL4')
SYMDEF(&ALLCLST1='CANCEL')
SYMDEF(&CMDLIST1='&SYSCONE.,00')
SYMDEF(&COMMDSN1='COMMON')
SYMDEF(&DFHSMCMD='00')
SYMDEF(&IFAPRDXX='00')
SYMDEF(&DCEPROC1='.')
SYMDEF(&DFSPROC1='.')
SYSDEF      HWNAME(SCZP901)
LPARNAME(A13)
SYSNAME(SC70)
SYSPARM(R3,70)
SYMDEF(&IFAPRDXX='4A')
SYMDEF(&DFHSMHST='6')
SYMDEF(&DFHSPRI='NO')
SYMDEF(&DB2='V8')
```

Símbolos del sistema (system symbols) en PARMLIB

En este ejemplo, la variable &SYSNAME tendrá el valor especificado por la palabra clave SYSNAME; en este caso, SC70.

Debido a que cada sistema en un sysplex tiene un nombre único, puede utilizar &SYSNAME en la especificación de recursos únicos del sistema.

Como ejemplo, puede especificar el nombre de un conjunto de datos SMF como SYS1.&SYSNAME..MAN1, con la sustitución que da como resultado el nombre SYS1.SC70.MAN1 cuando se ejecuta en SC70.

Se puede utilizar variables para construir los valores de otras variables.

```
SYSDEF      SYSCONE(&SYSNAME(3:2))
SYMDEF(&SYSR2=&SYSR1(1:5).2')
SYMDEF(&SYSR3=&SYSR1(1:5).3')
SYMDEF(&DLIB1=&SYSR1(1:3).DL1')
SYMDEF(&DLIB2=&SYSR1(1:3).DL2')
SYMDEF(&DLIB3=&SYSR1(1:3).DL3')
SYMDEF(&DLIB4=&SYSR1(1:3).DL4')
SYMDEF(&ALLCLST1='CANCEL')
SYMDEF(&CMDLIST1=&SYSCONE.,00')
SYMDEF(&COMMDSN1='COMMON')
SYMDEF(&DFHSMCMD='00')
SYMDEF(&IFAPRDXX='00')
SYMDEF(&DCEPROC1='.')
SYMDEF(&DFSPROC1='.')
SYSDEF      HWNAME(SCZP901)
LPARNAME(A13)
SYSNAME(SC70)
SYSPARM(R3,70)
SYMDEF(&IFAPRDXX='4A')
SYMDEF(&DFHSMHST='6')
SYMDEF(&DFHSMPRI='NO')
SYMDEF(&DB2='V8')
```

Símbolos del sistema (system symbols) en PARMLIB

¿Dónde es definido &SYSR1?

&SYSR1 está definido por el sistema con el VOLSER del volumen IPL.
(volviendo al ejemplo ante anterior, se ven los valores de &SYSR1 y &SYSR2).

Los símbolos del sistema se utilizan en casos en los que varios sistemas z/OS comparten un único PARMLIB.

El uso de símbolos permite utilizar miembros individuales con sustitución simbólica, en lugar de que cada sistema requiera un miembro único.

El miembro LOADxx especifica el miembro IEASYMxx que debe utilizar el sistema.



Orden de búsqueda de programas

Los módulos (programas), ya sea que estén almacenados como módulos de carga u objetos de programa, deben cargarse tanto en el almacenamiento virtual como en el almacenamiento central antes de poder ejecutarse.

Cuando un módulo llama a otro módulo, ya sea directamente al solicitar su ejecución o indirectamente al solicitar un servicio del sistema que lo utiliza, no comienza a ejecutarse instantáneamente.

El tiempo que tarda un módulo solicitado en comenzar a ejecutarse depende de en qué parte de su orden de búsqueda el sistema encuentra una copia utilizable y de cuánto tiempo le toma al sistema hacer que la copia que encuentra esté disponible.

Orden de búsqueda de programas

Cuando se solicita un programa a través de un servicio del sistema (como LINK, LOAD, XCTL o ATTACH) utilizando las opciones predeterminadas (default), el sistema lo busca en la siguiente secuencia:

- 1) Job Pack Area (JPA)
- 2) TASKLIB
- 3) STEPLIB o JOBLIB
- 4) LPA
- 5) Bibliotecas en linklist, como se especifica en PROGxx y LNKLISTxx

Algunos subsistemas de IBM (especialmente CICS e IMS) y aplicaciones (como ISPF) usan estas funciones para establecer otros órdenes de búsqueda para los programas.



Orden de búsqueda de programas

Job Pack Area (JPA)

Un programa en JPA ya se ha cargado en el addressspace.

Si la copia en JPA se puede utilizar, se utilizará.

De lo contrario, el sistema busca una nueva copia o pospone la solicitud hasta que la copia en JPA este disponible.

(Por ejemplo, el sistema posterga una solicitud hasta que finalice una llamada anterior antes de volver a utilizar un módulo reutilizable (serially-reusable) que ya está en JPA).



Orden de búsqueda de programas

TASKLIB

Un programa puede alocar asignar uno o más data sets a una concatenación TASKLIB.

Los módulos cargados por tareas no autorizadas que se encuentran en TASKLIB deben llevarse al almacenamiento virtual de área privada antes de que puedan ejecutarse.

Los módulos que se han cargado previamente en el almacenamiento virtual de área común (módulos LPA o aquellos cargados por un programa autorizado en CSA) deben cargarse en el almacenamiento virtual de área común antes de que puedan ejecutarse.



Orden de búsqueda de programas

STEPLIB o JOBLIB

Estos son nombres de DD específicos que se pueden utilizar para alocar data sets, que se buscarán antes del orden de búsqueda predeterminado del sistema para programas.

Data sets se pueden alocar asignar a las concatenaciones STEPLIB y JOBLIB en JCL o por un programa que use alocacion dinámica.

Sin embargo, solo se buscarán módulos en uno u otro.

Si se asignan STEPLIB y JOBLIB para un paso de trabajo en particular, el sistema busca STEPLIB e ignora JOBLIB.



Orden de búsqueda de programas

STEPLIB o JOBLIB

Los data sets concatenado a STEPLIB o JOBLIB se buscaran después de cualquier TASKLIB pero antes de LPA.

Los módulos que se encuentran en STEPLIB o JOBLIB deben llevarse al almacenamiento virtual de área privada antes de que puedan ejecutarse.

Los módulos que se hayan cargado previamente en el almacenamiento virtual de área común (módulos LPA o aquellos cargados por un programa autorizado en CSA) deben cargarse en el almacenamiento virtual de área común antes de que puedan ejecutarse.

Orden de búsqueda de programas

LPA

- a. Módulos LPA dinámicos, como se especifica en los miembros PROGxx
- b. Módulos LPA fijos (FLPA), como se especifica en los miembros IEAFIXxx
- c. Módulos LPA modificados (MLPA), como se especifica en los miembros IEALPAxx.
- d. Módulos LPA paginables (PLPA), cargados desde bibliotecas especificadas en LPALSTxx o PROGxx.

Los módulos LPA se cargan en un almacenamiento común, compartido por todos los espacios de direcciones en el sistema.

Debido a que estos módulos son reentrantes y no se modifican por sí solos, cada uno puede ser utilizado por cualquier cantidad de tareas en cualquier address space al mismo tiempo.

Los módulos que se encuentran en LPA no necesitan ser llevados al almacenamiento virtual, porque ya están en el almacenamiento virtual..

Orden de búsqueda de programas

Bibliotecas en linklist, como se especifica en PROGxx y LNKLISTxx

De manera predeterminada (default), la lista de enlaces comienza con SYS1.LINKLIB, SYS1.MIGLIB y SYS1.CSSLIB.

Sin embargo, puede cambiar este orden utilizando SYSLIB en PROGxx y agregar otras bibliotecas a la concatenación de la lista de enlaces.

El sistema debe llevar los módulos que se encuentran en la lista de enlaces al almacenamiento virtual de área privada antes de que los programas puedan ejecutarse.



Configuración de devices de entrada/salida (I/O)

Las configuraciones de I/O para el sistema operativo (software) y el subsistema de canal (hardware) deben ser definidos.

El componente de definición de configuración de hardware (HCD) de z/OS consolida los procesos de configuración de I/O de hardware y software en una única interfaz de usuario final interactiva.

La salida de HCD es un archivo de definición de I/O (IODF), que contiene datos de configuración del I/O.

Un IODF se utiliza para definir múltiples configuraciones de hardware y software para el sistema operativo z/OS.



Configuración de devices de entrada/salida (I/O)

Cuando se activa un nuevo IODF, HCD define la configuración de I/O para el subsistema de canal o el sistema operativo, o ambos.

Con la función de activación de HCD o el comando de operador ACTIVATE de z/OS, se pueden realizar cambios en la configuración actual sin tener que realizar una carga inicial del programa (IPL) del software o un reinicio de encendido (POR) del hardware. Realizar cambios mientras el sistema está en funcionamiento se conoce como configuración dinámica o reconfiguración dinámica.

Consolas configuración

El funcionamiento de z/OS implica la gestión de hardware, como procesadores y dispositivos periféricos (incluidas las consolas donde los operadores realizan su trabajo); y software, como el sistema de control operativo z/OS, el subsistema de entrada de trabajos, los subsistemas (como NetView) que pueden controlar las operaciones automatizadas y todas las aplicaciones que se ejecutan en z/OS.

El funcionamiento de un sistema z/OS implica lo siguiente:

Procesamiento de mensajes y comandos que forma la base de la interacción del operador con z/OS y la base de la automatización de z/OS.

Operaciones de consola, o cómo los operadores interactúan con z/OS para supervisar o controlar el hardware y el software.

Consolas configuración

La planificación de las operaciones para para un sistema z/OS, debe tener en cuenta cómo los operadores usan las consolas para hacer su trabajo y cómo administran los mensajes y comandos.

El programador del sistema debe asegurarse de que los operadores reciban los mensajes necesarios en sus consolas para realizar sus tareas y seleccionar los mensajes adecuados para supresión, automatización u otros tipos de procesamiento de mensajes.

En términos de operaciones de z/OS, cómo la instalación establece la recuperación de la consola o si un operador debe volver a realizar la carga inicial de programa (IPL) de un sistema para cambiar las opciones de procesamiento son consideraciones de planificación importantes.



Consolas configuración

Dado que los mensajes también son la base de las operaciones automatizadas, el programador de sistemas debe comprender el procesamiento de mensajes para planificar la automatización de z/OS.

A medida que más instalaciones utilizan entornos multisistema, la necesidad de coordinar las actividades operativas de esos sistemas se vuelve crucial. Incluso para sistemas z/OS individuales, una instalación debe pensar en controlar la comunicación entre áreas funcionales.

En entornos tanto de un solo sistema como de varios, los comandos que los operadores pueden ingresar desde las consolas pueden ser un problema de seguridad que requiere una coordinación cuidadosa.

Consolas configuración

Como planificador, el programador de sistemas debe asegurarse de que las personas adecuadas estén realizando las tareas correctas cuando interactúan con z/OS.

Una configuración de consola consta de las distintas consolas que utilizan los operadores para comunicarse con z/OS.

Su instalación primero define los dispositivos de I/O que puede utilizar como consolas a través de la definición de configuración de hardware (HCD), una interfaz interactiva en el host que permite al programador del sistema definir la configuración de hardware tanto para el subsistema de canal como para el sistema operativo.



Consolas configuración

El Administrador de configuración de hardware (HMC) es la interfaz gráfica de usuario para HCD.

HMC interactúa con HCD en una relación cliente/servidor (es decir, HMC se ejecuta en una estación de trabajo y HCD se ejecuta en el host).

Los sistemas host requieren un modelo interno de sus conexiones a los dispositivos (devices), pero puede ser más conveniente y eficiente para el programador del sistema mantener (y complemente) ese modelo en forma visual.

Consolas configuración

HMC mantiene los datos de configuración como un diagrama en un archivo en la estación de trabajo sincronizado con el IODF en el host.

Si bien es posible utilizar HCD directamente para tareas de configuración de hardware, muchos clientes prefieren utilizar HMC exclusivamente, debido a su interfaz gráfica.

Además de HCD, una vez que se han definido los dispositivos (devices), se le indica a z/OS qué dispositivos (devices) utilizar como consolas especificando los números de device apropiados en el miembro CONSOLxx de PARMLIB.

Generalmente, los operadores de un sistema z/OS reciben mensajes e ingresan comandos en las consolas (MCS y SMCS).

Consolas configuración

Pueden usar otras consolas (como las consolas NetView) para interactuar con z/OS, pero aquí describimos las consolas MCS, SMCS y EMCS tal como se usan comúnmente en los sitios z/OS:

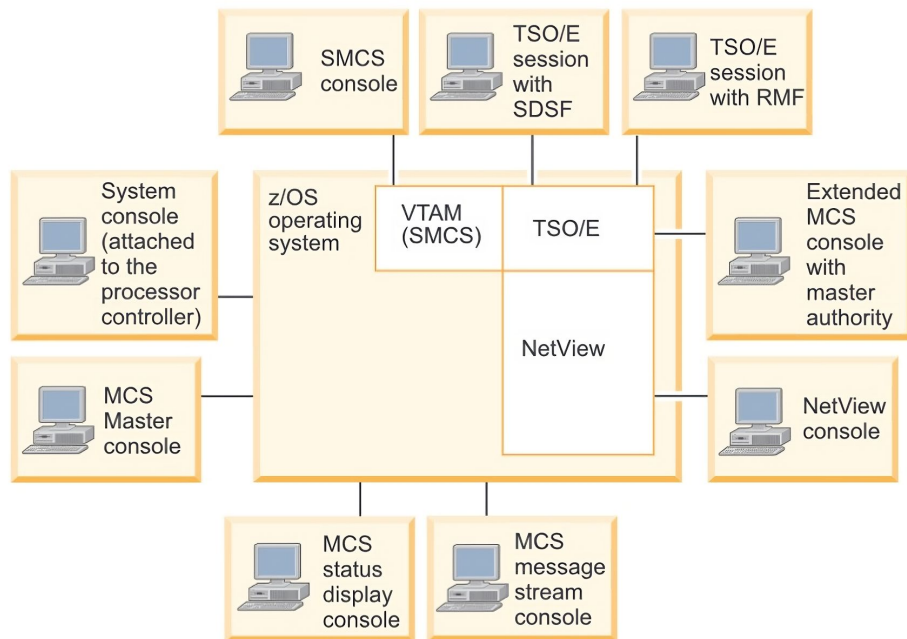
- **Las consolas MCS (Multiple Console Support)** son dispositivos que están conectados localmente a un sistema z/OS y proporcionan la comunicación básica entre operadores y z/OS.
- **Las consolas MCS** están conectadas a dispositivos de control que no admiten la arquitectura de red de sistemas ni los protocolos SNA.
- **Las consolas SMCS (SNA Multiple Console Support)** son dispositivos que no tienen que estar conectados localmente a un sistema z/OS y proporcionan la comunicación básica entre operadores y z/OS.
- **Las consolas SMCS** usan z/OS Communications Server para proporcionar comunicación entre operadores y z/OS, en lugar de E/S directa al dispositivo de consola.
- **Las consolas EMCS (Extended Multiple Console Support)** son dispositivos (distintos de las consolas MCS o SMCS) desde los cuales los operadores o programas pueden ingresar comandos y recibir mensajes.

Consolas configuración

La definición de consolas EMCS como parte de la configuración de la consola, permite al programador del sistema ampliar la cantidad de consolas más allá del límite de consolas MCS, que es de 99 para cada sistema z/OS en un sysplex.

Consolas configuración

El programador del sistema define estas consolas en una configuración según sus funciones. Los mensajes importantes que requieren una acción pueden dirigirse al operador, quien puede actuar ingresando comandos en la consola.



Otra consola puede actuar como un monitor para mostrar mensajes a un operador que trabaja en un área funcional como una biblioteca de grupos de cintas, o para mostrar mensajes sobre impresoras en su instalación.

La siguiente figura, muestra una configuración de consola para un sistema z/OS que también incluye la consola del sistema, una consola SMCS, NetView y TSO/E.

Consolas configuración

La función de consola del sistema se proporciona como parte de la consola de administración de hardware (HMC).

Un operador puede utilizar la consola del sistema para iniciar z/OS y otro software del sistema, y durante situaciones de recuperación cuando otras consolas no están disponibles.

Además de las consolas MCS y SMCS, el sistema z/OS que se muestra en la figura anterior, tiene una consola NetView definida.

NetView trabaja con mensajes del sistema y listas de comandos para ayudar a automatizar las tareas del operador de z/OS.

Consolas configuración

Muchas operaciones del sistema se pueden controlar desde una consola NetView.

Los usuarios pueden supervisar muchas funciones del sistema z/OS desde terminales TSO/E.

Mediante el uso de SDSF, y la medición de recursos (RMF), los usuarios de TSO/E pueden supervisar z/OS y responder a problemas de equilibrio de carga de trabajo y rendimiento.

Un usuario autorizado de TSO/E también puede iniciar una sesión de consola MCS extendida para interactuar con z/OS.



Consolas configuración

Las consolas MCS que se muestran en la figura anterior, son:

- **Una consola MCS** desde la que un operador puede ver mensajes e ingresar comandos z/OS
Esta consola está en modo de capacidad total (full capability) porque puede recibir mensajes y aceptar comandos.

Un operador puede controlar las operaciones del sistema z/OS desde una consola MCS o SMCS.

- **Una consola MCS en display mode (visualización de estado).**

Un operador puede ver información de estado del sistema desde los comandos DEVSERV, DISPLAY, TRACK o CONFIG.

Sin embargo, debido a que se trata de una consola de visualización de estado, un operador no puede ingresar comandos desde la consola.

Un operador en una consola de capacidad total puede ingresar estos comandos y enviar la salida a una consola de visualización de estado para su visualización.

- **Una consola MCS message-stream (flujo de mensajes).** Puede mostrar mensajes del sistema.

Un operador puede ver los mensajes enviados a esta consola.

Sin embargo, debido a que se trata de una consola de flujo de mensajes, un operador no puede ingresar comandos desde aquí.

Consolas configuración

En general, la consola de hardware (HMC) está separada, pero la mayoría de las otras terminales están combinadas.

Los sistemas se gestionan mediante alertas para condiciones de excepción desde un producto de automatización.

El Open Systems Adapter-Express Integrated Console Controller (OSA-ICC) es la forma de conectar consolas.

OSA-ICC utiliza conexiones TCP/IP a través de LAN Ethernet para conectarse a computadoras personales como consolas a través de una conexión TN3270 (telnet).



Inicio de z/OS: el proceso de inicialización (IPL)

¿Con qué frecuencia inicia o reinicia el sistema operativo de su computadora personal o portátil? Quizás una vez al día, si desea instalar actualizaciones o parches de seguridad.

Según el trabajo que esté realizando, es posible que reinicie casi automáticamente, sin realizar más que unas pocas operaciones de guardado.

Iniciar z/OS es, en un sentido muy general, un proceso similar, pero se realiza con mucha menos frecuencia y solo con mucha planificación previa.

La inicialización de z/OS, o una carga de programa inicial (IPL), es el acto de cargar una copia del sistema operativo desde el disco al almacenamiento real del procesador y ejecutarlo.

Este proceso consiste esencialmente en:

- Inicialización del sistema y del almacenamiento (storage), incluida la creación de address spaces de componentes del sistema
- Inicialización del Master scheduler, planificador maestro e inicialización de subsistemas.

Inicio de z/OS: el proceso de inicialización (IPL)

Los sistemas z/OS están diseñados para funcionar de forma continua con muchos meses entre recargas (IPLs), lo que permite que las cargas de trabajo de producción importantes estén disponibles de forma continua.

El cambio es la razón habitual para una recarga (IPL), y el nivel de cambio en un sistema dicta el cronograma de recarga.

Por ejemplo:

- Un sistema (o Lpar) de prueba puede cargarse a diario o incluso con mayor frecuencia.
- Un sistema bancario de alta disponibilidad puede cargarse solo una vez al año, o incluso con menor frecuencia, para actualizar los niveles de software.
- Las influencias externas pueden ser a menudo la causa de IPLs, como la necesidad de probar y mantener los sistemas de energía en la sala de máquinas.
- A veces, el software que posea un mal comportamiento y utiliza recursos del sistema que solo se pueden estabilizar con un nuevo IPL, pero este tipo de comportamiento normalmente es objeto de investigación y corrección.

Inicio de z/OS: el proceso de inicialización (IPL)

Muchos de los cambios que en el pasado requerían un IPL ahora se pueden realizar de manera dinámica.

Algunos ejemplos de estas tareas son:

- Agregar una biblioteca a la lista de linklist, para un subsistema como CICS.
- Agregar módulos a LPA.

El apagado de z/OS ocurre tan raramente como una IPL.

Para ‘bajar’ el sistema, cada tarea debe cerrarse por turno, en el orden correcto.

Las instalaciones de z/OS actuales utilizan alguna manera de automatización para controlar y ejecutar este proceso.

La bajada del sistema normalmente requiere un solo comando, lo que da como resultado la eliminación de la mayoría de las tareas excepto la tarea de automatización en sí.

La tarea de automatización se cierra manualmente, seguida de los comandos necesarios en caso de ‘sacar’ un sistema de un sysplex.

IPL: secuencia y controles

El proceso de inicialización comienza cuando el programador o el operador del sistema selecciona la función LOAD en la consola de administración de hardware (HMC).

Para realizar un IPL correcto en z/OS, el programador del sistema debe proporcionar la siguiente información:

- La dirección del disco del volumen IPL.
- El miembro LOADxx que contiene punteros a los parámetros del sistema.
- El dataset IODF que contiene la información de configuración.
- La dirección del dispositivo del volumen IODF.

z/OS localiza todo el almacenamiento central utilizable que está en línea y disponible, y comienza a crear las distintas áreas del sistema.

No todos los discos conectados a una CPU poseen código cargable.

Un disco que lo tiene se conoce generalmente como un disco "IPLable" y, más específicamente, como el volumen SYSRES.

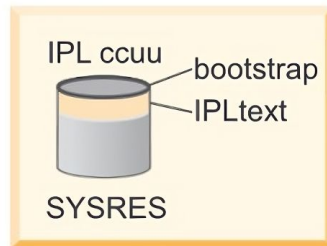
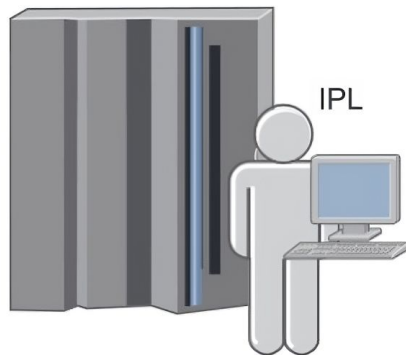
Los discos IPLable contienen un módulo de arranque (bootstrap) en la pista 0 del cilindro 0.

IPL: secuencia y controles

En el IPL, este arranque se carga en el almacenamiento en la dirección real cero y se le pasa el control.

El bootstrap lee entonces el programa de control IPL, IEAIPL00 (también conocido como texto IPL) y le pasa el control.

Esto, a su vez, inicia la tarea más compleja de cargar el sistema operativo y ejecutarlo.



LOADPARM



IPL: secuencia y controles

Una vez cargado el programa de arranque (bootstrap) y pasado el control a IEA IPL00; IEA IPL00 prepara un entorno adecuado para iniciar los programas y módulos que componen el sistema operativo, de la siguiente manera:

1. Limpia el almacenamiento central a cero antes de definir áreas de almacenamiento para el master scheduler.
2. Ubica el dataset de SYS1.NUCLEUS en el volumen SYSRES y carga una serie de programas desde allí conocidos como: IPL Resource Initialization Modules (IRIMs).
3. Estos IRIMs comienzan a crear el entorno normal del sistema operativo, bloques de control y subsistemas.



IPL: secuencia y controles

Algunas de las tareas más importantes que realizan los IRIMs son las siguientes:

- Leer la información LOADPARM introducida en la consola de hardware en el momento en que se ejecutó el mandato IPL.
- Buscar el dataset de IODF en el volumen especificado en el miembro LOADPARM. IRIM intentará primero localizar LOADxx en SYS0.IPLPARM.
- Si no lo consigue, buscará SYS1.IPLPARM, y así sucesivamente, hasta SYS9.IPLPARM incluido.
- Si en este punto todavía no se ha localizado, la búsqueda continúa en SYS1.PARMLIB. (Si no se puede localizar LOADxx, el sistema carga un estado de load wait).
- Si se encuentra un miembro LOADxx, abrir y leer su información, incluido el sufijo del núcleo (a menos que se anule (overridden) en LOADPARM), el nombre del catálogo maestro y el sufijo del miembro IEASYSxx que se utilizará.

IPL: secuencia y controles

- Cargar el núcleo del sistema operativo.
- Inicializa virtual storage en el master scheduler address space, para la System Queue Area (SQA), la SQA Extendida (ESQA), la local SQA (LSQA) y la Prefixed Save Area.
- Al final de la secuencia de IPL, la PSA reemplazará a IEAIPL00 en la ubicación de almacenamiento real cero, donde permanecerá.
- Inicializa la gestión de almacenamiento real (real storage management), incluida la tabla de segmentos para el master scheduler, las entradas de la tabla de segmentos para las áreas de almacenamiento comunes (common storage áreas) y la tabla de marcos de página (page frame table).

IPL: secuencia y controles

El último de los IRIM carga entonces la primera parte del programa del Nucleus Initialization Program (NIP), que invoca los módulos de Resource Initialization Modules (RIMs), uno de los primeros de los cuales inicia las comunicaciones con la consola del NIP definida en el IODF.

Puede durante la etapa del NIP el sistema solicitar, al programador o al operador del sistema que proporcione parámetros del sistema que controlen el funcionamiento de z/OS.

El sistema también emite mensajes informativos sobre las etapas del proceso de inicialización.

IEASYSnn, un miembro de PARMLIB, contiene parámetros y punteros que controlan la dirección que toma la IPL.

El programador o el operador del sistema pueden modificar estos parámetros según sea necesario.

IEASYSnn, un miembro de PARMLIB, contiene parámetros y punteros que controlan la dirección que toma la IPL.

IPL: secuencia y controles

La siguiente figura, ilustra el contenido parcial de un miembro IEASYSxx.

```
-----  
File Edit Edit Settings Menu Utilities Compilers Test Help  
-----  
EDIT          SYS1.PARMLIB(IEASYS00) - 01.68          Columns 00001 00072  
Command ==>          Scroll ==> CSR  
***** Top of Data *****  
000001 ALLOC=00,  
000002 APG=07,  
000003 CLOCK=00,  
000004 CLPA,  
000005 CMB=(UNITR,COMM,GRAPH,CHDR),  
000006 CMD=(&CMDLIST1.),  
000007 CON=00,  
000008 COUPLE=00, WAS FK  
000009 CSA=(2M,128M),  
000010 DEVSUP=00,  
000011 DIAG=00,  
000012 DUMP=DASD,  
000013 FIX=00,  
000014 GRS=STAR,  
000015 GRSCNF=ML,  
000016 GRSRNL=02,  
000017 IOS=00,  
000018 LNKAUTH=LNKLST,  
000019 LOGCLS=L,  
000020 LOGLMT=999999,  
000021 LOGREC=SYS1.&SYSNAME..LOGREC,  
000022 LPA=(00,L),  
000023 MAXUSER=1000,  
000024 MSTRJCL=00,  
000025 NSYSLX=250,  
000026 OMVS=&OMVSPARM.,
```


IPL: secuencia y controles

El sistema continúa el proceso de inicialización, interpretando y actuando sobre los parámetros del sistema que se especificaron.

- El NIP lleva a cabo las siguientes funciones de inicialización principales:
- Expande la SQA y la SQA extendida de acuerdo a las cantidades especificadas en el parámetro del sistema SQA.
- Crea el área de: PLPA (paginable link pack área) y la PLPA extendida ante un IPL de inicio en frío; restablece (resetea) tablas para que coincidan con una PLPA existente y una PLPA extendida para un inicio mas ágil o un IPL de inicio en caliente (warm start).

IPL: secuencia y controles

- Carga módulos en la FLPA (fixed link pack area) o en el FLPA extendida. El NIP lleva a cabo esta función solo si se especifica el parámetro del sistema FIX.
- Carga módulos en el área MLPA (modified link pack area) y en la MLPA extendida. El NIP lleva a cabo esta función solo si se especifica el parámetro del sistema MLPA.
- Aloca almacenamiento virtual para la CSA (common service área) y a la CSA extendida. La cantidad de almacenamiento asignado depende de los valores especificados en el parámetro del sistema CSA en el IPL.
- Protege las páginas de NUCMAP; PLPA y PLPA extendida; MLPA y MLPA extendida; FLPA y FLPA extendida; y partes del núcleo. Una instalación puede override (anular la protección de páginas) de MLPA y FLPA especificando NOPROT en los parámetros del sistema MLPA y FIX.

IPL: secuencia y controles

Para ver información de sobre cómo se realizó la carga inicial inicial (IPL) del sistema, se puede ejecutar el comando: D IPLINFO.

```
D IPLINFO
IEE254I 11.11.35 IPLINFO DISPLAY 906
SYSTEM IPLED AT 10.53.04 ON 08/15/2007
RELEASE z/OS 01.07.00 LICENSE = z/OS
USED LOADS8 IN SYS0.IPLPARM ON C730
ARCHLVL = 2 MTLSHARE = N
IEASYM LIST = XX
IEASYS LIST = (R3,65) (OP)
IODF DEVICE C730
IPL DEVICE 8603 VOLUME Z17RC1
```

IPL del sistema: creación de address space e inicialización de subsistemas

Además de inicializar áreas del sistema, z/OS establece address spaces de componentes del sistema.

Establece un address space para el master scheduler y otros address spaces para varios subsistemas y componentes del sistema.

Algunos de los address space de componentes son: *MASTER*, ALLOCAS, APPC, CATALOG, etc.

El address space del master scheduler es el primer address space que se crea (ASID=1).



IPL del sistema: creación de address space e inicialización de subsistemas

Luego, el master scheduler iniciar el subsistema de entrada de trabajo principal (JES2 o JES3).

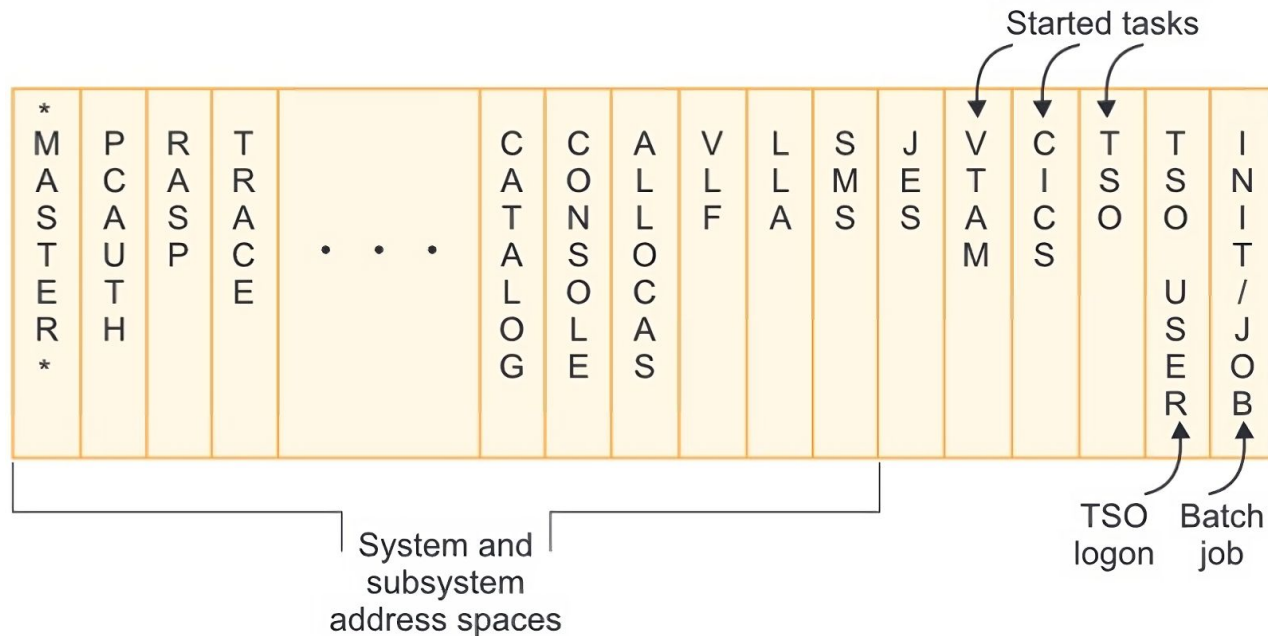
En muchos sistemas de producción, JES no se inicia inmediatamente; en su lugar, una automatización inicia todas las tareas en una secuencia controlada.

Luego, se inician otros subsistemas definidos.

Todos los subsistemas se definen en la biblioteca PARMLIB, en el miembro IEFSSNxx. La próxima figura muestra algunos de los address space de componentes del sistema importantes; para las tareas iniciadas VTAM, CICS, TSO; y para un usuario TSO y un iniciador batch (init).

IPL del sistema: creación de address space e inicialización de subsistemas

Cada espacio de direcciones tiene 2 GB de almacenamiento virtual de forma predeterminada, independientemente de si el sistema se ejecuta en modo de 31 bits o de 64 bits.



IPL del sistema: creación de address space e inicialización de subsistemas

Las áreas privadas están disponibles solo para ese address space, pero las áreas comunes están disponibles para todos.

Una vez que se inicializa el sistema y el subsistema de entrada de trabajos está activo, la instalación puede enviar trabajos para su procesamiento mediante el comando START, LOGON o MOUNT.

Cuando se activa un trabajo mediante START (para Jobs batch), LOGON (para trabajos de tiempo compartido (time-sharing jobs)) o MOUNT, se asigna un nuevo address space.



IPL del sistema: The master scheduler subsystem

El subsistema master scheduler establece comunicación entre el sistema operativo y el subsistema de entrada de trabajo principal, que puede ser JES2 o JES3.

Para realizar su trabajo, el subsistema del master scheduler utiliza un miembro MSTJCLxx, comúnmente llamado master JCL, que contiene data definition (DD) statements para todos los datasets de entrada y salida del sistema que se necesitan para realizar la comunicación entre el sistema operativo y JES.

Se puede encontrar un módulo de carga MSTJCL00 inicial en la biblioteca SYS1.LINKLIB.

IPL del sistema: The master scheduler subsystem

Tal como se envía, MSTJCL00 contiene una statement DD IEFPSI que define el dataset que contiene procedimiento para las tareas conjunto de datos que contiene el JCL de origen del procedimiento 'source' JCL para las started tasks.

Normalmente, este dataset es SYS1.PROCLIB; puede ser una concatenación. Para que se realice un trabajo real, SYS1.PROCLIB debe contener al menos el procedimiento para el JES principal.

Si se requieren modificaciones, el procedimiento recomendado es crear un nuevo miembro MSTJCLxx en PARMLIB.

El sufijo se especifica mediante el parámetro MSTRJCL en el miembro IEASYSxx de PARMLIB.

IPL del sistema: The master scheduler subsystem

Un miembro MSTJCLxx a modo de ilustración:

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
EDIT      SYS1.PARMLIB(MSTJCL00) - 01.07                Columns 00001 00072
Command ==>                                           Scroll ==> CSR
***** Top of Data *****
000100 //MSTRJCL JOB MSGLEVEL=(1,1),TIME=1440
000200 //          EXEC PGM=IEEMB860,DPRTY=(15,15)
000300 //STCINRDR DD SYSOUT=(A,INTRDR)
000400 //TSOINRDR DD SYSOUT=(A,INTRDR)
000500 //IEFPDSI DD DSN=SYS1.PROCLIB,DISP=SHR
000600 //          DD DSN=CPAC.PROCLIB,DISP=SHR
000700 //          DD DSN=SYS1.IBM.PROCLIB,DISP=SHR
000800 //IEFJOBS DD DSN=SYS1.STCJOBS,DISP=SHR
000900 //SYSUADS DD DSN=SYS1.UADS,DISP=SHR
***** Bottom of Data *****
```

IPL del sistema: The master scheduler subsystem

Cuando el master scheduler tiene que procesar el inicio de una started task, el sistema determina si el comando START se refiere a un procedimiento o a un job.

Si la DD IEFJOBS existe en el miembro MSTJCLxx, el sistema busca en la concatenación DD IEFJOBS el miembro solicitado en el comando START.

Si no hay ningún miembro con ese nombre en la concatenación IEFJOBS, o si la concatenación IEFJOBS no existe, el sistema busca en la DD IEFPSI el miembro solicitado en el comando START.

Si se encuentra el miembro, el sistema examina el primer registro en busca de una declaración JOB válida y, si existe, utiliza el miembro como el source JCL para la started task.

IPL del sistema: The master scheduler subsystem

Si el miembro no tiene una declaración JOB válida en su primer registro, el sistema supone que el JCL de origen es un procedimiento y crea un JCL para invocar el procedimiento.

Una vez que se ha creado (o encontrado) la fuente JCL, el sistema procesa el JCL.

Tal como viene, MSTJCL00 contiene una declaración IEFPDSI DD que define el dataset que contiene el procedimiento source JCL, para las started tasks. Normalmente, este dataset es SYS1.PROCLIB; puede ser una concatenación.



IPL del sistema: Job Entry Subsystem (JES)

Para inicializar JES, el subsistema master scheduler busca en SYS1.PROCLIB el procedimiento catalogado de JES.

Este procedimiento define las bibliotecas de procedimientos relacionados con los trabajos, incluidas aquellas que el subsistema JES utiliza para ubicar las statements de inicialización de JES.

Un ejemplo de procedimiento catalogado de JES es el siguiente:

```
//PROC00 DD DSN=SYS1.PROCLIB,DISP=SHR
//      DD DSN=SYS3.PROD.PROCLIB,DISP=SHR
//PROC01 DD DSN=SYS1.PROC2,DISP=SHR
...
//PROC99 DD DSN=SYS1.LASTPROC,DISP=SHR
...
```

Existen instalaciones que tienen listas muy largas de bibliotecas de procedimientos en el procedimiento JES.

Se debe tener cuidado con la cantidad de usuarios que pueden eliminar estas bibliotecas porque JES no se iniciará si falta una.

Normalmente, una biblioteca que está en uso no se puede eliminar, pero JES no las retiene, aunque las utiliza todo el tiempo.

IPL del sistema: Job Entry Subsystem (JES)

Puede anular la especificación predeterminada especificando esta declaración:

```
/*JOBPARM PROCLIB=
```

Después del nombre de la biblioteca de procedimientos, debe codificar el nombre de la declaración DD en el procedimiento JES2 que apunta a la biblioteca que se utilizará.

Si desea utilizar un procedimiento que reside en SYS1.LASTPROC, deberá incluir esta declaración en el JCL:

```
/*JOBPARM PROCLIB=PROC99
```

Otra forma de especificar una biblioteca de procedimientos es utilizar la declaración JCL JCLLIB.

```
//MYJOB JOB  
//MYLIBS JCLLIB ORDER=(MY.PROCLIB.JCL,SECOND.PROCLIB.JCL)  
//S1 EXEC PROC=MYPROC1  
...
```

IPL del sistema:Job Entry Subsystem (JES)

Esta declaración le permite codificar y utilizar procedimientos sin utilizar bibliotecas de procedimientos del sistema.

El sistema busca las bibliotecas en el orden en el que se las especifica en la declaración JCLLIB, antes de buscar cualquier biblioteca de procedimientos del sistema.

IPL del sistema: Job Entry Subsystem (JES)

Suponiendo que la biblioteca de procedimientos predeterminada del sistema incluye solo SYS1.PROCLIB, el sistema busca en las bibliotecas el procedimiento MYPROC1 en el siguiente orden:

1. MY.PROCLIB.JCL
2. SECOND.PROCLIB.JCL
3. SYS1.PROCLIB

IPL del sistema: Inicialización de subsistemas adicionales

La inicialización de un subsistema es el proceso de preparar al subsistema para su uso en el sistema.

Los miembros IEFSSNxx de SYS1.PARMLIB contienen las definiciones de los subsistemas principales, como JES2 o JES3, y los subsistemas secundarios, como NetView y DB2.

Durante la inicialización del sistema, se inicializan los subsistemas definidos.

IPL del sistema: Inicialización de subsistemas adicionales

Debe definir primero el subsistema principal (JES), ya que otros subsistemas, como DB2, requieren los servicios del subsistema principal en sus rutinas de inicialización. Pueden surgir problemas si los subsistemas que utilizan el servicio de afinidad de subsistema en su inicialización se inicializan antes que el subsistema principal. Después de que se inicializa el JES principal, los subsistemas se inicializan en el orden en el que se especifican los miembros PARMLIB de IEFSSNxx mediante el parámetro SSN.

Por ejemplo, para SSN=(aa,bb), el miembro PARMLIB IEFSSNa_a se procesaría antes de IEFSSNb_b.



Tipos de IPL

Según las circunstancias en las que desee iniciar o reiniciar z/OS, puede utilizar uno de los tres tipos de inicialización:

- **Cold start**
- **Quick start**
- **Warm**

La principal diferencia entre los tres tipos es si se trata o no cierto almacenamiento.

Tipos de IPL: Cold start

Un IPL que carga (o recarga) el área PLPA (pageable link pack área) y borra (clear) los VIO (virtual input/output) data set pages.

El primer IPL después de la instalación del sistema siempre es un inicio en frío porque se carga inicialmente el PLPA.

Los IPL posteriores son inicios en frío cuando se vuelve a cargar el PLPA, ya sea para modificar su contenido o para restaurarlo si se perdió.

Esto se hace generalmente cuando se han realizado cambios en el LPA (por ejemplo, cuando se carga un nuevo SYSRES que contiene mantenimiento).

VIO es un método de uso de memoria para almacenar pequeños conjuntos de datos temporales para un acceso rápido.

Obviamente, no debería haber demasiados datos almacenados de esta manera, por lo que el tamaño está restringido.

Tipos de IPL:Quick start

Un IPL que no vuelve a cargar el PLPA, sino que borra (clear) los VIO data set page.

(El sistema resetea las tablas de páginas y segmentos, para que coincidan con el último PLPA creado).

Esto se hace si no ha habido cambios en la LPA, pero se debe refrescar VIO.

Esto evita el inicio en caliente de trabajos que estaban usando VIO data sets.

Tipos de IPL: Warm start

Un IPL que no vuelve a cargar el PLPA y conserva los VIO data set pages.

Esto permitiría que los Jobs que se estaban ejecutando en el momento del IPL se reinicien con el contenido de los VIO data sets.

A menudo, el enfoque preferido es realizar un IPL de inicio en frío (especificando CLPA).

Se pueden utilizar las otras opciones, pero se debe tener mucho cuidado para evitar cambios inesperados o la anulación de los cambios.

Tipos de IPL: Warm start

Se puede utilizar un inicio en caliente cuando se tienen trabajos de larga duración que se desea reiniciar después de la IPL, pero un enfoque mas adecuado es dividir esos trabajos en partes más pequeñas que pasen conjuntos de datos en datasets, en lugar de utilizar VIO.

Los controladores de disco modernos con gran memoria caché han reducido la necesidad de conservar los datos VIO durante períodos prolongados.

Además, no hay que confundir un IPL de arranque en frío (normalmente se usaría CLPA en lugar del término "arranque en frío") con un arranque en frío JES.

El arranque en frío JES es algo que no se hace de manera habitual; en un sistema de producción, ya que destruye por completo los datos existentes en JES.

z/OS system tuning

La tarea de "tuning" del sistema es un proceso iterativo y continuo, y es la disciplina que afecta más directamente a todos los usuarios de los recursos del sistema en la empresa.

El componente z/OS Workload Management (WLM) es una parte importante de este proceso e incluye el ajuste inicial de la selección de parámetros apropiados para varios componentes y subsistemas del sistema.

z/OS system tuning

Una vez que el sistema está operativo y se han establecido los criterios para la selección de trabajos para su ejecución a través de clases de trabajos y prioridades, WLM controla la distribución de los recursos disponibles de acuerdo con los parámetros especificados por la instalación.

Sin embargo, WLM puede manejar sólo los recursos disponibles.

Si estos son inadecuados para satisfacer las necesidades de la instalación, incluso la distribución óptima puede no ser la respuesta; se deben examinar otras áreas del sistema para determinar la posibilidad de aumentar los recursos disponibles.

Cuando los requisitos del sistema aumentan y se vuelve necesario cambiar las prioridades o adquirir recursos adicionales (como un procesador más grande, más almacenamiento o más terminales), el programador del sistema necesita modificar los parámetros de WLM para reflejar las nuevas condiciones.

Mantenimiento de software z/OS

El programador de sistemas z/OS, será su responsabilidad garantizar que todos los productos de software y sus modificaciones estén instalados correctamente en el sistema.

También deberá asegurarse de que todos los productos estén instalados en el nivel adecuado para que los elementos del sistema puedan trabajar juntos.

Al principio, esto puede no parecer demasiado difícil, pero a medida que aumenta la complejidad de la configuración del software, también lo hace la tarea de supervisar todos los elementos del sistema.

La administración del centro de datos suele ser responsable de los acuerdos de nivel de servicio (SLA), a menudo a través de un equipo especializado de gerentes de servicio.



Mantenimiento de software z/OS

Las prácticas y procesos de control de cambios en un centro de datos se implementan para garantizar que se cumplan los SLA.

El programa SMP/E, es el medio principal para instalar y actualizar el software en un sistema z/OS.

SMP/E consolida los datos de instalación, permite una mayor flexibilidad para seleccionar los cambios que se instalarán, proporciona una interfaz de diálogo y soporta la alocacion dinamica de data sets.

SMP/E se puede ejecutar en batch o utilizando diálogos de ISPF.

Con los diálogos de SMP/E, puede consultar de forma interactiva la base de datos de SMP/E y crear y enviar trabajos para procesar comandos de SMP/E.

El software que se va a instalar con SMP/E debe empaquetarse como modificaciones del sistema o SYSMOD (system modifycation), que combinan el elemento actualizado con información de control.

Mantenimiento de software z/OS

Esta información describe los elementos y cualquier relación que tenga el software con otros productos o servicios que también puedan instalarse en el mismo sistema.

El JCL y los comandos de SMP/E serán utilizados con frecuencia por un programador de sistemas z/OS, sin embargo, las instrucciones de declaración de control de modificación -MCS- (Modification Control Statement) de SMP/E muy rara vez serán codificadas por el mismo programador de sistemas.

El producto y el empaquetado de SYSMOD incluirán las declaraciones MCS necesarias.

Una responsabilidad fundamental del programador de sistemas es trabajar con el soporte de defectos de IBM cuando surge un problema en z/OS o en productos opcionales de IBM.

La resolución de problemas requiere que el programador del sistema reciba (receive) y aplique (apply) las correcciones (fixes) al sistema.

Using SMP/E

SMP/E es la herramienta z/OS para gestionar la instalación de productos de software en un sistema z/OS y para realizar un seguimiento de las modificaciones de dichos productos.

SMP/E controla estos cambios a nivel de componente mediante:

- La selección de los niveles de código adecuados que se instalarán de entre una gran cantidad de cambios potenciales.
- La invocación de programas de utilidad del sistema para instalar los cambios.
- Llevar registro de los cambios instalados mediante una función que le permite consultar el estado de su software y revertir el cambio si es necesario.



Using SMP/E

Todo el código y sus modificaciones se encuentran en la base de datos SMP/E denominada inventario de software consolidado (CSI), que está compuesta por uno o más conjuntos de datos VSAM.

SMP/E se puede ejecutar mediante batch o mediante cuadros de diálogo en ISPF/PDF. Con los cuadros de diálogo SMP/E, puede consultar de forma interactiva la base de datos SMP/E y crear y enviar jobs para procesar comandos SMP/E.

SMP/E: Visión SMP/E del sistema

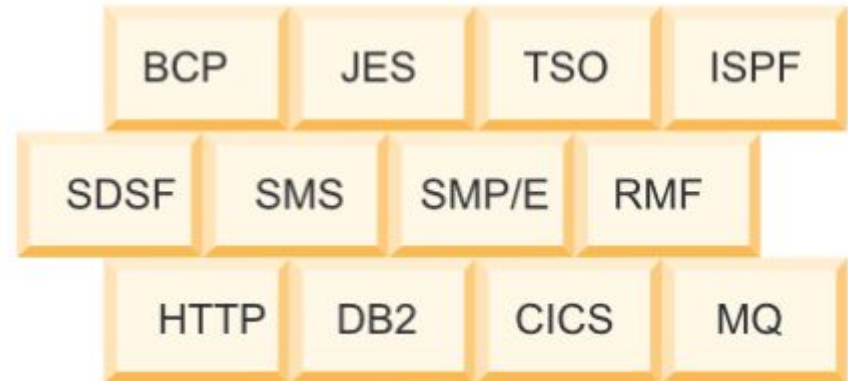
Un sistema z/OS puede parecer un gran bloque de código que controla la CPU.

En realidad, z/OS es un sistema complejo que comprende muchos bloques de código más pequeños y diferentes, y así es como SMP/E ve a z/OS.

Cada uno de esos bloques de código más pequeños realiza una función específica dentro del sistema.

Funciones que pueden aparecer en un sistema z/OS, incluyen:

- Base Control Program (BCP)
- Job entry subsystem (JES2 or JES3)
- Time Sharing Option/Extensions (TSO/E)
- Interactive System Productivity Facility (ISPF)
- System Display and Search Facility (SDSF)
- Data Facility Storage Management Subsystem (DFSMS)
- System Modification Program Extended (SMP/E)
- z/OS UNIX System Services (z/OS UNIX)
- Resource Measurement Facility (RMF)
- HTTP Server
- DB2
- Customer Information Control System (CICS)
- WebSphere MQ
- IMS



SMP/E: Visión SMP/E del sistema

Cada función del sistema está compuesta por uno o más módulos de carga.

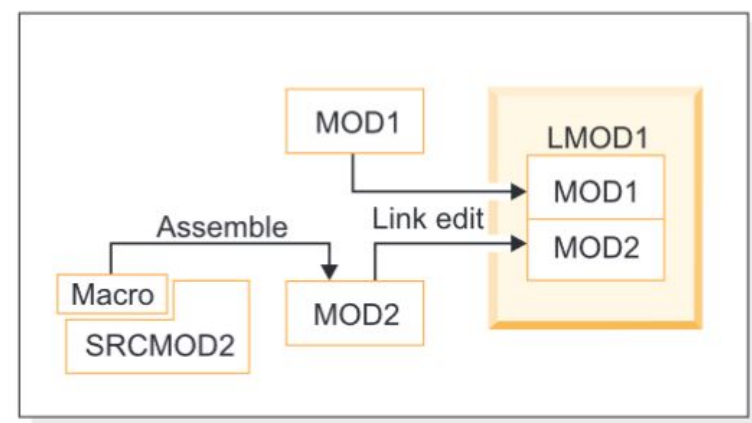
En un entorno z/OS, un módulo de carga representa la unidad básica de código ejecutable y legible por máquina.

Los módulos de carga se crean combinando uno o más módulos de objetos y procesándolos con una utilidad conocida como: link-edit.

El link-editing de módulos, es un proceso que resuelve referencias externas y direcciones.

Por lo tanto, las funciones de su sistema son uno o más módulos de objetos (object modules) que han sido combinados y link-edited.

Para ver de dónde proviene el módulo de objetos, observemos la siguiente figura ejemplo:



SMP/E: Visión SMP/E del sistema

La mayoría de las veces, object modules son enviados como parte de un producto.

En el ejemplo previo, el módulo de objeto MOD1 se envió como parte del producto.

En otras ocasiones, es posible que deba ensamblar el código fuente que le envían los empaquetadores de productos para crear el módulo de objeto.

Puede modificar el código fuente y luego ensamblarlo para producir un módulo de objeto.

En el ejemplo, SRCMOD2 es el código fuente que se ensambla para crear el módulo de objeto MOD2.

Una vez ensamblado, se link-edit el módulo de objeto MOD2 con el módulo de objeto MOD1 para formar el módulo de carga LMOD1.

SMP/E: Visión SMP/E del sistema

Además de los módulos de objeto y el código fuente, la mayoría de los productos distribuyen muchas otras partes, como macros, paneles de ayuda, CLIST y otros miembros de la biblioteca z/OS.

Estos módulos, macros y otros tipos de datos y código, son los bloques de construcción básicos del sistema.

Todos estos bloques de construcción se denominan elementos.

Los elementos están asociados con otros productos o servicios que pueden estar instalados en el mismo sistema z/OS y dependen de ellos.

Describen la relación que tiene el software con otros productos o servicios que pueden estar instalados en el mismo sistema z/OS.



SMP/E: similar a una biblioteca pública

Para realizar su procesamiento correctamente, SMP/E debe mantener una gran cantidad de información sobre la estructura, el contenido y el estado de modificación del software que administra.

Piense en toda la información que SMP/E tiene que mantener como si fuera toda la información contenida en la biblioteca pública.

En una biblioteca pública, verá estanterías llenas de libros y un catálogo de tarjetas con cajones que contienen una tarjeta para cada libro de la biblioteca.

Estas tarjetas contienen información, como el título, el autor, las fechas de publicación, el tipo de libro y un indicador del libro real en la estantería.



SMP/E: similar a una biblioteca pública

En el entorno SMP/E, hay dos tipos distintos de "estanterías".

Se las conoce como bibliotecas de distribución y bibliotecas de destino (target).

De la misma manera que las estanterías de la biblioteca pública contienen los libros de la biblioteca, las bibliotecas de distribución y target contienen los elementos del sistema.

Las bibliotecas de distribución contienen todos los elementos, como módulos y macros, que se utilizan como entrada para ejecutar el sistema.

Un uso muy importante de las bibliotecas de distribución es la copia de seguridad.

Si se produce un error grave con un elemento del sistema de producción, el elemento se puede reemplazar por un nivel estable que se encuentra en las bibliotecas de distribución.

SMP/E: similar a una biblioteca pública

Las bibliotecas de destino (target) contienen el código ejecutable necesario para ejecutar el sistema (por ejemplo, las bibliotecas desde las que se ejecuta el sistema de producción o el sistema de prueba).

Si se piensa en la analogía de la biblioteca pública, se puede ver que hay una parte importante de esa imagen que aún no ha sido considerada.

En la biblioteca pública, hay un catálogo de tarjetas que ayudan a encontrar el libro o la información que se está buscando.

SMP/E proporciona el mismo tipo de mecanismo de seguimiento en forma de inventario de software consolidado (CSI).



SMP/E: similar a una biblioteca pública

Los conjuntos de datos de CSI contienen toda la información que SMP/E necesita para rastrear la distribución y las bibliotecas de destino (target).

Como el catálogo de tarjetas contiene una tarjeta para cada libro de la biblioteca, el CSI contiene una entrada para cada elemento en sus bibliotecas.

Las entradas de CSI contienen el nombre del elemento, el tipo, el historial, cómo se introdujo el elemento en el sistema y un indicador del elemento en las bibliotecas de distribución y destino.



SMP/E: similar a una biblioteca pública

El CSI no contiene el elemento en sí, sino una descripción del elemento que representa.

Las tarjetas en el catálogo de tarjetas de la biblioteca pública están ordenadas alfabéticamente por el apellido del autor y por el tema y el título del libro.

En el CSI, las entradas de los elementos de las bibliotecas de distribución y destino se agrupan según su estado de instalación.

Es decir, las entradas que representan elementos encontrados en las bibliotecas de distribución se encuentran en la zona de distribución.



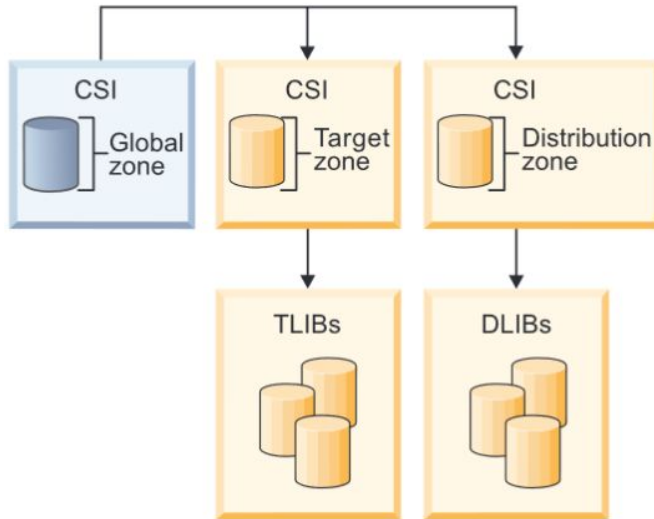
SMP/E: similar a una biblioteca pública

Las entradas que representan elementos encontrados en las bibliotecas de destino se encuentran en la zona de destino (target).

Ambas zonas tienen la misma función que los cajones del catálogo de tarjetas de la biblioteca pública.

Además de las zonas de distribución y destino, el CSI de SMP/E también contiene una zona global.

SMP/E: similar a una biblioteca pública



La zona global (Global Zone) contiene:

- Entradas necesarias para identificar y describir cada zona de destino (target) y distribución para SMP/E.
- Información sobre las opciones de procesamiento de SMP/E.
- Información de estado para todas las modificaciones del sistema (conocidas como SYSMOD) que SMP/E ha comenzado a procesar.
- Datos de excepción para SYSMOD que requieren un manejo especial o que están en error.

SMP/E: similar a una biblioteca pública

En SMP/E, el término excepción de datos, generalmente se refiere a HOLDDATA. HOLDDATA se suministra a menudo para un producto para indicar que una SYSMOD específica debe retenerse desde la instalación.

Las razones para retener un SYSMOD pueden ser:

- Un PTF tiene un error y no debe instalarse hasta que se corrija el error (ERROR HOLD).
- Es posible que se requieran determinadas acciones del sistema antes de la instalación del SYSMOD (SYSTEM HOLD).
- Es posible que el usuario desee realizar algunas acciones antes de instalar el SYSMOD (USER HOLD).



SMP/E: similar a una biblioteca pública

Toda la información ubicada en la zona global, combinada con la información que se encuentra en las zonas de destino y distribución, componen los datos que SMP/E requiere para instalar y realizar el seguimiento del software del sistema, que a menudo es una gran cantidad de datos.

Puede mostrar esta información utilizando las siguientes funciones de SMP/E:

- Cuadros de diálogo de consulta: la forma más fácil y rápida de obtener solo la información que desea.
- Comando LIST: cuando necesita una lista impresa completa de información sobre su sistema.
- Comandos REPORT: para verificar y comparar el contenido de la zona y generar una salida de comando que se pueda utilizar para actualizar su sistema.
- Interfaz de programación de aplicaciones CSI de SMP/E (API): para escribir un programa de aplicación para consultar el contenido de los conjuntos de datos CSI de su sistema.

SMP/E: data sets para inventario, zonas y libraries

Para instalar elementos z/OS en bibliotecas de destino y distribución, SMP/E utiliza una base de datos compuesta por varios tipos de conjuntos de datos.

SMP/E: SMPCSI (CSI) data sets

Estos son conjuntos de datos VSAM, que se utilizan para controlar el proceso de instalación y registrar los resultados del procesamiento.

Un CSI se puede dividir en varias particiones a través de la estructura de claves VSAM. Cada partición se denomina zona.

SMP/E: SMPCSI (CSI) data sets

Hay tres tipos de zonas:

- Se utiliza una única zona global para registrar información sobre los SYSMOD que se han recibido en el conjunto de datos SMPPTS.
- La zona global también contiene información que permite a SMP/E acceder a los otros dos tipos de zonas, información sobre las utilidades del sistema que SMP/E invoca para instalar elementos desde los SYSMOD e información que le permite personalizar el procesamiento de SMP/E.
- Se utilizan una o más zonas de destino (target) para registrar información sobre el estado y la estructura de las bibliotecas del sistema operativo (o de destino).
- Cada zona de destino (target) también apunta a la zona de distribución relacionada, que se puede utilizar durante las operaciones APPLY, RESTORE y LINK cuando SMP/E está procesando una SYSMOD y necesita comprobar el nivel de los elementos en las bibliotecas de distribución.
- Se utilizan una o más zonas de distribución para registrar información sobre el estado y la estructura de las bibliotecas de distribución (DLIB).
- Cada zona DLIB también apunta a la zona de destino relacionada, que se utiliza cuando SMP/E acepta un SYSMOD y necesita comprobar si ya se ha aplicado.

SMP/E: SMPCSI (CSI) data sets

Puede haber más de una zona, en un SMPCSI (de hecho, puede haber hasta 32766 zonas por data set).

Por ejemplo, un conjunto de datos SMPCSI puede contener una zona global, varias zonas objetivo (targets) y varias zonas de distribución.

Las zonas también pueden estar en SMPCSI separados.

Un conjunto SMPCSI puede contener solo la zona global, un segundo conjunto de datos SMPCSI las zonas objetivo (targets), y un tercer conjunto de datos SMPCSI las zonas de distribución.

SMP/E: SMPPTS (PTS) data set

Es un conjunto de datos para el almacenamiento temporal de SYSMOD que esperan ser instalados.

El PTS se utiliza estrictamente como un conjunto de datos de almacenamiento para SYSMOD.

El comando RECEIVE almacena SYSMOD directamente en el PTS sin ninguna modificación de la información SMP/E.

El PTS está relacionado con la zona global en que ambos conjuntos de datos contienen información sobre los SYSMOD recibidos.

Solo se puede utilizar un PTS para una zona global.

Por lo tanto, puede considerar la zona global y el PTS como un par de data sets que deben ser procesados (por ejemplo, eliminarse (deleted), guardarse (saved) o modificarse) simultáneamente.

SMP/E: SMPSCDS (SCDS) data set

Contiene copias de seguridad de las entradas de la zona de destino (target) modificadas durante el procesamiento de APPLY.

Por lo tanto, cada SCDS está directamente relacionado con una zona de destino (target) específica, y cada zona de destino debe tener su propio SCDS.

El SCDS es utilizado por SMP/E para almacenar copias de seguridad de las entradas de la zona de destino (target) modificadas durante el procesamiento de APPLY.

Por lo tanto, cada SCDS está directamente relacionado con una zona de destino (target) específica, y cada zona de destino debe tener su propio SCDS.

SMP/E: SMPSCDS (SCDS) data set

SMP/E también utiliza los siguientes data sets:

- El conjunto de datos SMPMTS (MTS) es una biblioteca en la que SMP/E almacena copias de macros durante la instalación cuando no se identifica ninguna otra biblioteca de macros de destino (target).
- Por lo tanto, el MTS está relacionado con una zona de destino específica y cada zona de destino debe tener su propio conjunto de datos MTS.
- El conjunto de datos SMPSTS (STS) es una biblioteca en la que SMP/E almacena copias de la fuente durante la instalación cuando no se identifica ninguna otra biblioteca de origen de destino. Por lo tanto, el STS está relacionado con una zona de destino específica, y cada zona de destino debe tener su propio conjunto de datos STS.
- El SMPLTS (LTS) es una biblioteca que mantiene la versión base de un módulo de carga.
- El módulo de carga de esta biblioteca especifica una asignación SYSLIB para incluir módulos implícitamente.
- Por lo tanto, el LTS está relacionado con una zona de destino (target) específica, y cada zona de destino (target) debe tener su propio LTS.
- Otros utilitarios y work data sets.

SMP/E: SMPSCDS (SCDS) data set

SMP/E utiliza la información de los CSI para seleccionar los niveles de elementos adecuados para la instalación, determinar qué bibliotecas deben contener qué elementos e identificar qué utilitarios del sistema deben llamarse para la instalación. Los programadores de sistemas también pueden utilizar los CSI para obtener la información más reciente sobre la estructura, el contenido y el estado del sistema.

SMP/E proporciona esta información en reportes, listados y cuadros de diálogo para ayudarle a:

- Investigar la función y los niveles servicio.
- Comprender las intersecciones y relaciones de los SYSMOD (ya sea instalados o en espera de ser instalados) .
- Generar job streams para el procesamiento de SMP/E.

```
//DEFINE      JOB  'accounting info',MSGLEVEL=(1,1)
//STEP01     EXEC  PGM=IDCAMS
//CSIVOL      DD   UNIT=3380,VOL=SER=valid,DISP=SHR
//SYSPRINT    DD   SYSOUT=A
//SYSIN       DD   *
              DEFINE CLUSTER(
                  NAME(SMPE.SMPCSI.CSI)
                  FREESPACE(10 5)
                  KEYS(24 0)
                  RECORDSIZE(24 143)
                  SHAREOPTIONS(2 3)
              VOLUMES(valid1)
              )
              DATA(
                  NAME(SMPE.SMPCSI.CSI.DATA)
                  CONTROLINTERVALSIZE(4096)
                  CYLINDERS(250 20)
              )
              INDEX(
                  NAME(SMPE.SMPCSI.CSI.INDEX)
                  CYLINDERS(5 3)
              )
              CATALOG(user.catalog)

/*
```

SMP/E: Qué es una SYSMOD ?

SMP/E puede instalar una gran variedad de actualizaciones del sistema, siempre que estén empaquetadas o armadas como una modificación del sistema o SYSMOD.

Una SYSMOD es el paquete real de elementos e información de control que SMP/E necesita para instalar y realizar un seguimiento de las modificaciones del sistema.

Las SYSMOD están compuestos por una combinación de elementos e información de control.

SMP/E: Qué es una SYSMOD ?

Se componen de dos partes, como se indica a continuación:

- Sentencias de control de modificación (MCS), designadas por ++ como los dos primeros caracteres, que indican a SMP/E:
- Qué elementos se están actualizando o reemplazando
- Cómo se relaciona el SYSMOD con el software del producto y otros SYSMOD
- Otra información específica de instalación
- Texto de modificación, que son los módulos de objeto, macros y otros elementos suministrados por la SYSMOD.

Existen cuatro categorías diferentes de SYSMOD, cada una de las cuales admite una tarea que quizás desee realizar:



SMP/E: tipos de SYSMOD

FUNCTION

Este tipo de SYSMOD introduce un nuevo producto, una nueva versión o ‘release’ de un producto; o funciones actualizadas para un producto existente en el sistema.

PTF (program temporary fix)

Una corrección temporal de programa (PTF) es una corrección proporcionada por IBM para un problema informado.

Están pensadas para instalarse en todos los entornos.

Las PTF se pueden utilizar como servicio preventivo para evitar ciertos problemas conocidos que pueden no haber aún aparecido en su sistema, o se pueden utilizar como servicio correctivo para solucionar problemas que ya haya encontrado.

La instalación de una PTF siempre debe ir precedida de la de una SYSMOD de función y, a menudo, también de otras PTF.

SMP/E: tipos de SYSMOD

APAR (Authorized Program Analysis Report)

Es un temporary fix para corregir o bypass un problema para quien reporto el problema. Un APAR puede no ser aplicable a su entorno.

La instalación de un APAR siempre debe estar precedida por una función SYSMOD y, a veces, de una PTF particular.

Es decir, un APAR está diseñado para instalarse en un nivel de servicio preventivo particular de un elemento.

SMP/E: tipos de SYSMOD

USERMOD

Este tipo de SYSMOD lo crea usted, ya sea para cambiar el código de IBM o para agregar funciones independientes al sistema.

La instalación de una USERMOD siempre debe estar precedida por una función SYSMOD, a veces de ciertas PTF, correcciones de APAR (fixes) u otros USERMOD.

SMP/E realiza un seguimiento de los niveles funcionales y de servicio de cada elemento y utiliza esta jerarquía SYSMOD para determinar aspectos como qué niveles funcionales y de servicio de un elemento se deben instalar y el orden correcto para instalar actualizaciones para los elementos.



SMP/E: Función SYSMOD

Función SYSMOD: Introducir un elemento en el sistema

Una forma de modificar el sistema es introducir nuevos elementos en él. Para lograr esto con SMP/E, puede instalar una función SYSMOD.

La función SYSMOD introduce un nuevo producto, una nueva versión o lanzamiento de un producto, o funciones actualizadas para un producto existente en el sistema.

Todos los demás tipos de SYSMOD dependen de la función SYSMOD, porque son todas modificaciones de los elementos introducidos originalmente por la función SYSMOD.

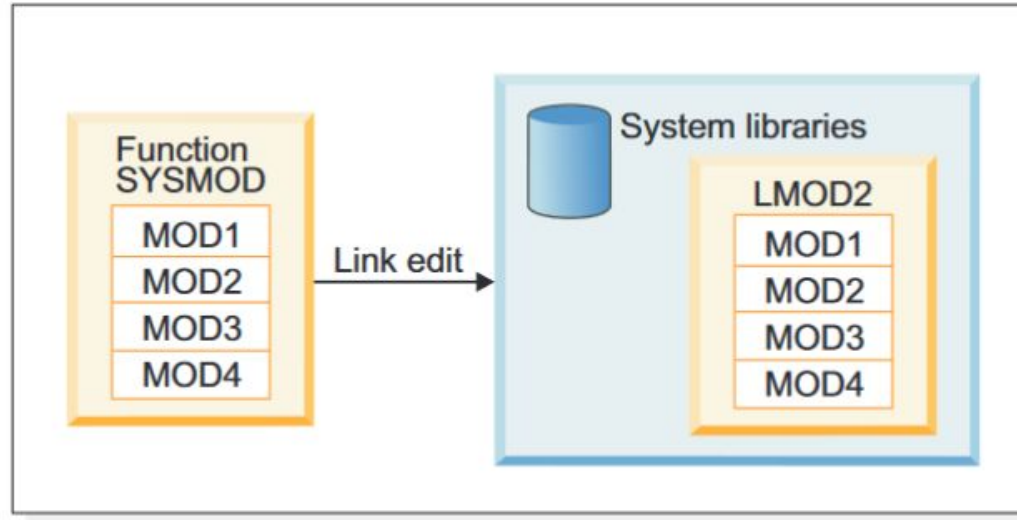
Al introducir una función SYSMOD, todos los componentes o el código del elemento se colocan en librerías o en system data sets.

Algunos ejemplos de estas bibliotecas son SYS1.LPALIB, SYS1.MIGLIB y SYS1.SVCLIB.

SMP/E: Función SYSMOD

La instalación de una función SYSMOD permite editar los módulos de objetos MOD1, MOD2, MOD3 y MOD4 para crear el módulo de carga LMOD2.

El código ejecutable creado en el módulo de carga LMOD2 se instala en las bibliotecas del sistema mediante la instalación de la función SYSMOD.



SMP/E: Función SYSMOD

Existen dos tipos de funciones SYSMOD:

- Una función base SYSMOD agrega o reemplaza una función de sistema completa. Ejemplos de funciones base son SMP/E y JES2.
- Una función dependiente SYSMOD proporciona una adición a una función de sistema existente. Se denomina dependiente porque su instalación depende de una función base que ya esté instalada. Ejemplos de funciones dependientes son las características de lenguaje (features) para SMP/E.

Tanto la función base SYSMOD como la función dependiente SYSMOD, se utilizan para introducir nuevos elementos en el sistema.

SMP/E: Función SYSMOD

Un ejemplo de una función SYSMOD simple que introduce cuatro elementos:

```
++FUNCTION(FUN0001)      /* SYSMOD type and identifier. */.  
++VER(Z038)              /* For MVS SREL */.  
++MOD(MOD1)  RELFILE(1)  /* Introduce this module */  
                  DISTLIB(AOSFB) /* in this distribution library. */.  
++MOD(MOD2)  RELFILE(1)  /* Introduce this module */  
                  DISTLIB(AOSFB) /* in this distribution library. */.  
++MOD(MOD3)  RELFILE(1)  /* Introduce this module */  
                  DISTLIB(AOSFB) /* in this distribution library. */.  
++MOD(MOD4)  RELFILE(1)  /* Introduce this module */  
                  DISTLIB(AOSFB) /* in this distribution library. */.
```

SMP/E: PTF SYSMOD

PTF SYSMOD: Prevención o solución de problemas (fixing) con un elemento

Cuando se descubre un problema con un elemento de software, IBM proporciona a sus clientes una solución probada (fix) para ese problema.

Este fix se presenta en forma de una solución temporal del programa (PTF).

Aunque es posible que no haya experimentado el problema, la PTF tiene como objetivo evitarlo, es recomendable instalar la PTF en su sistema.

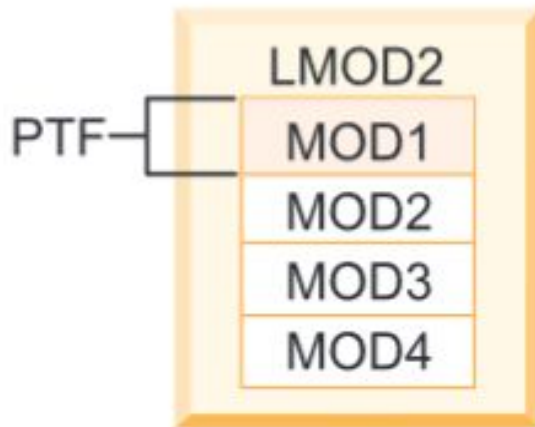
La PTF SYSMOD, se utiliza para instalar la PTF, evitando así la aparición de ese problema en su sistema.

Por lo general, las PTF están diseñadas para reemplazar o actualizar uno o más elementos completos de una función del sistema.

SMP/E: PTF SYSMOD

La próxima figura , muestra un módulo de carga instalado previamente, LMOD2. Para reemplazar el elemento MOD1, debe instalar un PTF SYSMOD que contenga el módulo MOD1.

Como parte de la instalación la PTF SYSMOD, SMP/E vuelve a relink LMOD2 para incluir la versión nueva y corregida de MOD1.



SMP/E: PTF SYSMOD

Las PTF SYSMOD, siempre dependen de la instalación de una función SYSMOD.

En algunos casos, algunas PTF SYSMOD también pueden depender de la instalación de otras PTF SYSMOD.

Estas dependencias se denominan pre-requisitos.

PTFs, APAR y USERMOD SYSMODs todas tienen función SYSMOD como pre-requisito. Además de sus dependencias en la función SYSMOD:

- Las PTF SYSMODs pueden depender de otras PPTF SYSMODs.
- Los APAR SYSMODs pueden depender de otras PTF SYSMODs y de otros APAR SYSMODs.
- Las USERMOD SYSMODs pueden depender de otras PTF SYSMODs, APAR SYSMODs y otras USERMOD SYSMODs.

SMP/E: PTF SYSMOD

Ejemplo de una simple PTF SYSMOD:

```
++PTF(PTF0001)          /* SYSMOD type and identifier. */.  
++VER(Z038) FMID(FUN0001) /* Apply to this product. */.  
++MOD(MOD1)             /* Replace this module */.  
                        DISTLIB(AOSFB) /* in this distribution library. */.  
  
...  
... object code for module  
...
```

A veces, una PTF o incluso un APAR depende de otras PTF SYSMOD, llamadas correquisitos.

Considere la complejidad de estas dependencias: cuando multiplica esa complejidad por cientos de módulos de carga, en docenas de bibliotecas, se hace evidente la necesidad de una herramienta como SMP/E.



SMP/E: APAR SYSMOD

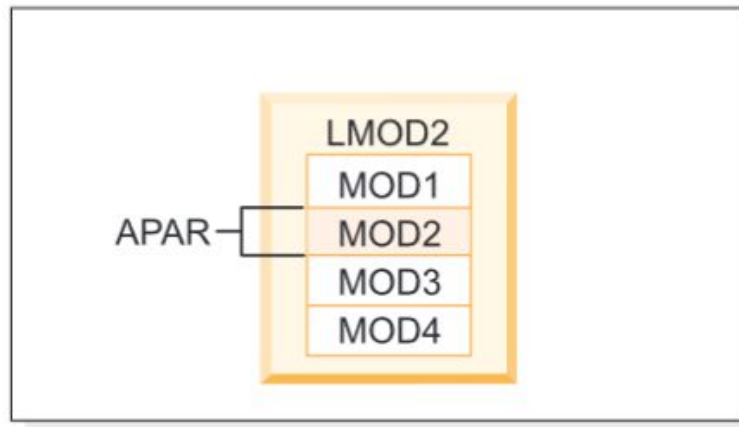
APAR SYSMOD: Fixing problemas con un elemento

Es posible que, en ocasiones, sea necesario corregir un problema grave que se produce en su sistema antes de que una PTF esté listo para su distribución.

Ante esta situación, IBM le proporciona un APAR.

Un APAR es una corrección diseñada para corregir rápidamente un área específica de un elemento o reemplazar un elemento con error.

Se instala un APAR SYSMOD para implementar el fix, actualizando así el elemento incorrecto. En la próxima figura, la sección sombreada muestra un área de MOD2 que contiene un error:



SMP/E: APAR SYSMOD

Al procesar el APAR SYSMOD proporciona una modificación para el módulo objeto MOD2.

Durante la instalación del APAR SYSMOD, MOD2 se actualiza (update y corrige) el módulo de carga LMOD2.

Ejemplo de un simple APAR SYSMOD:

```
++APAR(APAR001)          /* SYSMOD type and identifier. */.  
++VER(Z038) FMID(FUN0001) /* Apply to this product      */.  
                  PRE(UZ00004) /* at this service level. */.  
++ZAP(MOD2)              /* Update this module        */.  
                  DISTLIB(AOSFB) /* in this distribution library. */.  
...  
... zap control statements  
...
```

Un APAR SYSMOD siempre tiene la instalación de una función SYSMOD como prerequisite y también puede depender de la instalación de otras PTF o APAR SYSMODs.

PTF, APAR y USERMOD SYSMODs tienen una función SYSMOD como prerequisite previo.

Best practice: realizar un seguimiento de los elementos y las modificaciones del sistema

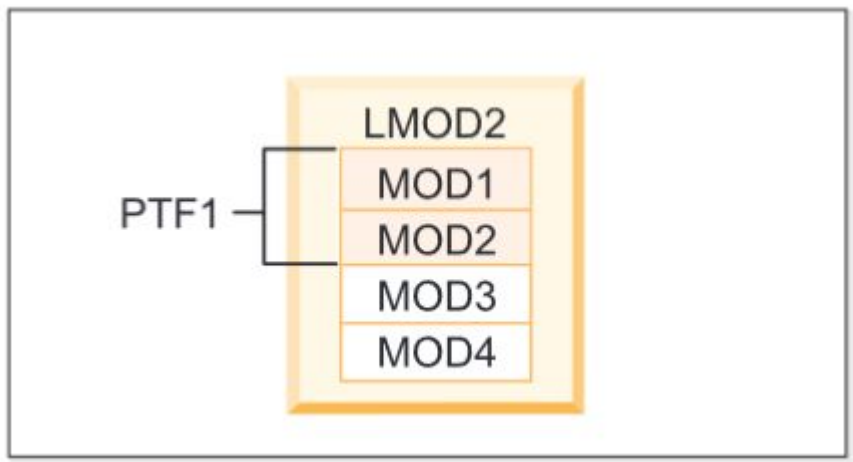
La importancia de realizar un seguimiento de los elementos del sistema y sus modificaciones se hace evidente al examinar el proceso de mantenimiento de z/OS.

A menudo, una PTF contiene múltiples reemplazos de elementos.

En próximo ejemplo se muestra en la figura, PTF1 contiene reemplazos para dos módulos, MOD1 y MOD2.

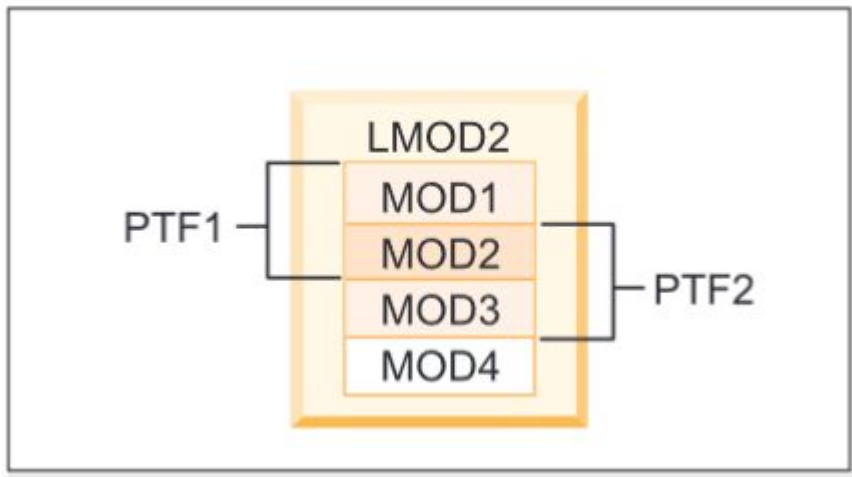
Best practice: realizar un seguimiento de los elementos y las modificaciones del sistema

Aunque el módulo de carga LMOD2 contiene cuatro módulos, solo dos de esos módulos se están reemplazando:



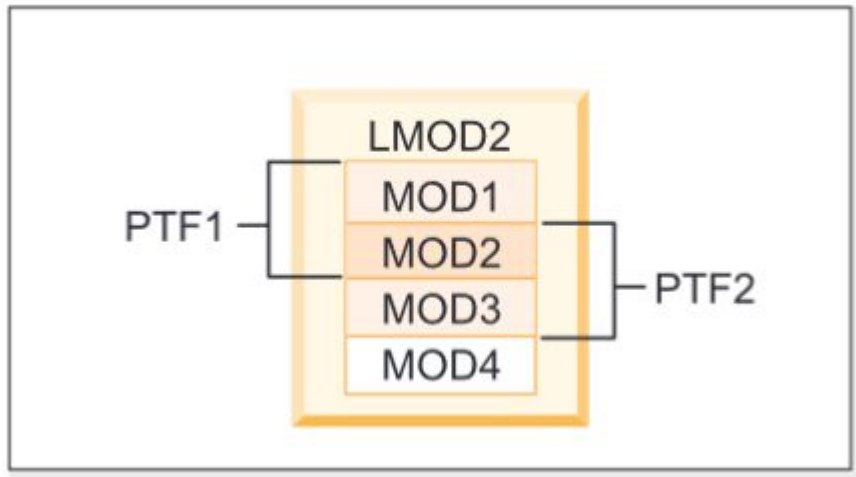
Best practice: realizar un seguimiento de los elementos y las modificaciones del sistema

Pero, ¿qué sucede si una segunda PTF reemplaza parte del código en un módulo que fue reemplazado por PTF1 ?, como se muestra:



Best practice: realizar un seguimiento de los elementos y las modificaciones del sistema

Pero, ¿qué sucede si una segunda PTF reemplaza parte del código en un módulo que fue reemplazado por PTF1 ?, como se muestra:



En este ejemplo, PTF2 contiene reemplazos para MOD2 y MOD3.

Para que MOD1, MOD2 y MOD3 interactúen correctamente, PTF1 debe instalarse antes que PTF2.

Esto se debe a que MOD3 suministrado en PTF2 puede depender de que la versión PTF1 de MOD1 esté presente.

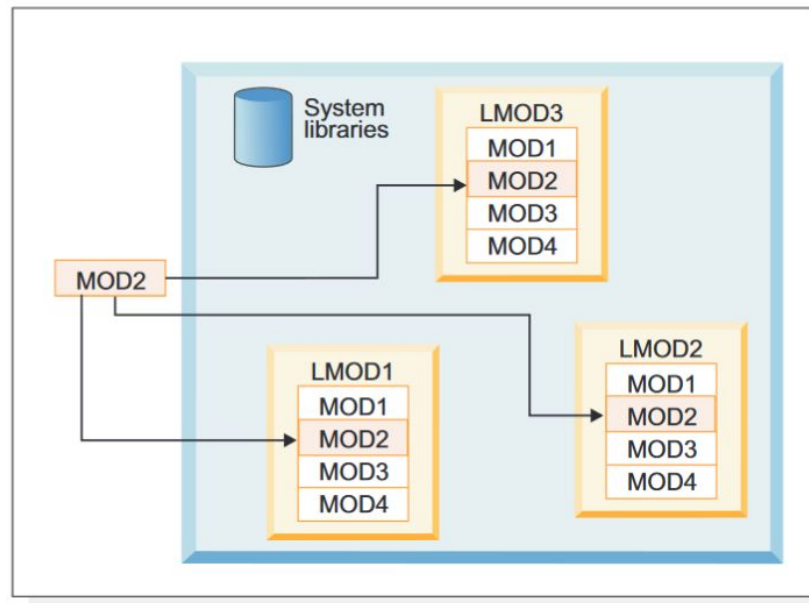
Esta dependencia constituye un requisito previo.

Best practice: realizar un seguimiento de los elementos y las modificaciones del sistema

Los requisitos previos de SYSMOD se identifican en la parte de declaraciones de control de modificación (MCS) del paquete SYSMOD.

Además de realizar un seguimiento de los requisitos previos, existe otra razón importante para realizar un seguimiento de los elementos del sistema.

El mismo módulo suele ser parte de muchos módulos de carga diferentes, como se muestra:

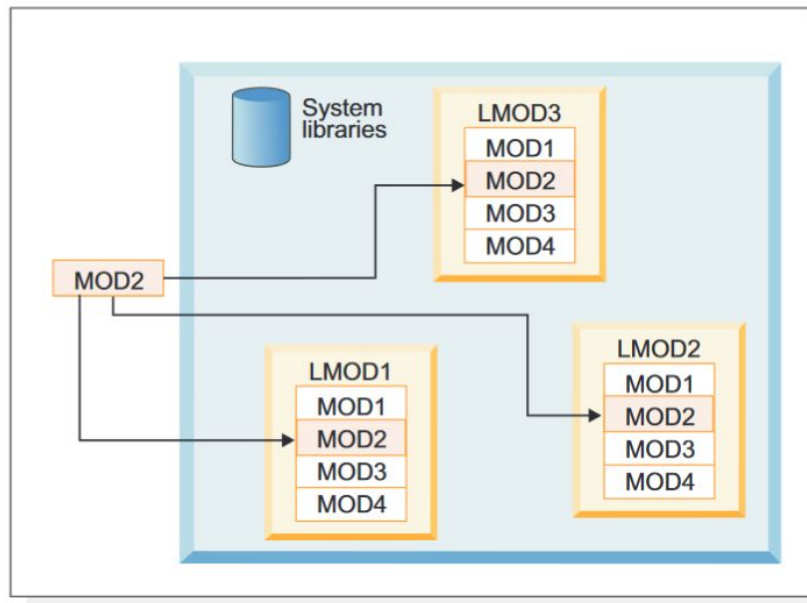


Best practice: realizar un seguimiento de los elementos y las modificaciones del sistema

El mismo módulo MOD2 está presente en LMOD1, LMOD2 y LMOD3.

Cuando se introduce una PTF que reemplaza al elemento MOD2, ese módulo debe ser reemplazado en todos los módulos de carga en los que existe.

Por lo tanto, es imperativo un seguimiento de todos los módulos de carga y los módulos que lo contienen.



SMP/E: Seguimiento y control de requisitos

Para realizar un seguimiento y controlar los elementos correctamente, todos los elementos y sus modificaciones y actualizaciones deben estar claramente identificados en SMP/E.

SMP/E se basa en identificadores de modificación para lograr esto.

Hay tres identificadores de modificación asociados con cada elemento:

- Los FMID identifican la función SYSMOD, que introduce el elemento en el sistema.
- Los RMID (identificadores de modificación de reemplazo) identifican la última SYSMOD (en la mayoría de los casos una PTF SYSMOD) que reemplaza un elemento.
- Los UMID (identificadores de modificación de actualización) identifican a la SYSMOD que actualiza un elemento desde que se reemplazó por última vez.

SMP/E utiliza estos identificadores de modificación para realizar un seguimiento de todos los SYSMOD instalados en el sistema.

Esto garantiza que se instalen en la secuencia adecuada.

SMP/E: instalar elementos o servicios en z/OS

Este resumen del procesamiento de SMP/E ilustra cómo se instalan los elementos de z/OS o las actualizaciones de servicio.

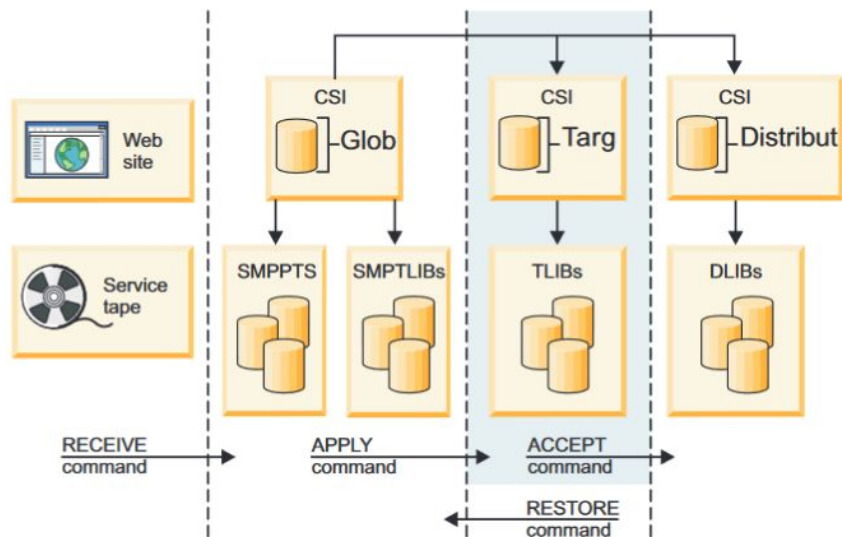
SMP/E proporciona un excelente inventario y referencia cruzada de dependencias de software, y un mecanismo para instalar nuevos productos o aplicar mantenimiento del software.

Estas capacidades garantizan la integridad del sistema en general.

La mecánica de uso de SMP/E para mantener un sistema puede ser bastante simple. El funcionamiento del SMP/E y el conocimiento necesario aplicar fixes (correcciones) y productos de software para SMP/E es significativamente complejo.

SMP/E: instalar elementos o servicios en z/OS

IBM proporciona instrucciones para que los programadores de sistemas las sigan con todas las correcciones (fixes), actualizaciones de software y productos bsjo SMP/E.



SMP/E: instalar elementos o servicios en z/OS

Ejemplo de un JCL, que podría usarse para automatizar el proceso desde la recepción de actualizaciones (receive), hasta su aplicación (apply) a un sistema de destino (target):

```
//SMPJOB    JOB  'accounting info',MSGLEVEL=(1,1)
//SMPSTEP   EXEC  SMPPROC
//SMPPTFIN  DD    ...
/* points to file or data set containing SYSMODs to be received */
//SMPHOLD   DD    ...
/* points to file or data set containing HOLDDATA to be received */
//SMPTLIB   DD    UNIT=3380,VOL=SER=TLIB01
//SMPCNTL   DD    *
SET         BDY(GLOBAL)          /* Set to global zone      */.
RECEIVE     SYSMOD                /* Receive SYSMODs        */.
            HOLDDATA              /* Receive HOLDDATA       */.
            SOURCEID(MYPTFS)      /* Assign a source ID     */.
            /*                      */.
LIST        MCS                   /* List the cover letters  */.
            SOURCEID(MYPTFS)      /* for the SYSMODs        */.
            /*                      */.
SET         BDY(TARGET1)          /* Set to target zone     */.
APPLY       SOURCEID(MYPTFS)      /* Apply the SYSMODs      */.
            /*                      */.
LIST        LOG                   /* List the target zone log */.
/*
```

SMP/E: Comandos SMP/E

El comando RECEIVE le permite incorporar una SYSMOD que está fuera de SMP/E y prepararlo para el dominio de la biblioteca SMP/E, que comienza a construir las entradas CSI que lo describen.

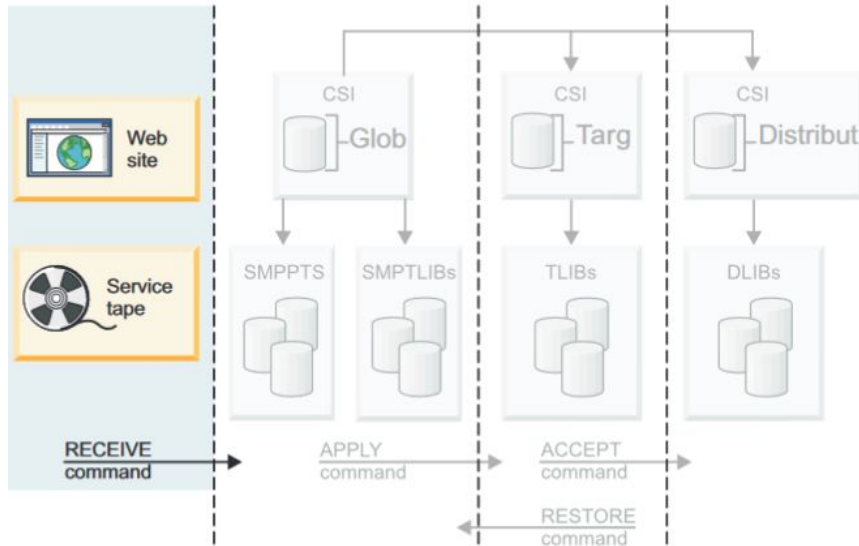
Esta preparación permite consultarlos en procesos posteriores.
Más recientemente, la fuente puede ser electrónica desde un sitio web.

SMP/E: Comandos SMP/E

El proceso RECEIVE realiza varias tareas que se muestran:

Estas tareas incluyen:

- Construir entradas en la Zona Global para describir la SYSMOD.
- Asegurarse de que el SYSMOD sea válido, como la sintaxis para las sentencias de control de modificación (MCS) asociadas a los productos instalados en el CSI.
- Instalar la SYSMOD en las bibliotecas.
- Evaluar HOLDDATA para asegurarse de que no se introduzcan errores.



SMP/E: Comandos SMP/E

Durante el procesamiento de RECEIVE, el MCS para cada SYSMOD se copia a un área de almacenamiento temporal de SMP/E denominado SMPPTS, que contiene el reemplazo o actualización de elementos para ese SYSMOD.

También hay archivos RELFILE que representa un package de los elementos, en archivos relativos que están separados de los MCS, que son utilizados principalmente por la/s función de SYSMOD.

Los archivos relativos (relfile) se almacenan en otra área de almacenamiento temporal denominada SMPTLIB.

SMP/E actualiza la zona global con información sobre las SYSMOD que han sido recibidas.

Durante el mantenimiento del sistema, debe instalar el servicio y procesar las HOLDATA relacionados.

SMP/E: Ejemplos de RECEIVE

Para recibir únicamente las HOLDDATA que puedan requerir un manejo especial o que presenten errores, se utiliza este comando:

```
SET BDY(GLOBAL).  
RECEIVE HOLDDATA.
```

Para recibir únicamente las SYSMOD para la instalación en la zona global, se utiliza este comando:

```
SET BDY(GLOBAL).  
RECEIVE SYSMODS.
```


SMP/E: El comando APPLY

El comando APPLY especifica cuáles de las SYSMOD recibidos se deben seleccionar para su instalación en las bibliotecas de destino (targets).

SMP/E también garantiza que todas los demás SYSMOD requeridos (prerrequisitos) se hayan instalado o se estén instalando simultáneamente, así como en la secuencia adecuada.

El 'source' de los elementos es SMPTLIB, el SMPPTS o las bibliotecas indirectas, según cómo se hayan empaquetado.

SMP/E: El comando APPLY

Esta fase del proceso SMP/E implica lo siguiente:

- Ejecutar la utilidad adecuada para instalar el SYSMOD en la biblioteca de destino, según el tipo de texto de entrada suministrado y el módulo de destino que se va a cambiar.
- Asegurarse de que la relación de la nueva SYSMOD con otras SYSMOD en la zona de destino sea correcta.
- Se modifica el CSI mostrando los módulos actualizados.



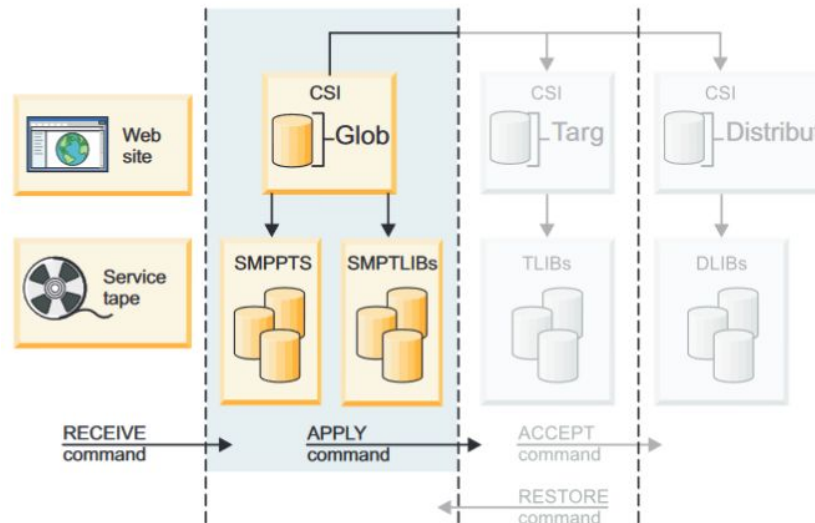
SMP/E: El comando APPLY

El comando APPLY actualiza las bibliotecas del sistema y debe utilizarse con cuidado en un sistema de producción en vivo.

Se recomienda que utilice inicialmente una copia de las bibliotecas y zonas (zones) de destino de producción.

La zona de destino refleja el contenido de las bibliotecas de destino.

Por lo tanto, una vez que se complete la utilidad y se actualice la zona, reflejará con precisión el estado de esas bibliotecas.



SMP/E: El comando APPLY

El proceso APPLY, mostrado, es donde la zona de destino se actualiza con precisión:

- Todas las entradas SYSMOD en la Zona Global se actualizan para reflejar que SYSMOD se ha aplicado a la zona de destino (taget zone).
- La zona de destino refleja con precisión cada entrada SYSMOD aplicada. Las entradas de elementos (como MOD y LMOD) también se crean en la zona de destino.
- Las entradas BACKUP se crean en el SMPSCDS para que una SYSMOD se pueda restaurar, si es necesario.

De manera similar al proceso RECEIVE, el comando APPLY tiene muchos operandos diferentes para brindar flexibilidad para seleccionar las SYSMOD que desea ver para su instalación en las bibliotecas de destino, y proporciona una variedad de 'outputs'. Las directivas utilizadas instruyen a SMP/E lo que desea instalar.

SMP/E: Ejemplos de comandos APPLY

Para instalar solo las PTF SYSMOD, se ingresa un comando como el siguiente:

```
SET BDY(ZOSTGT1).  
APPLY PTFS.
```

Para seleccionar las PTF SYSMOD, se nombran en las directivas, por ejemplo:

```
SET BDY(ZOSTGT1).  
APPLY SELECT(UZ00001, UZ00002).
```

En ocasiones, es posible que desee instalar solo correcciones correctivas (APAR) o modificaciones de usuario (USERMOD) en la biblioteca de destino, por ejemplo:

```
SET BDY(ZOSTGT1).  
APPLY APARS USERMODS.
```

SMP/E: Comando APPLY con el operando CHECK

Es posible que haya ocasiones en las que desee ver qué SYSMOD están incluidos antes de ser instaladas.

Es posible hacerlo, incluyendo el operando CHECK, con comandos como los siguientes:

```
SET BDY(MVSTGT1).
```

```
APPLY PTFS APARS FORFMID(HOP1) GROUPEXTEND CHECK.
```

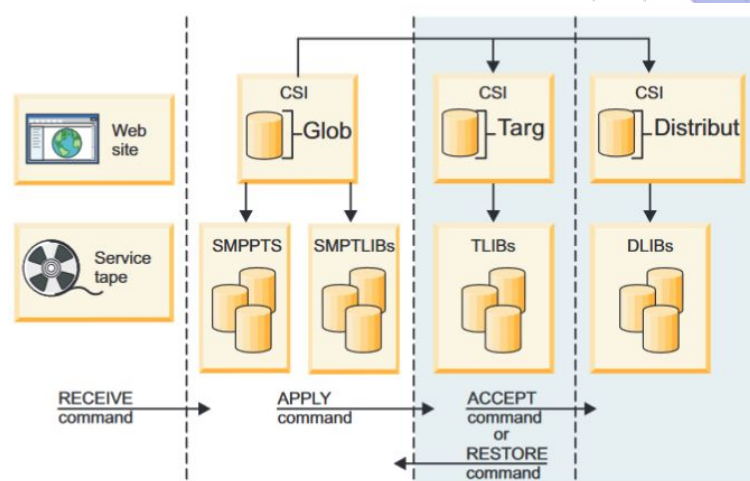
SMP/E: El comando ACCEPT

Cuando se instala una SYSMOD en su biblioteca de destino y se lo ha probado, se acepta el cambio a través del comando ACCEPT.

Este paso toma los SYSMOD seleccionados y los instala en las bibliotecas de distribución asociadas.

En el comando ACCEPT, se especifican operandos para indicar cuáles de las SYSMOD recibidos se deben seleccionar para la instalación.

Durante esta fase, SMP/E también garantiza que se seleccione el nivel funcional correcto de cada elemento.



SMP/E: El comando ACCEPT

El comando ACCEPT realiza las siguientes tareas, que son mostradas:

- Actualiza las entradas de CSI con los elementos de destino en la zona de distribución.
- Reconstruye o crea los elementos de destino en las bibliotecas de distribución utilizando el contenido de SYSMOD como entrada.
- Verifica las entradas CSI de la zona de destino para los módulos y SYSMOD afectados, asegurándose de que sean consistentes con el contenido de la biblioteca.
- Realiza el mantenimiento (housekeeping) de los elementos obsoletos o caducados.

El procesamiento de ACCEPT elimina las entradas CSI de la zona global, miembros del PTS y de SMPTLIB, de aquellas SYSMOD afectados. Por ejemplo, ACCEPT elimina las entradas SYSMOD de la zona global y las sentencias MCS en el conjunto de datos SMPPTS para aquellos SYSMOD que han sido aceptados en la zona de distribución.

SMP/E: Ejemplos de ACCEPT

Para instalar PTF SYSMODS seleccionando unas en particular:

```
SET BDY(ZOSDLB1).  
ACCEPT SELECT(UZ00001,UZ00002).
```

SMP/E: Ejemplos de ACCEPT

Hay situaciones en las que es posible que se desee actualizar un producto en particular con todos los SYSMOD:

```
SET BDY(ZOSDLB1).  
ACCEPT PTFS FORFMID(H28W500).
```

-O-

```
SET BDY(ZOSDLB1).  
ACCEPT FORFMID(H28W500).
```

En estos dos ejemplos, SMP/E acepta todos las PTF aplicables para el producto cuyo FMID es H28W500 ubicado en la zona de distribución ZOSDLB1.

SMP/E: ACCEPT para prerequisites de SYSMOD.

Al instalar una SYSMOD, es posible que no se sepa si tiene requisitos previos (a veces, se mantiene un ERROR SYSMOD).

En estas situaciones, puede indicarle a SMP/E que verifique si hay una SYSMOD equivalente (o que lo reemplace) disponible especificando el operando Groupextend:

```
SET BDY(ZOSDLB1).  
ACCEPT PTFS FORFMID(H28W500) GROUPEXTEND.
```

SMP/E: ACCEPT reporting.

Cuando se complete esta última fase, los siguientes informes ayudarán a evaluar los resultados:

- Informe de estado de SYSMOD: proporciona un resumen del procesamiento que se llevó a cabo para cada SYSMOD, en función de los operandos que especificados en el comando ACCEPT.
- Informe de resumen de elementos (summary): proporciona una descripción detallada de cada elemento afectado por el procesamiento de ACCEPT y en qué bibliotecas residen.
- Informe de resumen de SYSMOD causante: proporciona una lista de SYSMOD que provocaron que otros SYSMOD fallaran y describe los errores que se deben corregir para que se procesen correctamente.
- Informe de alocacion de archivos: proporciona una lista de los data sets utilizados para el procesamiento del ACCEPT y proporciona información acerca de estos data sets.