Abstract:
This session will cover the essential technical information for upgrading to z/OS 3.1. The focus in the first half is preparing your current system for upgrading from either z/OS V2.4 or z/OS V2.5. The system requirements to run and how to prepare your system for the upgrade are discussed. The second half covers the technical upgrade details.

The general availability date for z/OS 3.1 is September 29, 2023.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT").

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

2

Upgrading to z/OS 3.1: Planning Topics

- **Content of z/OS 3.1**
  - Added, Changed, and Withdrawn Elements and Features
- **z/OS Ordering and Deliverables**
  - Products Related to z/OS
- **z/OS Policies**
  - z/OS End of Service dates
  - z/OS Coexistence -Upgrade-Fallback
- **Planning for z/OS 3.1:**
  - Ensuring System Requirements are Satisfied
  - Coexistence System Requirements
  - Using z/OSMF Workflow for z/OS Release Upgrade
  - z/OSMF Driving System Requirement
  - Some Upgrade Actions You Can Do NOW
  - Programmatic Verification of Upgrade Actions

3

Scope

- **Focus on z/OS upgrade, not HW upgrade.**
- If upgrading to a new server level, see:
  - *for z14: z/OS z14 Workflow (or Upgrade action in z/OS V2.5 Upgrade Workflow) for "Upgrade to an IBM z14 server".*
  - for z15: **z/OS z15 Workflow** (or Upgrade action in *z/OS V2.5 Upgrade Workflow*) for "Upgrade to an IBM z15 server".
  - **for z16 A01 and A02: *z/OS z16 Workflow***
    - Found with FIXCAT IBM.Device.Server.z16A02 - 3932.RequiredService
    - Note: All HW upgrade actions are found in the HW Upgrade Workflow now, as they have been removed from the *z/OS 3.1 Upgrade Workflow.*

4

IBM Training
IBM courses are available for z/OS. For schedules and enrollment on the world wide web, IBM Global Campus URL: http://www.ibm.com/training/ .

z/OS 3.1 Elements (changing* in z/OS V2.5 and V3.1)

- BCP
  - Program Binder
  - Capacity Provisioning Manager
  - BCP Support for Unicode
  - Web Enablement Toolkit
- Common Information Model (CIM)
- Communications Server
- Cryptographic Services:
  - ICSF (FMID HCR77E0)
  - PKI Services
  - System SSL
- DFSMSdfp
- **DFSMSdss**
- **DFSMShsm**
- **DFSMSrmm**
- **DFSORT**
- HCD
- **HCM**
- *Future Function* (related to IBM Documentation for z/OS)
- IBM Tivoli Directory Server
- IBM Z Deep Neural Network (zDNN)
- **IBM z/OS Change Tracker** NEW!
- IBM z/OS Management Facility
- **IBM z/OS Workload Interaction Correlator (zWIC)**
- **Infoprint Server**
- Integrated Security Services:
  - Network Authentication Service
- ISPF
- JES2
- Language Environment
- Network File System
- **RMF**
- **RUCSA**
- **SDSF**
- **Security Server– RACF**
- TSO/E
- XML Toolkit (V1.11 level) NEW!
- **z/OS Advanced Data Gatherer**
- z/OS Data Gatherer
- z/OS File System (zFS)
- z/OS Font Collection
- z/OS OpenSSH
- ***z/OS Security Level 3:***
  - ***Communications Server***
  - ***IBM Tivoli Directory Server Security Level 3***
  - ***Network Authentication Service Level 3***
  - ***System SSL Level 3***
- z/OS UNIX

KEY:
- * **"Changing" means the FMID is changing.** Remember, PTFs might have added new functions on FMIDs that are not changing.
- **Black** (not in bold) are base elements
- **Green** (also in **bold**) are optional priced features
- ***Brown*** (also in ***italics***) are optional unpriced features with export controls
- → This element changed in z/OS V2.5
- → This element changed in z/OS 3.1
- All other elements not listed have not changed since z/OS V2.4.
- **New in V2.5: Data Gather, Adv Data Gatherer, IBM z/OS Change Tracker**
- **New in 3.1: XML Toolkit**

5

Important ordering changes for z/OS 3.1

→ ***z/OS Security Level 3:*** **Do not forget to order if you need it!**
- → ***Communications Server Security Level 3***
- → ***IBM Tivoli Directory Server Security Level 3***
- → ***Network Authentication Service Level 3***
- → ***SSL Level 3***

- Note that the Communications Server Security Level 3 optional unpriced export controlled feature is now part of the z/OS Security Level 3 feature.

→ ***IBM JES3, and BDT priced features***
- These priced features (as well as the BDT base element) have been removed from z/OS 3.1.
- JES2 will be installed into the z/OSMF portable software instance base z/OS SMP/E zone, and is not allowed to be removed.

→ ***DFSMStvs priced feature***
- This priced feature is now part of the base, and is entitled to use as part o the z/OS 3.1 base.

- **XML Toolkit** has been added as a base element
  - This had been the program product , 5655-J51, V1.11. Thus, this program product is not orderable with z/OS 3.1.
- **z/OS Alternate Base** has been removed
  - This had been provided for alternate usages of Communication Server, which are no longer applicable.
- ***Reminders from prior z/OS releases, continuing in z/OS 3.1:***
  - Enabling z/OS Advanced Data Gatherer feature also implicitly enables the z/OS Workload Interaction Correlator feature.
  - Ordering the RMF feature causes the z/OS Advanced Data Gatherer feature to be enabled.

6

z/OS Elements and Features

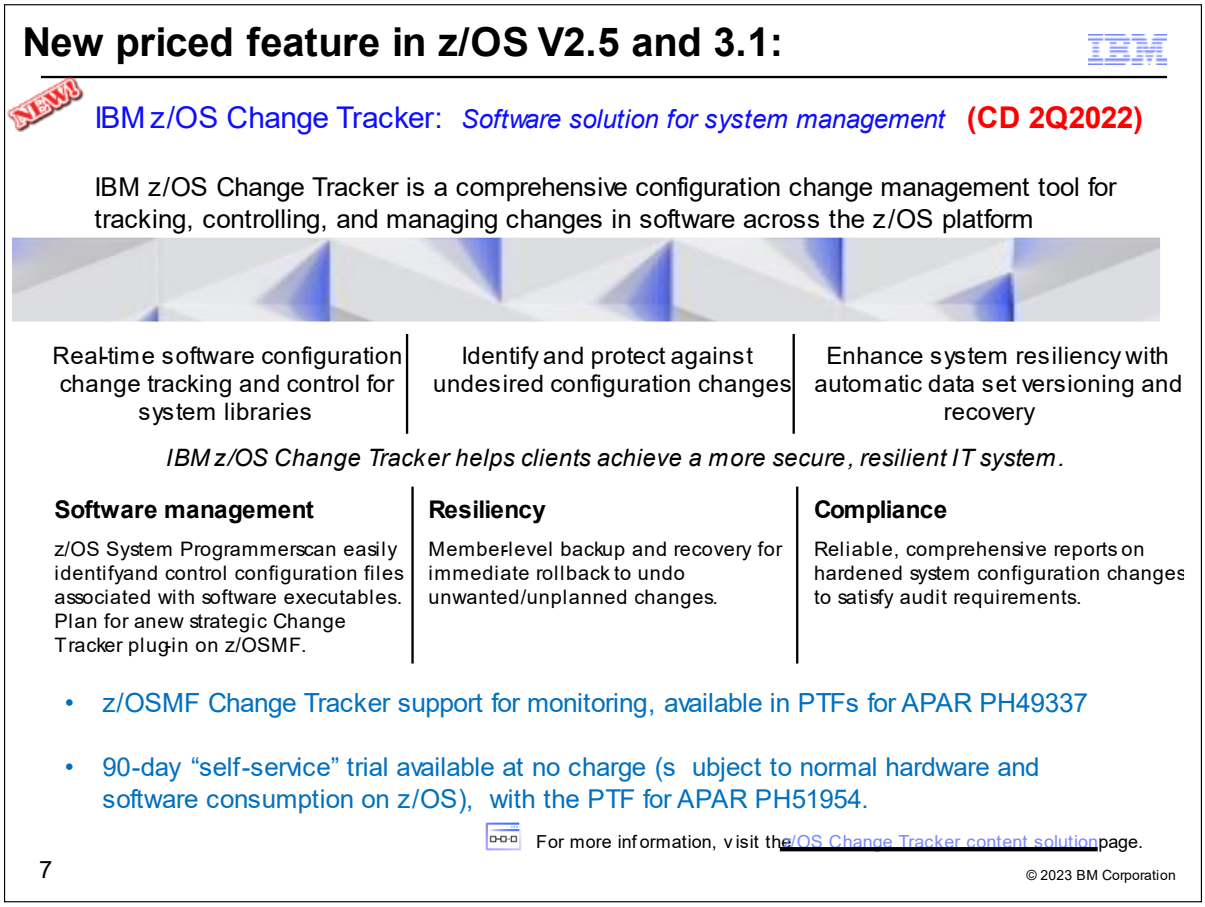
z/OS consists of base elements and optional features:

- The ***base elements*** (or simply *elements*) deliver essential operating system functions. When you order z/OS, you receive all of the base elements.
- The ***optional features*** (or simply *features)* are orderable with z/OS and provide additional operating system functions. Optional features are unpriced or priced:
  - *Unpriced features* are shipped to you **only if** you order them. If you plan to use any unpriced features, IBM recommends that you order them when you order your base elements. You must not wait until the next release becomes available. Once a release's base elements are no longer orderable, usually neither are its unpriced features.
  - *Priced features* are **always** shipped to you. When IBM packages your order, we *enable* the priced features that you ordered. These features are ready to use after you install z/OS (and customize them as needed). We *disable* the priced features that you did not order. Although they are installed on your system, you cannot use them. Later on, if you decide to use them, you notify IBM and you enable them dynamically (which is known as *dynamic enablement*). You dynamically enable by updating parmlib member IFAPRD*xx* and you notify IBM by contacting your IBM representative.

Elements and features may be exclusive or nonexclusive:

- An element or feature is called *exclusive* to z/OS if it exists only within z/OS (not also as a separately orderable, or stand-alone, product) and if future functional enhancements will occur only within z/OS.
- An element or feature is called *nonexclusive* if it exists both (1) within z/OS and (2) as a stand-alone product.

Listed in the slide above are the changing FMIDs (elements) within z/OS 3.1 since z/OS V2.4.



IBM z/OS Change Tracker

On z/OS V2.5, this new priced feature has a simple control interface to identify, manage, and audit configuration files, IBM z/OS Change Tracker can help clients with aspects of the change configuration management experience.

Element and Functions Withdrawn from z/OS V2.5

HFS	Base Element support: Use zFS instead. Use z/OS utilities to help with the conversion of the entire file system hierarchy.	**z/OS V2.5**
ISPF Workstation Agent (WSA), also known as ISPF Client/Server component	Base element support: Use more current transfer solution, such as Zowe Data Set Explorer or ftp.	**z/OS V2.5**
VTAM Common Management Information Protocol (CMIP)	Base element support: is an API that enables a management application program to gather various types of SNA topology data from a CMIP application called the *topology agent* that runs within VTAM. IBM recommends using the SNA network monitoring network management interface (NMI) to monitor SNA Enterprise Extender and High Performance Routing data.	**z/OS V2.5**
Direct invocation of System SSL APIs for TLS/SSL by TN3270 Telnet server, FTP server, and Digital Certificate Access Server (DCAS)	Base element support, the only TLS/SSL protection option for these servers is AT-TLS. Convert these servers to use AT-TLS.	**z/OS V2.5**

8

Element and Functions Withdrawn from z/OS V2.5

WLM service coefficients specification (on Service Definition Details page)	Base element support: Use recommended values of CPU=1, SRV=1, MSO=0, and IOC=0 which will be the default values. Adjust now.	**z/OS V2.5**
EIM, OCSF, and all of its plug-ins such as OCEP and PKITP.	Base element support: Use other applications such as ICSF and System SSL for comparable functionality.	**z/OS V2.5**
Network Configuration Assistant (NCA) z/OSMF plug-in for policy data import function, for importing existing Policy Agent configuration files.	Base element support: Import of policy configuration files for AT-TLS, IPSec, PBR, and IDS technologies. Import of TCP/IP profile into NCA is not affected.	**z/OS V2.5**
z/OSMF "classic" tree mode interface	Base element support: Use the more desktop-style interface which has more capabilities.	**z/OS V2.5**
DFSMSrmm Web Services removal	Priced feature function: RMM API to access the RMM control data set to obtain information about RMM managed resources using either a high-level or assembler language.	**z/OS V2.5**

9

Withdrawn in z/OS V2.5 (last delivered in z/OS V2.4)

This section lists items that were withdrawn in z/OS V2.,5. You should take this into account if you are upgrading from z/OS V2.4 to z/OS 3.1. The removal of these functions may have upgrade actions which you can perform now, in preparation for z/OS 3.1.

- z/OS 2.4 is the last release of the operating system to support the HFS (Hierarchical File System) data structure used by the z/OS UNIX environment. IBM has provided equivalent if not superior functionality with the z/OS File System (zFS). Customers should migrate from HFS to zFS using the utilities provided in the operating system to convert their entire file system hierarchy.
- z/OS V2.4 is the last release to support the ISPF Workstation Agent (WSA), also known as the ISPF Client/Server Component. WSA is an application that runs on your local workstation and maintains a connection between the workstation and the ISPF host. It is primarily used to transfer files between the workstation and the host. IBM recommends using more current file transfer solutions such as those provided by the Zowe Dataset Explorer, z/OS FTP, and similar file transfer mechanisms. These solutions have more capabilities, including the ability to provide secure communications.
- z/OS V2.4 is the last release to support the VTAM Common Management Information Protocol (CMIP). CMIP services is an API that enables a management application program to gather various types of SNA topology data from a CMIP application called the *topology agent* that runs within VTAM. IBM recommends using the SNA network monitoring network management interface (NMI) to monitor SNA Enterprise Extender and High Performance Routing data.
- z/OS V2.4 is the last release in which the z/OS TN3270E Telnet server, FTP server, and Digital Certificate Access Server (DCAS) will support direct invocation of System SSL APIs for TLS/SSL protection. In the future, the only TLS/SSL protection option for these servers will be Application Transparent Transport Layer Security (AT-TLS). The direct System SSL support in each of these components is functionally outdated and only supports TLS protocols up through TLSv1.1. IBM recommends converting your TN3270E Telnet, FTP server, and DCAS configurations to use AT-TLS, which supports the latest System SSL features, including the TLSv1.2 and TLSv1.3 protocols and related cipher suites. Note that while native TLS/SSL support for z/OS FTP client is not being withdrawn at this time, no future enhancements are planned for that support. IBM recommends using ATTLS to secure FTP client traffic.
- z/OS V2.4 is the last release of z/OS to allow specifying service coefficients in the Workload Manager (WLM) service definition on the Service Definition Details page. The IBM recommended values are CPU=1, SRB=1, MSO=0, and IOC=0, which will be the default values in a later release. IBM recommends that you adjust your service coefficients before upgrading to a later release.
- z/OS V2.4 is the last release to support EIM (Enterprise Identity Mapping) and OCSF (Open Cryptographic Services Facility), and all of its plug-ins, such as OCEP (Open Cryptographic Enhanced Plug-ins) and PKITP (PKI Services Trust Policy). These components have not been widely utilized nor enhanced for several releases of z/OS. IBM recommends using other applications such as ICSF (Integrated Cryptographic Services Facility) and System SSL for comparable functionality.
- z/OS V2.4 is the last release that the Network Configuration Assistant (NCA) z/OS MF plug-in supports the policy data import function, which allows you to import existing Policy Agent configuration files into the Network Configuration Assistant. After z/OS V2.4, import of policy configuration files will no longer be supported for AT-TLS, IPSec, PBR, and IDS technologies. Import of TCP/IP profiles into NCA is not affected.
- z/OS V2.4 is the last release to support the z/OS MF classic-style user interface (the tree mode interface) and in future releases will only support the desktop-style user interface. The z/OS MF desktop-style user interface supports all the functions that the traditional tree mode interface does, and provides a more modernized and personalized UI, by displaying the z/OS MF tasks in a desktop style with task icons, taskbar, and other desktop elements that can be user tailored, which allows users to interact with z/OS using a familiar interface that is similar to other operating environments. The desktop UI also has more capabilities, such as the ability to search for data set names, quickly locate a task, group tasks in a folder, and perform similar actions.
- z/OS V2.4 is the last release of z/OS to support DFSMSrmm Web Services. Today, RMM provides support for remote JavaTM applications to connect to the RMM application programming interface (API) running on a z/OS system over the internet via a package that is deployed on a web server such as z/OS WebSphere(R) Application Server or Apache Tomcat. Use of the RMM API, which accesses the RMM control data set to obtain information about RMM managed resources, would still be available to applications using either high-level or assembler languages.

To determine if you are using RMM's Web Services support, verify if you have the following packages deployed: /usr/lpp/dfsms/rmm/rmmapi.ear for IBM WebSphere and /usr/lpp/dfsms/rmm/rmmapitc.war for Apache Tomcat. This can also be checked by listing the deployed web applications in the Tomcat Web application Manager or on the WebSphere Integrated Solutions console.

Functions Withdrawn from z/OS 3.1

JES3	Priced feature – many JES2 functions already added. Contact jes3q@us.ibm.com if you need more information. **As of z/OS V2.4, JES2 SMP/E zones are merged into the base zones, for z/OSMF portable software instances (ServerPac).**	z/OS 3.1
IBM Bulk Data Transfer (BDT) Features	Functional replacements for BDT F2F are IBM MQ Advanced for z/OS (5655-AV9), which include IMB MQ Managed File Transfer and MQ Advanced Message Security, and IBM Sterling Connect:Direct for z/OS (5655-X11).	z/OS 3.1
IBM z/OS Global Mirror(XRC)	New functions to support asynchronous replication technology are intended to be developed only for DS8000 Global Mirror, and it is intended that no new z/OS Global Mirror functions will be provided with DS8900F and z/OS.	z/OS 3.1
Distributed File Manager	Base element – DFSMS. If you use DFM to enable remote client, it is recommended to use z/OS NFS instead.	z/OS 3.1
ISFPARMS assembler macros	Priced feature – SDSF. Use ISFPRMxx parmlib member instead. This has been a long time recommendation.	z/OS 3.1
Knowledge Center for z/OS (KC4Z)	Base function – IBM intends to deliver a new component called DOC4Z on z/OS. DOC4Z is a web application that provides IBM product publication content to web browser clients directly from a local z/OS server system.	z/OS 3.1

10

Withdrawn in z/OS 3.1 (last delivered in z/OS V2.5)

This section lists items that were withdrawn in z/OS 3.1. You should take this into account if you are upgrading from z/OS V2.5 or V2.4 to z/OS 3.1. The removal of these functions may have upgrade actions which you can perform now, in preparation for z/OS 3.1.

- IBM announced that JES2 is the strategic Job Entry Subsystem (JES) for the z/OS Operating System and that JES3 would continue to be supported and maintained. To date, IBM has made significant investment in JES2 by delivering unique functions such as email support in JCL, spool migration and merge, and dynamic checkpoint expansion and tuning to make management easier. In z/OS V2.4, IBM plans to deliver in JES2 Spool Encryption and a new user exit alternative based on defining policies that allow exit programs to be implemented in a parameterized rule-based approach. To help JES3 to JES2 migration efforts, JES2 has added functionality, including dependent job control, deadline scheduling, 8-character job classes, and interpreting JES3 JECL control statements. For z/OS V2.4, additional function to aid in migrations is planned, including Disk Reader capability and enhanced JES3 JECL support in JES2 (ROUTE XEQ). Today, as a result of our strategic investment and ongoing commitment to JES2, as well as continuing to enhance JES3 to JES2 migration aids, IBM is announcing that the release following z/OS V2.4 is planned to be the last release of z/OS that will include JES3 as a feature. If you are one of the clients who remains on JES3, IBM encourages you to start planning your migration. For questions, contact jes3q@us.ibm.com.
- z/OS 2.5 will be the last release that BDT is included in z/OS. This applies to both priced features, BDT SNA NJE and BDT File-to-File (F2F). BDT SNA NJE offers JES3 clients the capability to send information over SNA networks to other end points. Note that BDT SNA NJE does not apply to JES2 clients because this function has always been included as part of JES2. The BDT F2F feature offers both JES3 and JES2 clients the capability of managed file copying from one system to another system. Functional replacements for BDT F2F are IBM MQ Advanced for z/OS (5655-AV9), which includes IBM MQ Managed File Transfer and MQ Advanced Message Security, and IBM Sterling™ Connect:Direct® for z/OS (5655-X11). Support is planned to be provided for BDT, BDT SNA NJE, and BDT F2F until the end of support for the next z/OS release after z/OS V2.4.

- z/OS support for z/OS Global Mirror: For decades, IBM has offered two asynchronous replication strategies, IBM z/ OS Global Mirror, also known as extended remote copy, or XRC, and DS8000 Global Mirror. IBM plans to support and maintain z/OS Global Mirror on z/OS with its current function only, and z/OS V2.5 will be the last release to provide such support. This withdrawal aligns with what was previously announced in Hardware Announcement 920-001, dated January 7, 2020, which indicated the DS8900F family would be the last platform to support z/OS Global Mirror. New functions to support asynchronous replication technology are intended to be developed only for DS8000 Global Mirror, and it is intended that no new z/OS Global Mirror functions will be provided with DS8900F and z/OS.
- Deprecation of DFSMS Distributed FileManager**:** z/OS V2.5 is planned to be the last release to support the DFSMS Distributed FileManager (DFM), a seldom used function in z/OS. To determine if DFM is being used, it is recommended to look for JCL that starts DFM; for example, START DFM,SUB=MSTR. If you use DFM to enable remote clients in your network to access data on z/OS systems, it is recommended to use the z/OS Network File System (NFS) instead.
- System Display and Search Facility (SDSF) ISFPARMS removal: For many z/OS releases, a recommended update action has been to specify z/OS SDSF customization with the ISFPRMxx parmlib member. There are several major advantages to using the ISFPRMxx parmlib member format over the original format, which involves an assembler module and SDSF macros. Beginning with the following release after z/OS V2.5, IBM plans that only the ISFPRMxx parmlib member format will be supported. For this reason, if the parmlib member ISFPRMxx is not currently being used, IBM recommends clients convert to using ISFPRMxx to avoid being impacted in the future.
- IBM intends to deliver a new component called DOC4Z on z/OS to replace Knowledge Center for z/OS (KC4Z). DOC4Z is a web application that provides IBM product publication content to web browser clients directly from a local z/OS server system. IBM also intends to provide IBM Documentation APIs for clients to programmatically interact with DOC4Z.

Functions Planned to be Withdrawn

in the releases after z/OS 3.1…

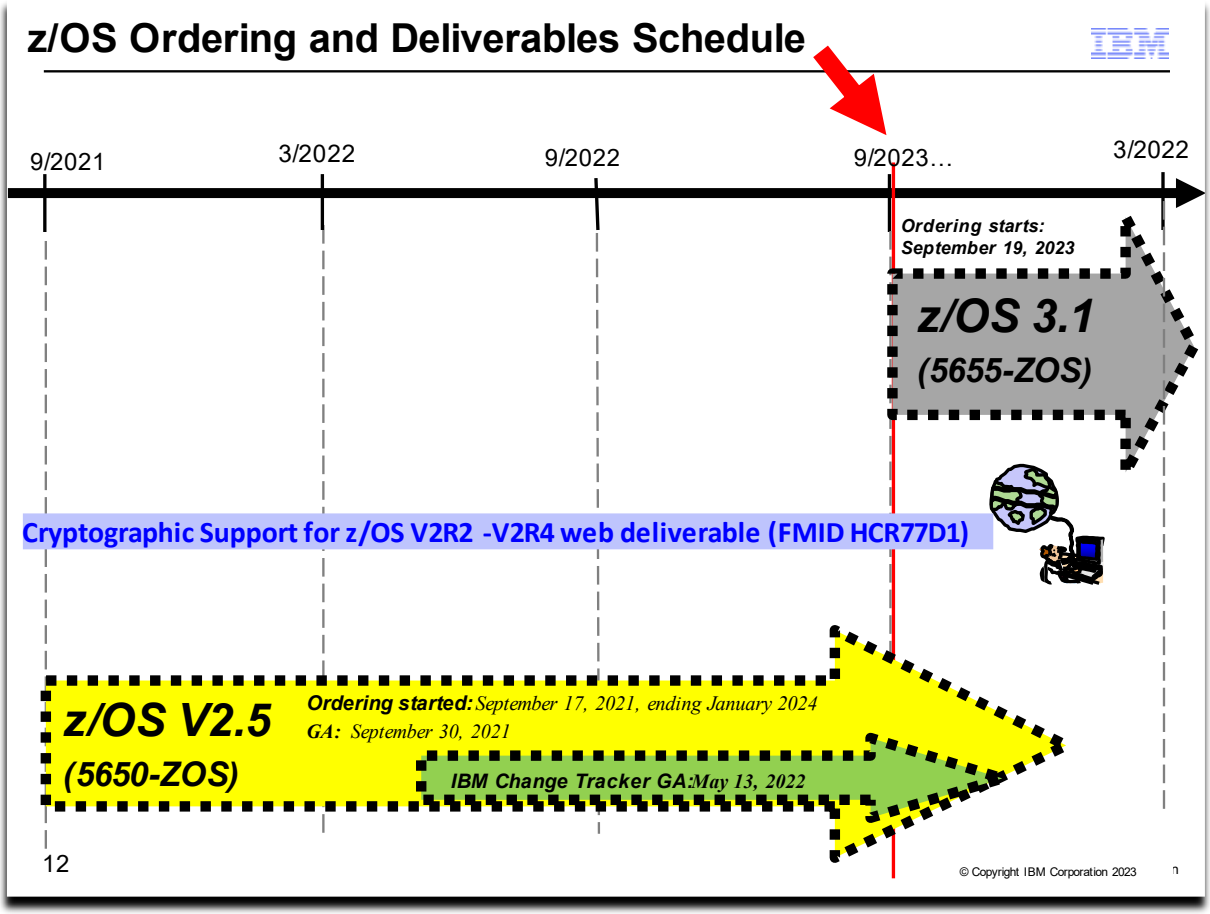
DFSMSdfp Checkpoint/Restart	Base function- Any remaining z/OS software that still depends on checkpoint restart capability may need to be redesigned to remove the dependency on checkpoint/restart. (CHKPT macro is intended to be syntax checked and ignored. Restart capability will need to be redesigned to remove the dependency on checkpoint/restart.) z/OS Generic Tracking is planned with APAR OA64519 on z/OS V2.4 and higher.	Planned for the release after z/OS 3.1
Common Information Module	Base element– All z/OS software that depends on a CIM server running on z/OS will be to be updated to remove the dependency.	Planned for the release after z/OS 3.1

11

Planned for removal in the release following z/OS V2.5

This section lists items that IBM has announced it intends to remove in the releases after z/OS V2.5 and beyond. You are encouraged to consider these removals when making your plans for system upgrades. These statements represent IBM's current intentions. IBM development plans are subject to change or withdrawal without further notice.

- z/OS 3.1 is planned to be the last release to support DFSMSdfp Checkpoint/Restart. The intent is not to require changes to applications with regards to usage of the CHKPT macro. Usage of the CHKPT macro is intended to be syntax checked and ignored. Any remaining z/OS software that still depends on checkpoint restart capability may need to be redesigned to remove the dependency on checkpoint/restart. Updates to allow identification of usage of Checkpoint/Restart are planned to be available via the Generic Tracking Facility, with the PTFs for APAR OA64519 on z/OS V2.4 and later. z/OS continues to provide Job restart processing, which works on a step basis as well as capabilities like Transactional VSAM which may provide the basis for solutions that could replace checkpoint/restart.
- z/OS 3.1 is planned to be the last z/OS release in which IBM intends to include the Common Information Model (CIM) server. All z/OS software that depends on a CIM server running on z/OS will need to be upgraded to remove the dependency.



z/OS Ordering and Deliverable Key Dates
Key dates for recent z/OS releases and functions:

- **September 2019:** Availability date for the Cryptographic Support for the z/OS V2R2-V2R4 web deliverable. (The FMID is HCR77D1.)
- **September 2021:** z/OS V2.5 general availability.
- **May 13, 2022:** priced feature on z/OS V2.5 is generally available, IBM z/OS Change Tracker
- **September 19, 2023:** z/OS 3.1 ordering begins
- **September 29, 2023:** z/OS 3.1 general availability.
- **January 2024:** Ordering complete for z/OS V2.5.

Web deliverables

Sometimes enhancements are provided as Web deliverables***, and not integrated in your ServerPac or CBPDO deliverable***. For example, some of the ICSF enhancements are available this way. z/OS Web deliverables are available from http://www.ibm.com/eserver/zseries/zos/downloads/. They are packaged as two files that you download:

- A **readme** file, which contains a sample job to uncompress the second file, transform it into a format that SMP/E can process, and invoke SMP/E to RECEIVE the file. This file must be downloaded as text.

- A **pax.z** file, which contains an archive (compressed copy) of the FMIDs to be installed. This file needs to be downloaded to a workstation and then uploaded to a host as a binary file.

For Web downloads, you perform the SMP/E installation work yourself.

Cryptographic Support for z/OS V2R2-V2R4 Web deliverable (ICSF FMID HCR77D1) was available September 2019. This web deliverable supports z/OS V2.2, V2.3, and V2.4. ICSF provides the following new features:

- Support for the new Crypto Express7S adapter, configured as a CCA coprocessor, an EP11 coprocessor, or an accelerator.
- The ability to use CP Assist for Cryptographic Functions (CPACF) for certain clear key ECC operations. ICSF can now call CPACF instructions to perform ECC key generation, key derivation, and digital signature generation and verification using a subset of the NIST curves. The CPACF on IBM z15 also supports the ed448 and ec25519 curves.
- A new SMF record whenever a master key is changed. Certain compliance regulations mandate the periodic rotation of encryption keys, including the master keys loaded into coprocessors. As part of the master key change process, an SMF record will now be written every time the new master key is promoted to the current master key as part of the change master key ceremony.
- A health check that verifies a system's ability to use the NIST recommended PSS signature algorithms. It is not obvious that the ECC master key is required when generating and using RSA keys enabled for PSS signatures, so a health check will help clients understand the need for this additional master key so they can begin to exploit the recommended algorithms.
- New quantum safe algorithms for sign and verify operations. With this release of ICSF, it is now possible to use quantum safe encryption algorithms for digital signature operations, which also includes the ability to generate and store new keys. These algorithms will be clear key only and available via the PKCS #11 interfaces.
- Support for CCA Release 5.5 and CCA Release 6.3 including:
    - New services in support of ANSI TR-34 Remote Key Loading.
    - PCI HSM Compliance for AES and RSA keys.
    - Additional AES based financial services.
    - Note: These functions were made available on ICSF FMD HCR77D0 with PTFs for APAR OA57089.

Crypto Support in z/OS Releases

Your ICSF upgrade actions depend on where you are coming from, and where you are going…

Release	*Level Integrated and Available*
z/OS 3.1	**ICSF FMID HCR77E0 is incorporated.**
z/OS V2.5	**ICSF FMID HCR77D2 is incorporated.** • **Future ICSF HW support will be provided in PTFs** • **No Web deliverable!** • **PTFs will be marked with the HW FIXCAT**
z/OS V2R4	**ICSF FMID HCR77D0 is incorporated.** • **Cryptographic Support for z/OS V2R2 -V2R4 Web deliverable (FMID HCR77D1) available 9/2019.**

Happy news!

13

z/OS ICSF Release Levels

The support for cryptography (z/OS base element ICSF) has been delivered via Web deliverables and release incorporations over the years.

z/OS Releases and Crypto Web Deliverables		
Deliverable	**General availability**	**Delivery method**
z/OS V2R2, FMID HCR77B0 Incorporated	September 30, 2015	New release of products
Cryptographic Support for z/OS V1R13-V2R2 (FMID HCR77B1)	November 2015	Web deliverable
Cryptographic Support for z/OS V2R1-V2R2 (FMID HCR77C0)	October 2016	Web deliverable
Cryptographic Support for z/OS V2R1-V2R3 (FMID HCR77C1)	September 2017	Web deliverable
Cryptographic Support for z/OS V2R2-V2R3 (FMID HCR77D0)	December 2018	Web deliverable
Cryptographic Support for z/OS V2R2-V2R4 (FMID HCR77D1)	September 2019	Web deliverable
z/OS V2R3, FMID HCR77C0 Incorporated	September 29, 2017	New release of products
Cryptographic Support for z/OS V2R1-V2R3 (FMID HCR77C1)	September 2017	Web deliverable
Cryptographic Support for z/OS V2R2-V2R3 (FMID HCR77D0)	December 2018	Web deliverable
Cryptographic Support for z/OS V2R2-V2R4 (FMID HCR77D1)	September 2019	Web deliverable
z/OS V2R4, FMID HCR77D0 Incorporated	September 2019	New release of products
Cryptographic Support for z/OS V2R2-V2R4 (FMID HCR77D1)	September 2019	Web deliverable
z/OS V2.5, FMID HCR77D2 Incorporated	September 2021	New release of products

ICSF Support for z/OS V2.5 for HW	September 2021	PTFs
z/OS 3.1, FMID HCR77E0 Incorporated	September 2023	New release of products

Refer to this technote: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD103782 for a complete history of ICSF deliverables, and functions contained in those deliverables.

Some orderable no-charge** products (with z/OS 3.1, 5655-ZOS)

- *IBM Semeru Runtime Certified Edition for z/OS, 11 (5655-DGJ, 5655-I48) 64-bit only*
  - *At GA, prereq for z/OS 3.1 (note few exceptions)!*
- *IBM 31-bit SDK for z/OS V8 (5655-DGG, 5655-I48)*
  - *At GA, prereq for z/OS 3.1 for those few exceptions!*
- *IBM 64-bit SDK for z/OS V8 (5655-DGH, 5655-I48)*
- *IBM Semeru Runtime Certified Edition for z/OS, 17 (5655-UA1) 64-bit only*
  - *Sometime, in the life of z/OS 3.1, this will move to be the prereq!*
- ~~*XML Toolkit for z/OS V1 R15 (5655-J51, 5655-I30)*~~
  - **XML Toolkit can only be ordered as a Product ServerPac (z/OSMF Portable Software Instance), or with z/OS V2.5. It is not orderable with z/OS 3.1**
- *IBM AI System Services for IBM z/OS (5655-I64)** ← new product!*
- *SDK for Node.js - z/OS 16.0 (5655-NOE, 5655-SDS)***
- *SDK for Node.js - z/OS 18.0 (5655-NOE, 5655-SDS)***

***Subscription & Support (S&S) may be priced*

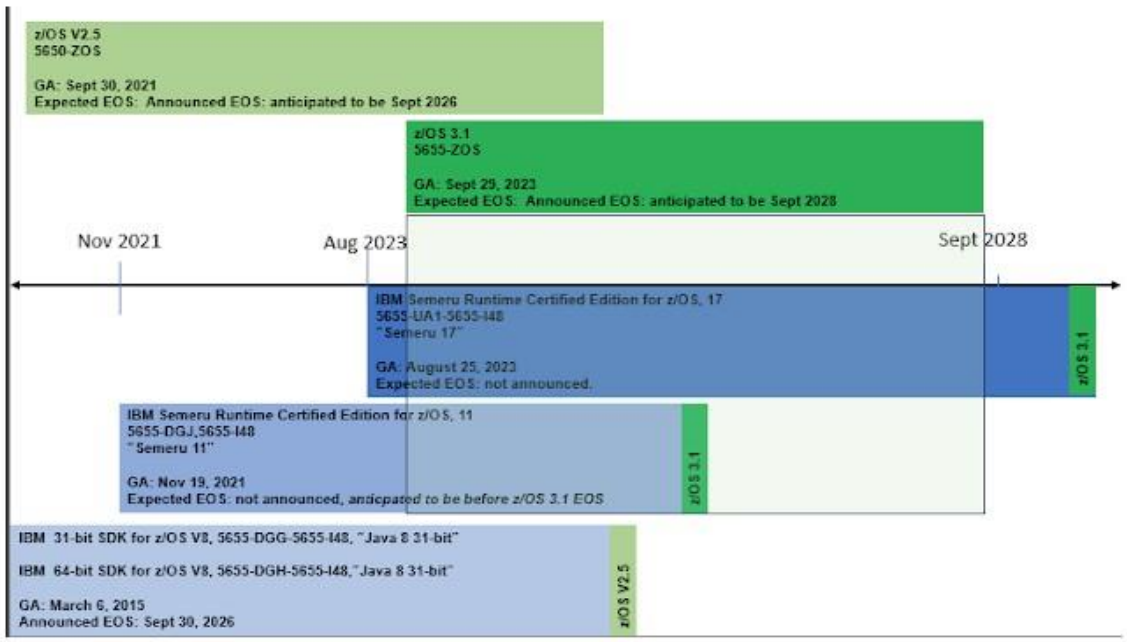
Announced Java 8 dates:

End of Marketing: IBM 64bit SDK for z/OS V8 (5655-DGH) and 31-bit (5655-DGG): Jan 29, 2024

End of Service: IBM 64bit SDK for z/OS V8 (5655-DGH) and 31-bit (5655-DGG): Sept 30, 2026

14

z/OS 3.1 Java Dependencies



Read this blog entry for a more in depth discussion:
https://www.marnasmusings.com/2023/08/having-your-java-and-drinking-it-too.html

15

IBM Java SDK for z/OS V17 (5655-UA1-5655-UA2). “Semeru 17”
The IBM Semeru Runtime Certified Edition for z/OS, version 17.0 product is certified with the Java Compatibility Kit as a fully compliant Java product, which includes the IBM z/OS platform ports of the OpenJDK Java 17 class libraries and the Eclipse OpenJ9 Java Virtual Machine.
The IBM Semeru Runtime Certified Edition for z/OS, version 17.0 is operational within the z/OS V2.5 operating system. It provides a Java execution environment equivalent to that available on any other server platform.

IBM Java SDK for z/OS V11 (5655-DGJ-5655-I48). “Semeru 11”
IBM Semeru Runtime Certified Edition for z/OS, Version 11, formerly known as IBM 64-bit SDK for z/OS, Java Technology Edition, is certified with the Java Compatibility Kit as a fully compliant Java product. This rebranding aligns the naming for Java on the z/OS platform with that on other platforms where OpenJDK class libraries are similarly powered by the Eclipse OpenJ9 JVM technology. This offering includes the IBM z/OS platform ports of the OpenJDK Java 11 class libraries and the Eclipse OpenJ9 Java virtual machine (JVM), giving z/OS developers the capability to use new Java language features and currency with the Java community. Furthermore, this compliance can enable developers to confidently port Java applications developed on distributed platforms to z/OS.

Significant changes introduced with Semeru Runtime Certified Edition for z/OS, Version 11 (5655-DGJ) might require existing applications running 64-bit SDK for z/OS, Java Technology Edition, V8 (5655-DGJ), or earlier, to be updated.Client applications that previously used the 31-bit Java SDK might need to be modified to run in 64-bit mode.

IBM AI System Services for IBM z/OS (September 29, 2023) 5655-164 and 5655-165 (S&S)

AI System Services for IBM z/OS 1.1 delivers foundational AI capabilities and represents one of the key components of the AI Framework for IBM z/OS that is intended to support AI infusion into z/OS. This solution provides IT data ingestion and filtering capabilities to allow the collection of data for training and inference purposes. Furthermore, it delivers AI model server capabilities that support the AI model lifecycle, including AI model training, deployment, inference, monitoring, and retraining. The combination and integration of AI System Services for IBM z/OS with the rest of the AI Framework for IBM z/OS components that are delivered within z/OS 3.1 enable you to put prebuilt AI models into operation. Based on those capabilities, system programmers can leverage prebuilt and operationalized AI models for use cases that are geared towards helping to simplify the management of z/OS and its offerings by augmenting them with AI to help:

- Reduce skill requirements
- Optimize IT processes
- Improve performance

AI System Services for IBM z/OS enables the setup and use of AI-infused capabilities into z/OS base components, starting with the initial use case: AI-powered WLM batch initiator management.

AI System Services for IBM z/OS represents an important set of capabilities that are part of the AI Framework for IBM z/OS. The AI Framework for IBM z/OS, including AI System Services for IBM z/OS, is a new AI platform introduced with z/OS 3.1 that is designed to seamlessly integrate a set of components to enable the operationalization and usage of AI-infused capabilities into z/OS base components. It is intended to offer a seamless and simplified installation, setup, and management experience of the AI-infused capabilities without requiring additional data science or AI skills. It is designed to pave the way for AI use case providers that can harness the foundational AI capabilities to address AI model operationalization requirements, simplify the process to put future AI use cases to work, and accelerate time to market.

IBM Open Enterprise SDK for Node.js 16.0 (December 16, 2022) 5655-NOE and 5655-SDS (S&S)

IBM Open Enterprise SDK for Node.js is a server-side JavaScript runtime environment available on z/OS. Node.js is one of the fastest growing language runtimes in the market, with a large open-source community. This popularity is driven by its simplicity, speed, ease of use, and massive ecosystem with more than 1 million npm packages that can be used immediately in applications.

Open Enterprise SDK for Node.js 16.0 is based on the open-source Node.js 16 LTS (long-term support) community release. Node.js 16 is the latest major LTS release, incorporating V8 JavaScript engine version 9 features, npm version 8 features, and many other upgrades, including the Timers Promises API that is promoted to stable status. The z/OS-specific features include native EBCDIC I/O support and improved code-page auto-detection heuristics.

IBM Open Enterprise SDK for Node.js 18.0 (December 16, 2022) 5655-NOJ and 5655-SDS (S&S)

IBM Open Enterprise SDK for Node.js 18.0, the follow-on product to Open Enterprise SDK for Node.js 16.0, provides a stand-alone JavaScript runtime and server-side JavaScript solution for building Node.js native and JavaScript modules for the IBM zSystems platform. Open Enterprise SDK for Node.js 18.0 is based on the open-source Node.js 18 LTS (long-term support) community release.

Open Enterprise SDK for Node.js is a server-side JavaScript runtime environment available on z/OS. Node.js is one of the fastest growing language runtimes in the market, with a large open source community. This popularity is driven by its simplicity, speed, ease of use, and massive ecosystem with more than 1.8 million npm packages that can be used immediately in applications. With more than 1 billion downloads, Node.js continues to gain traction in the developer community. Available and supported on the IBM zSystems platform, Open Enterprise SDK for Node.js provides extra security and performance by leveraging the capabilities of IBM zSystems. Open Enterprise SDK for Node.js enables clients to develop on IBM zSystems in the same way as on any other platform.

End of Service Dates for Older IBM XML and Java SDK levels:

- **XML V1R9** was out of service on September 30, 2013.
- **IBM 64-bit SDK for z/OS, Java 2 Technology Edition, V1 Release 4 (5655-I56):** was out of service as of September 30, 2008.
- **IBM 31-bit SDK for z/OS, Java 2 Technology Edition, V1 Release 4 (5655-M30):** was out of service as of September 30, 2011. z/OS R11 was the last release for which IBM SDK V1R4 support was available.
- **IBM 64-bit SDK for z/OS, Java 2 Technology Edition, V5 Release 0 (5655-N99):** was out of service as of September 30, 2013.
- **IBM 31-bit SDK for z/OS, Java 2 Technology Edition, V5 Release 0 (5655-N98):** was out of service as of September 30, 2013.
- **IBM 64-bit SDK for z/OS V6 Release 0 (5655-R32):** was out of service as of September 30, 2018.
- **IBM 31-bit SDK for z/OS V6 Release 0 (5655-R31):** was out of service as of September 30, 2018.
- **IBM 64-bit SDK for z/OS V7 Release 0 (5655-W44):** was out of service as of September 30, 2019.
- **IBM 31-bit SDK for z/OS V7 Release 0 (5655-W43):** was out of service as of September 30, 2019.
- **IBM 64-bit SDK for z/OS V7 Release 1 (5655-W44):** was out of service as of April 30, 2022.
- **IBM 31-bit SDK for z/OS V7 Release 1 (5655-W43):** was out of service as of April 30, 2022.
- **IBM 64-bit SDK for z/OS V8 (5655-DGH) and 31-bit (5655-DGG):** announced to be end of marketing on January 29, 2024, and end of service on September 30, 2026.

z/OS Service Policy

- **Priced service extensions available for older releases:**
  - V2.2 and V2.3: Contact your IBM rep.
- **With the V2 release cycle, IBM plans to provide 5 years of z/OS support, with 3 years of optional extended service (5+3).**

	GA Date	*End of Service Date*
z/OS V2.3	**30 Sept 2017**	**Occurred to be Sept 2022 (5 years!)**
z/OS V2.4	**29 Sept 2019**	**Planned to be Sept 2024 (5 years)**
z/OS V2.5	**30 Sept 2021**	**Plann to be Sept 2026 (5 years)**
z/OS 3.1	**29 Sept 2023**	**Planned to be Sept 2028 (5 years)**

One year!

15

Service Policy
With the two-year z/OS release frequency, the z/OS support policy is five years of z/OS support, with three years of optional extended service (5+3).

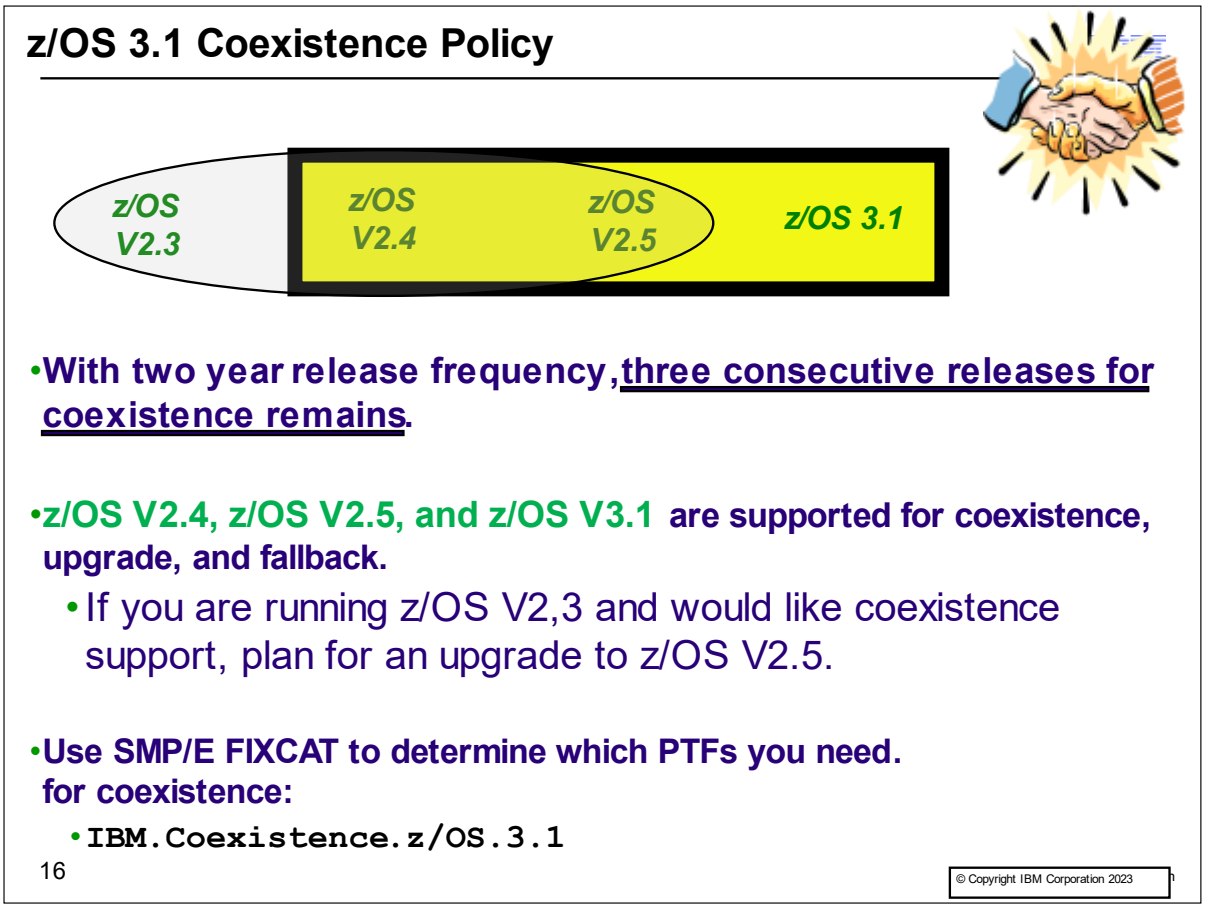
Prior to withdrawing service for any version or release of z/OS or z/OSMF, IBM intends to provide at least 12 months notice. The service policy for z/OS also applies to any enhancements (including but not limited to web deliverables).

See the table below for expiration dates for service support.

Version and release	General availability (GA)	End of service (EOS)
OS/390 V2R8	**24 September 1999**	**Occurred 30 September 2002**
OS/390 V2R9	**31 March 2000**	**Occurred 31 March 2003**
OS/390 V2R10	**29 September 2000**	**Occurred 30 September 2004**
z/OS V1R1	**30 March 2001**	**Occurred 31 March 2004**
z/OS V1R2	**26 October 2001**	**Occurred 31 October 2004**
z/OS V1R3	**29 March 2002**	**Occurred 31 March 2005**
z/OS V1R4 **z/OS V1R5** **z/OS V1R6**	**27 September 2002** **26 March 2004** **24 September 2004**	**Occurred on 31 March 2007** **Occurred on 31 March 2007** **Occurred on 30 September 2007**
z/OS V1R7	**30 September 2005**	**Occurred on 30 September 2008 *** ***The "z/OS V1.7 Lifecycle Extended Service" offering expired on 30 September 2010.** If you require support for defects for z/OS V1R7 beyond September 2010, contact an IBM representative for a special bid.
z/OS V1R8	**29 September 2006**	**Occurred 30 September 2009 *** ***The "z/OS V1.8 Lifecycle Extended Service" offering expired on 30 September 2011.** If you require support

		for defects for z/OS V1R8 beyond September 2011, contact an IBM representative for a special bid.
z/OS V1R9	**28 September 2007**	**Occurred 30 September 2010 *** ***The "z/OS V1.9 Lifecycle Extended Service" offering expired on 30 September 2012.** If you require support for defects for z/OS V1R9 beyond September 2012, contact an IBM representative for a special bid.
z/OS V1R10	**26 September 2008**	**Occurred 30 September 2011 *** ***The "z/OS V1.10 Lifecycle Extended Service" offering expired on 30 September 2013.** If you require support for defects for z/OS V1R10 beyond September 2013, contact an IBM representative for a special bid.
z/OS V1R11	**25 September 2009**	**Occurred 30 September 2012 *** ***See "z/OS V1.11 Lifecycle Extended Service" below for a fee-based accommodation, through 30 September 2014.** ***See "IBM Software Support Services for z/OS V1.11" below for a fee-based accommodation, through 30 September 2016.**
z/OS V1R12	**24 September 2010**	**Occurred 30 September 2014** ***See "IBM Software Support Services for z/OS V1.12" below for a fee-based accommodation, through 30 September 2017**
z/OS V1R13	**30 September 2011**	**Occurred 30 September 2016**
z/OS V2R1	**30 September 2013**	**Occurred 28 September 2018**
z/OS V2R2	**30 September 2015**	**Occurred 30 September 2020**
z/OS V2R3	**29 September 2017**	**Announced for September 2022**
z/OS V2R4	**29 September 2019**	**Planned for September 2024**
z/OS V2.5	**30 September 2021**	**Planned for September 2026**
z/OS 3.1	**29 September 2023**	**Planned for September 2028**

If you wish to purchase corrective extended service on an unsupported z/OS release, contact your IBM representative.



z/OS Coexistence

Coexistence occurs when two or more systems at different software levels share resources. The resources could be shared at the same time by different systems in a multisystem configuration, or they could be shared over a period of time by the same system in a single-system configuration. Examples of coexistence are two different JES releases sharing a spool, two different service levels of DFSMSdfp sharing catalogs, multiple levels of SMP/E processing SYSMODs packaged to exploit the latest enhancements, or an older level of the system using the updated system control files of a newer level (even if new function has been exploited in the newer level).

The sharing of resources is inherent in multisystem configurations that involve Parallel Sysplex implementations. But other types of configurations can have resource sharing too. Examples of configurations where resource sharing can occur are:

- A single processor that is time-sliced to run different levels of the system, such as during different times of the day
- A single processor running multiple images by means of logical partitions (LPARs)
- Multiple images running on several different processors
- Parallel Sysplex or non-Parallel Sysplex configurations

Note: The term coexistence does not refer to z/OS residing on a single system along with VSE/ESA, VM/ESA, or z/VM in an LPAR or as a VM guest.

z/OS systems can coexist with specific prior releases. This is important because it gives you flexibility to migrate systems in a multisystem configuration using rolling IPLs rather than requiring a systems-wide IPL. The way in which you make it possible for earlier-level systems to coexist with z/OS is to install coexistence service (PTFs) on the earlier-level systems.

You should complete the upgrade of all earlier-level coexisting systems as soon as you can. Keep in mind that the objective of coexistence PTFs is to allow existing functions to continue to be used on the earlier-level systems when

run in a mixed environment that contains later-level systems. Coexistence PTFs are not aimed at allowing new functions provided in later releases to work on earlier-level systems.

Rolling z/OS across a multisystem configuration

A *rolling IPL* is the IPL of one system at a time in a multisystem configuration. You might stage the IPLs over a few hours or a few weeks. The use of rolling IPLs allows you to migrate each z/OS system to a later release, one at a time, while allowing for continuous application availability. For example, data sharing applications offer continuous availability in a Parallel Sysplex configuration by treating each z/OS system as a resource for processing the workload. The use of rolling IPLs allows z/OS systems running these applications to be IPLed one at a time, to migrate to a new release of z/OS, while the applications continue to be processed by the other z/OS systems that support the workload. By using LPAR technology, you can use rolling IPLs to upgrade your systems without losing either availability or capacity.

You can use rolling IPLs when both of the following are true:

- The release to which you're migrating falls is supported for coexistence, fallback, and upgrade with the releases running on the other systems.
- The appropriate coexistence PTFs have been installed on the other systems in the multisystem configuration.

Even when you're using applications that do not support data sharing, rolling IPLs often make it easier to schedule z/OS software upgrades. It can be very difficult to schedule a time when all applications running on all the systems in a multisystem configuration can be taken down to allow for a complex-wide or Parallel Sysplex-wide IPL. The use of rolling IPLs not only enables continuous availability from an end-user application point of view, but it also eliminates the work associated with migrating all z/OS systems in a multisystem configuration at the same time.

Understanding fallback

Fallback (backout) is a return to the prior level of a system. Fallback can be appropriate if you upgrade to z/OS V2.5 and, during testing, encounter severe problems that can be resolved by backing out the new release. By applying fallback PTFs to the "old" system before you migrate, the old system can tolerate changes that were made by the new system during testing.

Fallback is relevant in all types of configurations, that is, single-system or multisystem, with or without resource sharing. As an example of fallback, consider a single system that shares data or data structures, such as user catalogs, as you shift the system image from production (on the "old" release) to test (on the new release) and back again (to the old release). The later-level test release might make changes that are incompatible with the earlier-level production release. Fallback PTFs on the earlier-level release can allow it to tolerate changes made by the later-level release.

As a general reminder, always plan to have a backout path when installing new software by identifying and installing any service required to support backout.

Fallback is at a system level, rather than an element or feature level.

Fallback and coexistence are alike in that the PTFs that ensure coexistence are the same ones that ensure fallback.

Note: Keep in mind that new functions can require that all systems be at z/OS 3.1 level before the new functions can be used. Therefore, be careful not to exploit new functions until you are fairly confident that you will not need to back out your z/OS 3.1 systems, as fallback maintenance is not available in these cases. You should consult the appropriate element or feature documentation to determine the requirements for using a particular new function.

Which releases are supported for coexistence, fallback, and upgrade?

- IBM plans to continue to support an n-2 (three consecutive release) coexistence, fallback, and upgrade policy.
- Do note that with the two-year release cycle that z/OS support is intended for five years with three optional extended service years (5+3). You should plan on completing your upgrade plans during the period of time while your older z/OS release is still in service.
- **Starting with z/OS V1R6, IBM has aligned the coexistence, fallback, and upgrade policy with the service policy.** IBM intends to continue with the practice of providing coexistence, fallback, and policy support for those releases which are still in support.

z/OS 3.1 is coexistence, fallback, and upgrade supported with the following two z/OS releases: V2.4 and V2.5. This means that:

- Coexistence of a 3.1 system with a V2.4 or V2.5 system is supported.
- Fallback *from* 3.1 *to* V2.4 or V2.5 is supported.
- Upgrade *to* 3.1 *from* V2.4 or V2.5 is supported

The z/OS coexistence, fallback, and upgrade policy applies to the elements and features of z/OS, not to customer-developed applications, vendor-developed applications, or IBM products that run on z/OS. IBM performs integration testing and will provide service as necessary to support the z/OS coexistence, fallback, and upgrade policy.

See the table below for a summary of current and planned coexistence, fallback, and upgrade support.
These statements represent IBM's current intentions. IBM reserves the right to change or alter the coexistence, fallback, and upgrade policy in the future or to exclude certain releases beyond those stated. IBM development plans are subject to change or withdrawal without further notice. Any reliance on this statement of direction is at the relying party's sole risk and does not create any liability or obligation for IBM.

Releases that are coexistence, fallback, and upgrade supported as of z/OS R10

z/OS release	Releases that are coexistence, fallback, and upgrade supported with the release in column	Explanation
R10	R10, R9, R8	General availability of R10 was September 26, 2008. R8 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with R10.
R11	R11, R10, R9	General availability of R11 was September 25, 2009. R9 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with R11.
R12	R12, R11, R10	General availability of R12 was September 24, 2010. R10 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with R12.
R13	R13, R12, R11	General availability for R13 was September 30, 2011. R11 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with R13.
V2R1	V2R1, R13, R12	General availability for V2R1 was September 30, 2013. R12 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with V2R1.
V2R2	V2R2, V2R1, R13	General availability for V2R2 was September 30, 2015. R13 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with V2R2.
V2R3	V2R3, V2R2, V2R1	General availability for V2R3 was September 29, 2017. V2R1 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with V2R3.
V2R4	V2R4, V2R3, V2R2	General availability for V2R4 was September 2019. V2R2 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with V2R4.
V2.5	V2.5, V2R4, V2R3	General availability for V2.5 was September 2021. V2R3 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with V2R5.
3.1	3.1, V2.5, V2.4	General availability for 3.1 is September 2023. V2.4 is the oldest release that is service supported at that time and therefore the oldest release that is coexistence, fallback, and upgrade supported with 3.1.

Positioning for z/OS 3.1

- **Perform Workflow and Read Documentation**
  - ***z/OS 3.1 Upgrade Workflow*** and ***z/OS 3.1 Planning for Installation***
- Approximate **DASD Storage Requirements for z/OS**
  - All z/OS features, including Japanese (as of June 2023)
  - Your sizes will vary significantly, based on what was ordered with z/OS

3390 cyl	z/OS V2R4	z/OS V2.5	z/OS 3.1
Target libraries (PDS and PDSE)	**10,944**	**11,239**	**10,246**
DLIB	**19,255**	**19,231**	**19,022**
Root file system	**4,980 (HFS)** **4,990 (zFS)**	**4,479**	**5,555** Approaching the necessary! SMS zFS 4GB limit which is when EA is -management is NOT required.
Font file system	**2,800 (HFS or zFS)**	**2,795**	**2,795**
Liberty file system	**2,400 (HFS or zFS)**	**2,400**	**2,400**
zCX (z/OS Container Extensions) file system	**5,250 (HFS or zFS)**	**5,250**	**5,250**

17

Positioning for z/OS 3.1

- **Ensuring System Requirements Are Satisfied**
  - *Driving System Requirements for z/OSMF Portable Software Instance*
    - Minimally z/OS V2.4, with z/OSMF configured and active, and Software Management available for use.
    - Common pitfalls:
      - ✓ Your user ID requires READ access to data set names that begin with CB.OS* and CB.ST* for IBM ServerPacs.
      - ✓ Necessary PTFs are found with `IBM.DrivingSystem-RequiredService` FIXCAT.
    - For package signature verification, a keyring with the RACF -delivered **STG Code Signing Certificate Authority - G2** connected.
      - Package signing verification is completely optional and compatible.
      - No Shopz ordering indication is necessary. All product packages will arrive signed. You choose to verify, or not.
      - New news! IBM service (PTF) packages are being signed now!
    - Use the SMP/E FIXCAT `IBM.DrivingSystem-RequiredService` .
  - *Target System Requirements*
    - Product requirements for running on z/OS 3.1
    - Hardware and Software
  - *Coexistence System Requirements*
    - Allows z/OS 3.1 to coexist with other z/OS systems
- **Upgrade Actions You Can Do NOW**
- **Using Programmatic Assistance – Health Checks and Workflow**

19

z/OS Documentation:

To gain an overview of z/OS and plan for the installation, review:

- *z/OS 3.1 Upgrade* Workflow , supplied via PTF with FIXCAT **IBM.Coexistence.z/OS.3.1**
- *z/OS 3.1 Planning for Installation*
- *zSeries Platform Test Report for z/OS and Linux Virtual Servers* (formerly, the *z/OS Parallel Sysplex Test Report*)
- *z/OS 3.1 Introduction and Release Guide* - great for finding new functions in a release to exploit!

To install z/OS, review Preventive Service Planning (PSP) Buckets for:

- ServerPac (if using ServerPac to install)
- z/OS and individual elements (including ZOSGEN, which helps you with general z/OS level information)
- Hardware, if you will using specific HW functions or upgrading your server

To install z/OS using CBPDO, review the *z/OS Program Directory.*

PSP Buckets

The supported way to search for PSP buckets on the Web, is via IBM Support now. (The PSP Web site has been removed.) Use this website to search for PSP buckets: https://www.ibm.com/support/pages/ibmsearch. For z/OS 3.1, search for "**`upgrade zos31`**".

The upgrade for the z/OS 3.1 PSP bucket is **ZOS31**. Recognizing that there are many PSP buckets to review, z/OS uses descriptive element names, instead of FMIDs for the subsets. This reduces the number of PSP buckets that must be reviewed, since most elements are composed of multiple FMIDs. There are subsets in the ZOS31 upgrade for general topics (ZOSGEN), and for the ServerPac deliverable (SERVERPAC) that should be reviewed also. DFSMS is consolidated into one subset. All PSP upgrades and subset IDs are listed in the *z/OS Program Directory*. However, the non-exclusive elements' stand-alone product upgrade and subsets are used.

Hardware PSP upgrade identifiers

Hardware PSP bucket upgrade IDs are in the form xxxxDEVICE and contain the latest software dependencies for the hardware, and recommended PTFs and APARs required for specific processor models. The PSP hardware upgrade identifiers are:

- 3931DEVICE for the z16 A01 server.
  - The FIXCAT names are IBM.Device.Server.z16-3931.RequiredService, IBM.Device.Server.z16-3931.Exploitation, and IBM.Device.Server.z16-3931.RecommendedService.
- 3932DEVICE for the z16 A02 server.
  - The FIXCAT names are IBM.Device.Server.z16A02-3932.RequiredService, IBM.Device.Server.z16A02-3932.Exploitation, and IBM.Device.Server.z16A02-3932.RecommendedService.
- 8561DEVICE for the z15 T01 server.
  - The FIXCAT names are IBM.Device.Server.z15-8561.RequiredService, IBM.Device.Server.z15-8561.Exploitation, and IBM..Device.Server.z15-8561.RecommendedService.
- 8561DEVICE for the z15 T02 server.
  - The FIXCAT names are IBM.Device.Server. z15T02-8562.RequiredService, IBM.Device.Server.z15T02-8562.Exploitation, and IBM..Device.Server. z15T02-8562.RecommendedService.
- 3907DEVICE for the z14 ZR1 server.
  - The FIXCAT names are IBM.Device.Server.z14ZR1-3907.RequiredService, IBM.Device.Server. z14ZR1-3907.Exploitation, and IBM.Device.Server. z14ZR1-3907.RecommendedService.
- 3906DEVICE for the z14 server.
  - The FIXCAT names are IBM.Device.Server.z14-3906.RequiredService, IBM.Device.Server. z14-3906.Exploitation, and IBM.Device.Server. z14-3906.RecommendedService.

Specific functions for each server also have corresponding FIXCATs.

DASD Storage Requirements

Keep in mind the DASD required for your z/OS system includes all z/OS elements (per the z/OS Policy). That is, it includes ALL elements, ALL features that support dynamic enablement, regardless of your order, and ALL unpriced

features that you ordered. This storage is in addition to the storage required by other products you might have installed. All sizes include 15% freespace to accommodate the installation of maintenance.

The estimated total storage required for z/OS 3.1 data sets is provided below. If you add other products to your z/OS 3.1 z/OSMF Portable Software Instance (ServerPac), you will need additional space for those other products.

For z/OS 3.1 (as of June 2023):

- The total storage required for all the target data sets is approximately 10,246 cylinders on a 3390 device. **This total size exceeds the space on a 3390-9.**
- The total storage required for all the distribution data sets is approximately 19,022 cylinders on a 3390 device.
- The total executable root file system storage is approximately at 5,555 cylinders on a 3390 for zFS. Use IBM Health Checker for z/OS check ZOSMIGREC_ROOT_FS_SIZE to determine whether a volume has enough space for the z/OS version root file system, available back to z/OS R9 in APARs OA28684 and OA28631.

  - Note that this size is extremely close to the 4GB limit in which Extended Addressability (EA) is required. EA does not require SMS-management.

- z/OS V2.1 introduced the z/OS Font Collection base element. This element installs into a separate file system (the “font file system”). You may choose to merge the font file system and root file system if you desire. z/OSMF Software Management supports this merge capability. The total font file system storage is estimated at 2,795 cylinders on a 3390 device.
- z/OS V2.3 introduces the IBM z/OS Liberty Embedded base element. This element installs into a separate file system (the “Liberty file system”). IBM recommends that you keep the Liberty file system separate from other file systems in case of space fluctuations. The total Liberty file system storage is estimated at 2,400 cylinders on a 3390 device.
- z/OS V2.4 introduces the IBM Container Extensions (zCx) base element. This element installs into a separate file system (the “zCX file system”). IBM recommends that you keep the zCX file system separate from other file systems in case of space fluctuations. The total xCX file system storage is estimated at 5,250 cylinders on a 3390 device.

- For configuration and execution, additional file system space is needed:
  - You will need 50 cylinders for the /etc file system.
  - For the CIM element, the space required for the /var VARWBEM file system is 165 cylinders primary, 16 cylinders secondary.
  - For Predictive Failure analysis, a separate file system is created and mounted at mountpoint the /var/pfa. The total space required for zFS is 300 cylinders primary; 50 cylinders secondary.
  - For z/OSMF additional file system space is needed for the repositories. Refer to your z/OSMF ServerPac Workflow (sample jobs provided).

z/OS Driving System Requirements

The *driving system* is the system image (hardware and software) that you use to install the target system. The *target system* is the system software libraries and other data sets that you are installing. You log on to the driving system and run jobs there to create or update the target system. Once the target system is built, it can be IPLed on the same hardware (same LPAR or same processor) or different hardware than that used for the driving system.

If your driving system will share resources with your target system after the target system has been IPLed, *be sure to install applicable coexistence service* on the driving system before you IPL the target system. If you don't install the coexistence service, you will probably experience problems due to incompatible data structures (such as incompatible data sets, VTOCs, catalog records, GRS tokens, or APPC bind mappings).

Customized Offerings Driver (5751-COD)

The Customized Offerings Driver V3.1 (5751-COD) is an entitled driving system you can use if:

- you don't have an existing system to use as a driving system, or
- your existing system does not meet driving system requirements and you don't want to upgrade it to meet those requirements.

This driver is currently a subset of a z/OS V2.4 system and is available on a DVD or electronically. This COD activates z/OSMF so that you can use it as a driving system for a z/OSMF ServerPac, if you don't have a driving system which has z/OSMF.

The Customized Offerings Driver requires three DASD volumes configured as 3390-9, or larger; a non-Systems Network Architecture (SNA) terminal used for a z/OS MVS™ system console; and a locally attached SNA terminal for a Time Sharing Option Extended (TSO/E) session. Also, if you select tape media, a tape drive that can read 3590 or 3592 tape is required. The Customized Offerings Driver can also be ordered on DVDs, which removes the requirement for a tape drive.

The Customized Offerings Driver is intended to run in single-system image and monoplex modes only. Its use in multisystem configurations is not supported. The Customized Offerings Driver is intended to be used only to install new levels of z/OS using ServerPac or CBPDO, and to install service on the new software until a copy (clone) of the new system can be made. The use of the Customized Offerings Driver for other purposes is not supported.

As of z/OS V2R2, the Customized Offerings Driver Installation Guide is no longer shipped in hardcopy format. Instead, this publication is shipped in PDF format on a separate DVD.

The Customized Offerings Driver includes a zFS file system and the necessary function to use Communications Server (IP Services), Security Server, and the system-managed storage (SMS) facility of DFSMSdfp, but these items are not customized. However, existing environments can be connected to, and used from, the Customized Offerings Driver system for the purposes of installation.

Identifying Driving System Software Requirements for ServerPac for z/OS 3.1

Driving system requirements for installing z/OS 3.1 by way of ServerPac or dump-by-data-set SystemPac are:

- *An operating system:* Use either of the following:
  - A supported z/OS release (V2.4 or later), with the following PTFs installed, which have been identified with the FIXCAT of IBM.DrivingSystem-RequiredService. z/OSMF should be activated, with the z/OSMF Software Management plug-in available.
  - The Customized Offerings Driver V3 (5751-COD).

- *A terminal:* A locally-attached or network-attached terminal that can be used to establish a TSO/E session on the IPLed system is required.

- *Proper authority:* Use the RACFDRV installation job as a sample of the security system definitions required so that you can perform the installation tasks.

- Proper security:
  - To deploy a ServerPac Portable Software Instance with z/OSMF Software Management, observe the following requirements:

– **The user ID that you use must have READ access to the SAF (System Authorization Facility) resource that protects the IBM data sets that are produced during the creation of the ServerPac portable software instance. That is, your user ID requires READ access to data set names that begin with CB.OS* and CB.ST*.**

 In order for you to install into the zFS, the user ID you use must have read access to the SUPERUSER.FILESYS.PFSCTL resource in the RACF FACILITY class.

 In order for you to install the z/OS UNIX files, the following is required:

- The user ID you use must be a superuser (UID=0) or have read access to the BPX.SUPERUSER resource in the RACF facility class.
- The user ID you use must have read access to facility class resources BPX.FILEATTR.APF, BPX.FILEATTR.PROGCTL, and BPX.FILEATTR.SHARELIB (or BPX.FILEATTR.* if you choose to use a generic name for these resources). The commands to define these facility class resources are in SYS1.SAMPLIB member BPXISEC1.

    - Group IDs uucpg and TTY, and user ID uucp, must be defined in your security database. These IDs must contain OMVS segments with a GID value for each group and a UID value for the user ID. (For ease of use and manageability, define the names in uppercase.)
        - The group ID and user ID values assigned to these IDs cannot be used by any other IDs. They must be unique.
- You must duplicate the required user ID and group names in each security database, including the same user ID and group ID values in the OMVS segment. This makes it easier to transport the HFS data sets from test systems to production systems. For example, the group name TTY on System 1 must have the same group ID value on System 2 and System 3. If it is not possible to synchronize your databases you will need to continue running the FOMISCHO job against each system after z/OS UNIX is installed.

    If names such as uucp, uucpg, and TTY are not allowed on your system, or if they conflict with existing names, you can create and activate a user ID alias table. For information about defining these group and user IDs to RACF and about creating a user ID alias table (USERIDALIASTABLE), see *z/OS UNIX System Services Planning*. Other sources of information are SYS1.SAMPLIB member BPXISEC1. (**Note:** You can use the RACFDRV installation job as a sample of the security system definitions required to perform the installation tasks.)

- *Language Environment run-time options:* As of z/OS R7, ServerPac requires that the following Language environment run-time options are **not** specified as nonoverrideable (NONOVR) in the CEEDOPT CSECT: ALL31, ANYHEAP, BELOWHEAP, DEPTHCONDLIMIT, ERRCOUNT, HEAP, HEAPCHK, HEAPPOOLS, INTERRUPT, LIBSTACK, PLITASKCOUNT, STACK, STORAGE, THREADHEAP, and THREADSTACK .

- *Language Environment:* The CustomPac Installation Dialog uses the Language Environment run-time library SCEERUN. If SCEERUN is not in the link list on the driving system, you must edit the ServerPac installation jobs to add it to the JOBLIB or STEPLIB DD statements.

- *OMVS address space active:* For ServerPac only an activated OMVS address space with z/OS UNIX kernel services operating in full function mode is required.

- *SMS active:* The Storage Management Subsystem (SMS) must be active to allocate zFS and PDSE data sets, whether they are SMS-managed or non-SMS-managed. Also, the use of zFS data sets is supported only when SMS is active in at least a null configuration, even when the data sets are not SMS-managed. Do either of the following:
    - To allocate non-SMS-managed zFS and PDSE data sets, you must activate SMS on the driving system in at least a null configuration. You must also activate SMS on the target system.
    - To allocate SMS-managed zFS and PDSE data sets, you must activate SMS on the driving system in at least a minimal configuration. Then you must define a storage group, create SMS-managed volumes, and write, translate, and activate a storage class ACS routine that allows the allocation of PDSE and zFS data sets. You must also activate SMS on the target system.

- For any zFS data sets that exceed the 4 GB size limit, you must define an SMS Data Class with extended format and extended addressability. **z/OS 3.1 ships with a version root file system that is extremely close to 4GB in size. If you merge other zFS data sets with this version root, it will exceed 4 GB in size.**

If you intend to receive your order using Secure FTP (FTPS)

To use Secure FTP (FTPS) to download the ServerPac portable software instance files from the IBM Download Server by using the z/OSMF Software Management **Add > From Download Server** action, you require the following:

- z/OS SMP/E V3R7 in z/OS V2R4 or higher
- Either of the following features is installed, which enables strong cryptographic ciphers to be used for SSL/TLS connections in non-FIPS mode:
    - System SSL Security Level 3 Feature
    - CP Assist for Cryptographic Functions (CPACF) DES/TDES Enablement Feature 3863 with PTF UA95810.
- ICSF is configured and active, or either of the following SDK levels is installed:
    - IBM 31-bit SDK for z/OS Java Technology Edition V8.0 (5655-DGG) or later
    - IBM 64-bit SDK for z/OS Java Technology Edition V8.0 (5655-DGH) or later

This function enables SMP/E to calculate SHA-1 hash values to verify the integrity of data that is being transmitted. If ICSF is not configured and active, SMP/E uses its Java application class instead for calculating the SHA-1 hash values. IBM recommends the ICSF method because it is likely to perform better than the SMP/E method. For information about how to configure and activate ICSF, see *z/OS Cryptographic Services ICSF System Programmer's Guide*

- A download file system. The *Download to host - View Server XML* link on the Shopz order download page contains order size information.
- Ensure that the DigiCert Global Root CA certificate (and the Root 2 - GeoTrust Global CA Certificate) is connected to your security manager key ring or stored in your default Java keystore file and is trusted on your system. Also, ensure that the user ID that runs SMP/E is authorized to use the key ring or default Java keystore file.
- Ensure that the FTP.DATA data set statements that are used to receive your order are set for your environment. For example, an FTPKEEPALIVE statement with a value of 0 (the default) can cause an FTP control connection to expire in some environments. Also, the security manager key ring file that is specified by the key ring statement in the FTP.DATA file might require certificates to be added.
- Firewall configuration. If your enterprise requires specific commands to allow the download of your ServerPac portable software instance files through a local firewall, you must identify these commands when you specify the Client XML information on the z/OSMF Software Management *Add Portable Software Instance from a Download Server page.*

If you intend to download your order using HTTP Secure (HTTPS):
If you intend to download your order through HTTP Secure (HTTPS) by downloading the ServerPac portable software instance files from the IBM Download Server by using the z/OSMF Software Management **Add -> From Download Server** action, you need the following:

- z/OS SMP/E V3R7 in z/OS V2R4 or higher
- SMP/E uses the services of either of the following SDK levels:
    - IBM 31-bit SDK for z/OS Java Technology Edition V8.0 (5655-DGG) or later
    - IBM 64-bit SDK for z/OS Java Technology Edition V8.0 (5655-DGH) or later
- A download file system. The *Download to host - View Server XML* link on the Shopz order download page contains order size information.
- Ensure that the DigiCert Global Root CA certificate and the Root 2 - GeoTrust Global CA Certificate is connected to your security manager key ring or stored in your default Java keystore file. It must be trusted on your system. Also, ensure that the user ID that runs SMP/E is authorized to use the key ring or default Java keystore file.
- HTTP or SOCKS Proxy Server configuration. Your enterprise might require specific commands to allow the download of your ServerPac portable software instance files through an HTTP or SOCKS Proxy Server. If so, you must identify these commands when you enter the Client XML information on the z/OSMF Software Management *Add Portable Software Instance from a Download Server page*.

If you intend to receive your order by way of DVD, you need the following:

- Order available on z/OS host system. To make the order available on your z/OS host system, upload the order to the z/OS host from the DVD(s). Refer to readme.pdf on the first DVD to find the various methods for making your order available on your z/OS host system.
- Space requirements on z/OS. Ensure you have the required space on your z/OS host system. To get the actual size of the order, refer to dialog.txt on the first DVD.
- Space requirements on a workstation. If you chose to copy your order from the DVD(s) to a workstation before uploading the contents to your z/OS host system, ensure you have the required space available on your workstation.

If you intend to verify the digital signature of the ServerPac portable software instance package by using the z/OSMF Software Management **Add -> From Download Server** action, you need the following:

- z/OSMF with APAR PH49385
- z/OS SMP/E V3R7, plus the PTFs that are identified with the SMP/E fix category IBM.DrivingSystem-RequiredService
- STG Code Signing Certificate Authority - G2 certificate added to your security manager database
- STG Code Signing Certificate Authority - G2 certificate connected to that security manager key ring that you will use when validating package signing certificates and the package signatures

Driving system requirements for Validated Boot for z/OS

To signing in-scope IPL artifacts for Validated Boot for z/OS, you must satisfy the following requirements on the driving system:

- z/OS V2R5 or later, plus the PTFs that are identified with the following SMP/E FIXCAT: IBM.Function.ValidatedBoot
- Signing certificate is set up on the driving system.

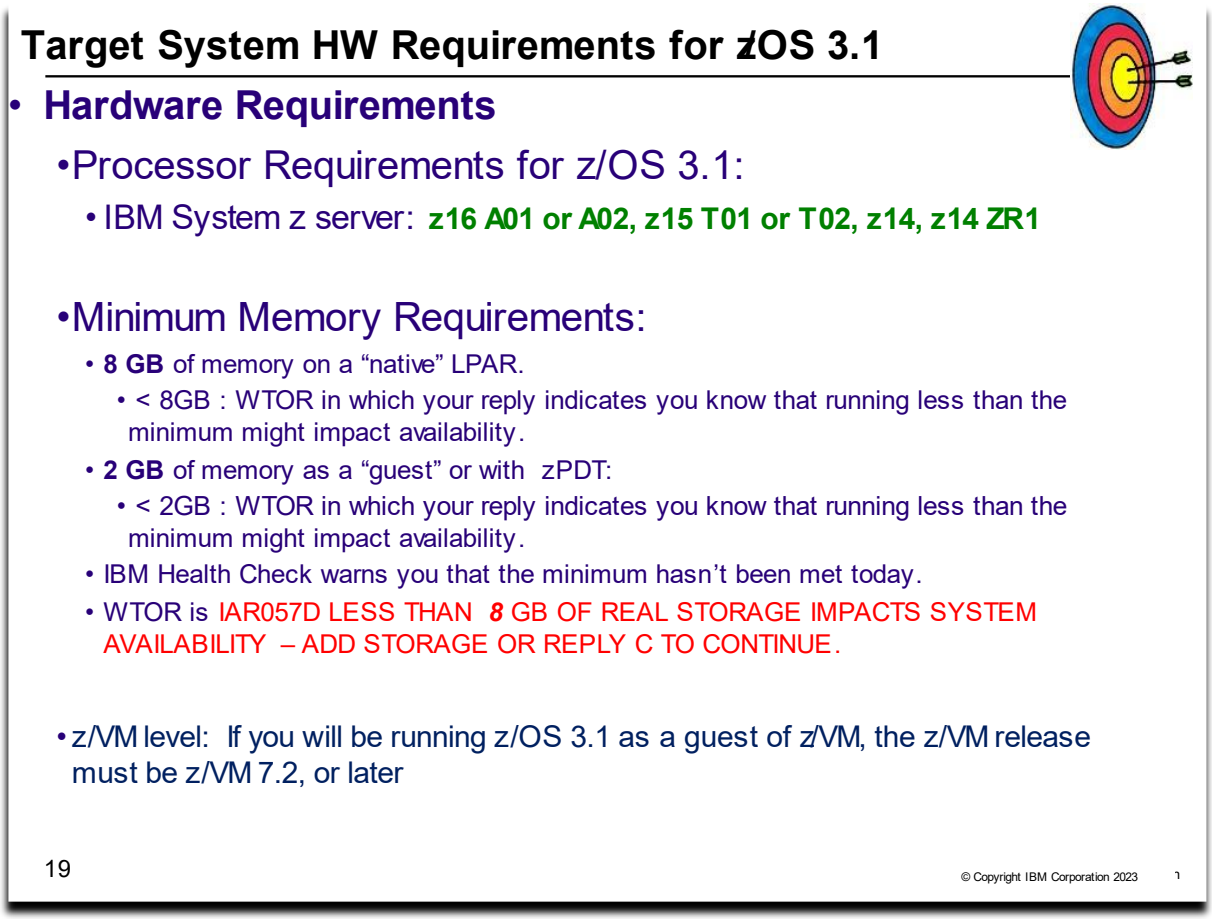
Use the SMP/E REPORT FIXCAT command to verify that all required PTFs are installed on your driving system.

To perform the validation of signatures, your target system must meet a separate set of requirements, including an IBM z16 with the appropriate microcode level, HMC security, and z/OS V2.5 or later with the PTFs that are identified with the SMP/E FIXCAT: IBM.Function.ValidatedBoot.

- For information about how to get started with Validated Boot for z/OS, see Validated Boot for z/OS (www.ibm.com/support/z-content-solutions/validated-boot-for-zos/) content solution.
- For information about setting up Validated Boot for z/OS, see the white paper, z/OS Validated Boot (ibm.biz/zosValidatedBoot).
- For the collected z/OS publication updates for Validated Boot for z/OS, see Validated Boot for z/OS

Proper level for service

In order for you to install service on the target system that you're building, your driving system must minimally meet the driving system requirements for CBPDO Wave 1 and must have the current (latest) levels of the program management binder, SMP/E, and HLASM. Another way to install service is from a copy of your target system.



Target System Hardware Requirements
The minimal hardware requirements for z/OS, as well as additional hardware needed by specific z/OS elements and features, are documented in *z/OS Planning for Installation*.

Identifying Processor Requirements
z/OS 3.1 supports these System z server models:

- IBM z16 A01 and A02
- IBM z15 T01 and T02
- IBM z14 and z14 ZR1

The following IBM System z servers, and earlier servers, are not supported with **z/OS 3.1** and later releases:

- IBM z13 and z13s
- IBM zEnterprise zBC12 and zEC12

Important: If you IPL z/OS on a server that it does not support, you might receive wait state 07B. The number of IEA434I messages is limited to 32 during IPL/NIP to avoid exhausting initial ESQA. An IEA444I message will be reported one time during IPL/NIP to indicate that additional IEA434I messages have been suppressed: IEA444I NUMBER OF IEA434I MESSAGES EXCEEDS NIP MAXIMUM .

z/OS 3.1 needs these machine facilities to IPL:

- The entropy-encoding-compression facility.
- The miscellaneous instruction extension 2 facility.
- The instruction execution protection facility.
- The vector binary-coded-decimal facility.
- The vector enhancements facility 1.

Identifying Minimum Storage Requirements as of z/OS 3.1

IBM z/OS 3.1 requires a minimum of 8 GB of memory. When running as a z/VM guest or on a IBM System z Personal Development Tool, a minimum of 2 GB will be required. If the minimum is not met, a warning WTOR will be issued at IPL. Continuing with less than the minimum memory could impact availability. An IBM health check warns you when an LPAR has been configured with less than 8 GB.

Identifying Coupling Facility Requirements

There are coupling facility level (CFLEVEL) considerations. See https://www.ibm.com/downloads/cas/EEGKM5OM

When you change coupling facility control code (CFCC) levels, your coupling facility structure sizes might change. If, as part of your upgrade to a z16 server, you change CFCC levels, you might have larger structure sizes than you did previously. If your CFCC levels are identical, structure sizes are not expected to change when you migrate from a previous server to a newer generation server.

In addition, at CF Level 17 and later, ensure that the CF LPAR has at least 512MB of storage.

If you are moving your coupling facilities and the coupling facility structures will be on higher CFCC levels than they were previously, run the Coupling Facility Structure Sizer (CFSIZER) tool to find out if you have to increase coupling facility structure sizes. Prepare to make the necessary changes to the CFCC level as indicated by the tool.
You can download the CFSIZER tool at Coupling Facility sizer (http://www.ibm.com/systems/support/z/cfsizer/).

Note: After you make a coupling facility available on the new hardware, you can run the Sizer utility, an authorized z/OS program, to evaluate structure size changes. The Sizer utility is distinct from CFSizer, and should be run after the new hardware (CFLEVEL) is installed, but before any CF LPAR on the new hardware is populated with structures. You can download the Sizer utility at http://www.ibm.com/systems/support/z/cfsizer/altsize.html.

- IBM z16 A01 and A02 servers initially ship with CFCC level 25.
- IBM z15 T01 and T02 servers initially ship with CFCC level 24.
- IBM z14 ZR1 servers initially ship with CFCC level 23.
- IBM z14 servers initially ship with CFCC level 22.
- IBM z13s servers initially ship with CFCC level 21.
- IBM z13 servers initially ship with CFCC level 20.

Identifying z/VM Requirements

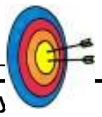
If you will be running IBM z/OS 3.1 as a guest under IBM z/VM, the z/VM release must be z/VM 7.2, or later.

Software Requirements for z/OS 3.1

- **Software Requirements:**
  - **Coexistence Software** (on other z/OS systems)
  - **Target Software** (correct levels of IBM non-z/OS and non-IBM products on the z/OS system)
    - ***Generally,* you may use the product levels on z/OS V2.5 from your prior system, as long as those product levels are still service supported.**
    - Note, however, that if you are using any of the functions in *z/OS 3.1 Planning for Installation*, Appendix B**, verify that those functional requirements are satisfied.**
- At GA, z/OS 3.1 has an *overall* dependency on ***"IBM Semeru11 Runtime Certified Edition" (64-bit only)***
  - Some z/OS 3.1 functions still require IBM Java SDK 31 -bit V8, but are planned to be converted to IBM Semeru 11.
  - IBM 64-bit SDK for z/OS V8 and IBM 31 -bit SDK for z/OS V8 are supported for applications for as long as they remain supported.
  - IBM has issued a statement of direction indicating a future plan to deliver IBM Semeru Java 17.

20

Selected z/OS 3.1 SW Functional Requirements



Function	Minimum functional dependency
Most z/OS 3.1 functions at GA: z/OSMF, SDSF, RACF, Communications Server, SCRT, HCD, ...	IBM Semeru 11 for z/OS (5655-DGJ)
z/OS XML System Services, CPM, and Infoprint Server (at GA)	IBM Java SDK 31 -bit V8 (5655-DGG)
IBM Security zSecure™ products: • IBM Security zSecure Adapters for QRadar® SIEM (5655-AD8) • IBM Security zSecure Admin (5655-N16) • IBM Security zSecure Audit (5655-N17) • IBM Security zSecure Command Verifier (5655-N19) • IBM Security zSecure Visual (5655-N20) • IBM Security zSecure Alert (5655-N21)	3.1 level at a minimum. Earlier levels of this product are not supported for use with z/OS 3.1.
IBM Tivoli Event Pump for z/OS (5698-B34)	IBM Tivoli Event Pump for z/OS, Version 4.2.2 requires PTF UA92963 for APAR OA51799. Prior releases (V4R2.0 and V4R2.1) of IBM Tivoli Event Pump for z/OS do not have a corresponding fix and therefore do not run on z/OS 3.1

→Use FIXCAT `IBM.TargetSystem-RequiredService.z/OS.3.1` for program product PTF dependencies.

NEW!

→Use FIXCAT `IBM.TargetSystem-RequiredService.Semeru.*` for z/OS 3.1 PTF dependencies when the Java level dependency changes during z/OS 3.1.

22→Website http://www-306.ibm.com/software/support/lifecycle/ can be helpful.

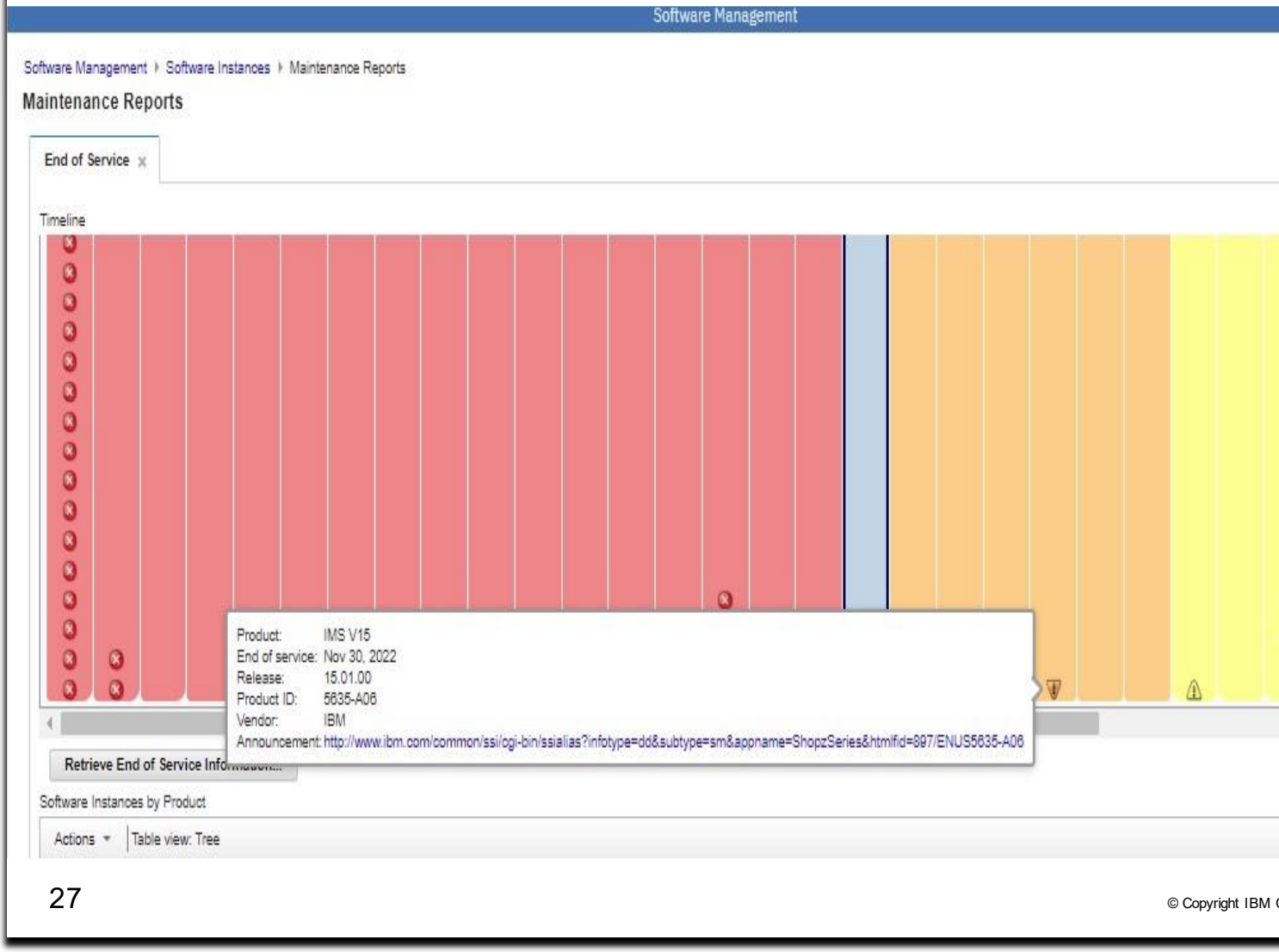
Choosing IBM Products That You Want to Run with z/OS
You must determine the minimum product release levels and release levels for functional requirements. IBM middleware and application products require a specific level (version, release, or PTF) so that the products will run on z/OS 3.1. You cannot use the FIXCAT support to determine these release levels. Instead, you can refer to *z/OS 3.1 Planning for Installation,* Appendix B, for the functions of z/OS that require specific z/OS optional features, IBM middleware products, or IBM application products.

If you are upgrading from z/OS V2.4 or z/OS V2.5, you may generally use the product levels on z/OS 3.1 that you used on your prior z/OS release, as long as the product levels are still service-supported.
Note, however, that if you are using any of the functions in *z/OS Planning for Installation,* Appendix B, and those functions have dependencies on IBM middleware or application products, you must use the product levels shown (or later).

Many of these products can be ordered as part of your z/OS ServerPac portable software instance order. Note that there may be differences between what is minimally service supported, what is minimally supported with z/OS 3.1s, and what is currently orderable.

If you're upgrading to z/OS 3.1, you can find out which products have new levels by using Shopz and loading your current inventory to see what higher levels of products are orderable.

➔Best way to check End of Service products!



Tip! Finding End of Service Dates for IBM Products
A handy website for finding end of service dates for IBM products is *http://www.ibm.com/software/support/lifecycle/* .

An especially useful way of identifying if any of the products you are approaching or have met end of service is to use z/OSMF Software Management, and look at the End of Service report!

Target System PTF Verification for z/OS 3.1

1. **RECEIVE the latest HOLDDATA. (If you pull HOLDDATA from the ftp website, make sure you use FULL!)**
   - **HOLDDATA is produced to associate a particular PTF with a minimum or functional level for z/OS 3.1**

```
++HOLD(HRKN560) FIXCAT FMID(HRKN560) REASON(AA64385)
RESOLVER(UJ92664)
 CATEGORY(IBM.TargetSystem-RequiredService.z/OS.3.1)
 DATE(23117).
```

2. **Run the REPORT MISSINGFIX command* to see what is needed, but not yet installed.**

```
SET BDY(GLOBAL).     /* Your program product global zone */
REPORT MISSINGFIX ZONES(pp_tgt)
FIXCAT(IBM.TargetSystem-RequiredService.z/OS.3.1).
```

* if you have your target system defined as a z/OSMF Software Management software instance (easy to do!), you can use the z/OSMF's Maintenance Reports → Missing FIXCAT SYSMODs in a couple of clicks!

23

Programmatic Help with Target System PTF Verification for z/OS 3.1

The IBM PTFs needed to support z/OS 3.1 are identified with a FIXCAT called **IBM.TargetSystem-RequiredService.z/OS.3.1,** in Enhanced HOLDDATA. You must use the SMP/E REPORT MISSINGFIX command to help identify those PTFs on your current system which would be needed for your upgrade to z/OS 3.1.

It is a good idea to periodically re-run the REPORT MISSINGFIX command to determine if any new PTFs have been identified that you are missing.

Coexistence System PTF Verification for z/OS 3.1

1. **RECEIVE the latest HOLDDATA. (If you pull HOLDDATA from the ftp website, make sure you use FULL!)**
   - **HOLDDATA is produced to associate a particular PTF as coexistence between z/OS V2.4 or V2.5, with z/OS 3.1**

```
++HOLD(HBB77D0) FIXCAT FMID(HBB77D0) REASON(DA63269)
RESOLVER(UJ93003) CATEGORY(  IBM.Coexistence.z /OS.3.1)
 DATE(23165).
```

2. **Run the REPORT MISSINGFIX command* to see what is needed, but not yet installed.**

```
SET BDY(GLOBAL). /* Your z/OS V2.5 global */
REPORT MISSINGFIX ZONES(ZOS25T)
FIXCAT(IBM.Coexistence.z/OS.3.1,
       IBM.Function.HealthChecker).
```

For the upgrade health checks ☺!

* if you have your coexisting system defined as a z/OSMF Software Management software instance (easy to do!), you can use z/OSMF's Maintenance Reports→ Missing FIXCAT SYSMODs, in just a couple of clicks.

25

Using FIXCAT for coexistence PTFs for z/OS 3.1
For coexistence verification for z/OS 3.1, the fix category of interest is **IBM.Coexistence.z/OS.3.1**. You can use the FIXCAT of ++HOLD statement to identify APARs, their fix categories, and the PTF that resolves the APAR. Another fix category that is helpful when doing the coexistence verification is **IBM.Function.HealthChecker**, for verifying that you've got the latest migration IBM Health Checks for z/OS installed on your coexisting system.

When FIXCAT HOLDDATA statements are received into a global zone, SMP/E assigns the fix category values as sourceids to the PTFs that resolve the APARs. These sourceids then simplify selecting and installing required fixes. During APPLY and ACCEPT command processing you can specify the assigned sourceids on the SOURCEID and EXSRCID operands to select the SYSMODs associated with a particular fix category.

In addition, for the APPLY and ACCEPT commands you can specify which Fix Categories are of interest using the FIXCAT operand. This tells SMP/E to process only FIXCAT HOLDDATA for the categories you specify, and all others are ignored.

Finally, SMP/E uses the FIXCAT HOLDDATA to identify what required fixes are missing. The REPORT MISSINGFIX command analyzes the FIXCAT HOLDDATA and determine which fixes (APARs) identified by the HOLDDATA are not yet installed. Only the fixes associated with the fix categories of interest to you, specified by you, are analyzed and identified. For example, you can identify only the missing fixes associated with a particular hardware device or coexistence for a specific new software release.

Note that you can use wildcards in the FIXCAT name in the REPORT MISSINGFIX command. For example, if you wanted to verify coexistence for z/OS V2.4 as well as z/OS V2.5 on your z/OS V3.1 system, your command could be:

```
REPORT MISSINGFIX ZONES(ZOS24T) FIXCAT(IBM.Coexistence.z/OS.*,
                                       IBM.Function.HealthChecker).
```

Do notice though, that z/OS V2.5 coexistence fixes have also been "backmarked" for z/OS 3.1 coexistence, so it is not necessary to specify interim z/OS releases for coexistence verification in the REPORT MISSINGFIX command.

Some Upgrade Tasks You Can Do NOW

1. **Transition off of removed functions and elements:**
   - In V2.5: HFS
   - In 3.1: JES3, BDT, ISFPARMS assembler macros,…
2. **z/OSMF is a driving system requirement for all ServerPacs**
3. **Prepare Target Systems:**
   - Target systems -
     - HW: z14 or higher, with 8GB “native”.
     - SW: Java V11 is the general requirement for z/OS 3.1 functions, some still use Java 8 31-bit for now.
     - DASD storage for z/OS 3.1 – root file system is very close to 4GB
     - IBM product level research: FIXCAT **IBM.TargetSystem-RequiredService.z/OS.3.1**
     - Find z/OS 3.1 PTFs needed for Semeru FIXCAT **IBM.TargetSystem-RequiredService.Semeru.***
     - ISV research for z/OS 3.1
   - Coexisting systems - FIXCAT **IBM.Coexistence.z/OS.3.1**
4. **Use IBM Health Checker for z/OS** FIXCAT **IBM.Function.HealthChecker**
   - Activate the Upgrade Health Checks (they are shipped INACTIVE).
5. **You need z/OSMF to use the z/OS 3.1 Upgrade Workflow.**
   - **No migration book!**
6. **Perform z/OS 3.1 upgrade actions you can do NOW.**

26

Prepare for your upgrade to z/OS 3.1!

In this presentation you’ve seen many things you can do right now, on your current z/OS release to help make your z/OS 3.1 upgrade smooth. Listed above are a recap of the things that were shown in this presentation, but make sure you review the upgrade actions in the z/OS Upgrade Workflow so that you know a more complete upgrade picture.

IBM Health Checker for z/OS Checks for Upgrade

1. ZOSMIGREC_ROOT_FS_SIZE
2. XCF_SYSPLEX_CDS_CAPACITY
3. XCF_SYSSTATDET_PARTITIONING
4. RSM_MEMLIMIT
5. ALLOC_TAPELIB_PREF
6. SUP_ASVT_ABOVE_16M
7. ZOSMIGV2R4_NEXT_WLM_ServCoeff
8. ZOSMIGV2R4_NEXT_VSM_CHECKREGINLOSS
9. JES2_UPGRADE_CKPT_LEVEL_JES2
10. SDSF_ISFPARMS_IN_USE
11. SDSF_CLASS_SDSF_ACTIVE
12. RMF_DDS_OPTS
13. USS_HFS_DETECTED
14. ZOSMIGV2R4_NEXT_CS_OSIMGMT
15. ZOSMIGV2R4_NEXT_CS_DCAS_NTVSSL
16. ZOSMIGV2R4_NEXT_CS_TN3270_NTVSSL
17. ZOSMIGV2R4_NEXT_CS_FTPSRV_NTVSSL
18. ZOSMIGV2R5_NEXT_CS_LSA
19. ZOSMIGV2R5_NEXT_CS_OSADLH
20. *OpenSSH Config check planned*
21. ISPF_WSA

These health checks can be directly invoked by the z/OS 3.1 Upgrade Workflow when using z/OSMF!

27

Using IBM Health Checker for z/OS for upgrade purposes
The IBM Health Checker for z/OS infrastructure is exploited for upgrade purposes. Health Checks that are helpful for determining upgrade action applicability are provided. These checks ("Migration Health Checks") should be used prior to your upgrade to the new z/OS release to assist with your upgrade planning, and re-run after your upgrade to verify that the upgrade action was successfully performed. As with any Health Check, no updates are performed to the system. Migration Health Checks only report on the applicability of a specific upgrade action on a system; and only report on the currently active system.

Details on how to run the Migration Health Checks are provided in beginning of the *z/OS Upgrade Workflow*.

System REXX health check considerations
All exploiters of the System REXX support in z/OS require that the System REXX customization be performed. Using the IBM Health Checker for z/OS health checks is one example of possible System REXX exploitation. In particular, any compiled REXX execs must have the proper runtime support available from the Alternate Library for REXX (available in z/OS since V1R9) or from the IBM Library for REXX on zSeries (5695-014). Several IBM Health Checker for z/OS migration health checks have been written in compiled System REXX. These health checks rely upon the System REXX customization and runtime activities being completed. If System REXX (and the security environment that System REXX requires) have not been properly customized, then System REXX health checks will not execute successfully.

- For System REXX customization activities, refer to "System REXX" in *z/OS MVS Programming: Authorized Assembler Services Guide*.
- For compiled REXX exec runtime availability, see "Alternate Library for REXX Customization Considerations" in *z/OS Program Directory*, or refer to product documentation accompanying IBM Library for REXX on zSeries.

Migration Health Checks and Best Practice Health Checks
Migration Health Checks are not different from other Health Checks, but they do have some characteristics which allow them to be uniquely identified.

For z/OS, the convention is **ZOSMIGVvvRrr (or ZOSMIGnn)**_*component_program_name* The names of migration checks begin with the characters **ZOSMIG**. Following this prefix is a value to help you plan the timing of the upgrade action, as follows:

- **ZOSMIGVvRr_Next :** Upgrade action is recommended, but will become a required upgrade action in the release after V*v*R*r* .
- **ZOSMIGVvRr_Next2 :** Upgrade action is recommended, but will become a required upgrade action two releases after V*v*R*r*.
- **ZOSMIGVvRr :** Upgrade action is required in the release indicated by V*v*R*r*.
- **ZOSMIGVvRrPREV**: Upgrade action is required in the release indicated by VvRr and in prior releases, with the appropriate service.
- **ZOSMIGREC :** Upgrade action is recommended for the foreseeable future. The upgrade action might never be required.
- **ZOSMIGREQ :** Upgrade action that is recommended now, but will be required in a future release.

For the ICSF element while Web deliverables are used, the convention is **ICSFMIGnnnn**_*component_program_name*). When ICSF does not provide a Web deliverable, it uses the **ZOSMIG** standard.

Migration health checks are shipped with a status of **INACTIVE** by default. Because you may not want to know about upgrade actions during non-upgrade periods, Migration Health Checks will not automatically be active.

There are Best Practice Health Checks that can help with upgrade actions, and yet they do not have the Migration Health Check naming convention. That is because the component owners felt that the practice is recommended for reasons above and beyond upgrade purposes. All Health Checks (whether they are Migration Health Checks or Best Practice Health Checks) will be cross-referenced in the *z/OS Upgrade Workflow* when they can assist with a specific upgrade action. *Be aware, your upgrade assistance is not just limited to the checks that follow the Migration Health Check naming convention!*

For description of the health checks above, which release they run on, and their severity refer to the z/OS *Health Checker User's Guide.*

Finding the latest z/OS 3.1 Upgrade Workflow

- The z/OS 3.1 Upgrade Workflow is part of z/OS!!
  - Not supplied via GitHub anymore.
  - IBM Service will also support this workflow
  - We still welcome any feedback or comments to zosmig@us.ibm.com.
- The files needed to create your workflow can be found in the directory **/usr/lpp/bcp/upgrade**
  - zOS_3.1_from_V2.5_UpgradeWorkflow.xml
  - zOS_3.1_from_V2.4_UpgradeWorkflow.xml
  - z15_zOS_Upgrade_Workflow.xml
  - z16_zOS_Upgrade_Workflow.xml

  HW info not included in z/OS info anymore
  - When creating your workflow, choose the appropriate file for the system you are upgrading from.
- The initial workflows are shipped back to z/OS V2.4 and V2.5 with a PTF, and found with the FIXCAT **`IBM.Coexistence.z/OS.3.1`**
- Updates to the z/OS 3.1 Upgrade Workflow, will also have the same FIXCAT.
  - Use “Create new based on existing” to pick up updates.

27

Upgrade Workflow	Where to find	SMP/E FIXCATs
z/OS 2.4 to 3.1	On z/OS in /usr/lpp/bcp/upgrade/zOS_3.1_from_V2.4_Upgrade_Workflow.xml	IBM.Coexistence.z /OS.3.1
z/OS 2.5 to 3.1	On z/OS in /usr/lpp/bcp/upgrade/zOS_3.1_from_V2.5_Upgrade_Workflow.xml	IBM.Coexistence.z /OS.3.1
z/OS 2.3 to z/ OS 2.5	On z/OS in /usr/lpp/bcp/upgrade/zOS_V2.5_from_V2.3_Upgrade_Workflow.xml	IBM.Coexistence.z /OS.V2R5
z/OS 2.4 to z/ OS 2.5	On z/OS in /usr/lpp/bcp/upgrade/zOS_V2.5_from_V2.4_Upgrade_Workflow.xml	IBM.Coexistence.z /OS.V2R5
z/OS 2.2 to z/ OS 2.4	On github - https://github.com/IBM/IBM -Z-zOS/tree/main/zOS - Workflow/zOS%20V2.4%20Upgrade%20Workflow	N/A
z/OS 2.3 to z/ OS 2.4	On github - https://github.com/IBM/IBM -Z-zOS/tree/main/zOS - Workflow/zOS%20V2.4%20Upgrade%20Workflow	N/A
z16	On z/OS in /usr/lpp/bcp/upgrade/z16_zOS_Upgrade_Workflow.xml (And combined into /usr/lpp/bcp/upgrade/zOS_V2.5_from_V2.3_Upgrade_Workflow.xml /usr/lpp/bcp/upgrade/zOS_V2.5_from_V2.4_Upgrade_Workflow.xml)	IBM.Device.Server.z16.3931.RequiredService (IBM.Coexistence.z /OS.V2R5)
z15	On z/OS in /usr/lpp/bcp/upgrade/z15_zOS_Upgrade_Workflow.xml (And combined into /usr/lpp/bcp/upgrade/zOS_V2.5_from_V2.3_Upgrade_Workflow.xml /usr/lpp/bcp/upgrade/zOS_V2.5_from_V2.4_Upgrade_Workflow.xml) On github - https://github.com/IBM/IBM -Z-zOS/tree/main/zOS - Workflow/zOS%20z15%20Workflow	IBM.Device.Server.z15.8561.RequiredService (IBM.Coexistence.z /OS.V2R5)
z14	On github - https://github.com/IBM/IBM -Z-zOS/tree/main/zOS - Workflow/zOS%20z14%20Workflow	N/A
Z13	On github - https://github.com/IBM/IBM -Z-zOS/tree/main/zOS - Workflow/z13%20Workflow	N/A

Very Important Links:
Exported version of the z/OS 3.1 Upgrade Workflow will be available on IBM Documentation for z/OS 3.1, under "System Level" category.

Exported V2.4 to V2.5 Upgrade Workflow:
https://www.ibm.com/docs/en/zos/2.4.0?topic=SSLTBW_2.4.0/com.ibm.zos.v2r4.e0zm100/Export_zOS_V2R4_to_V2R5_Upgrade_Workflow.html

Exported V2.3 to V2.5 Upgrade Workflow:
https://www.ibm.com/docs/en/zos/2.4.0?topic=SSLTBW_2.4.0/com.ibm.zos.v2r4.e0zm100/Export_zOS_V2R3_to_V2R5_Upgrade_Workflow.html

Content Solution webpage for learning in an easy way about ServerPac packaged as a z/OSMF Portable Software **Continuing the advancement in z/OS upgrade assistance!** https://www.ibm.com/support/z-content-solutions/serverpac-install-zosmf/

We have two z/OS Management Facility (z/OSMF) z/OS Upgrade Workflow versions, one for the n-1, and one for the n-2 path. Using the z/OSMF workflow, you can go through a z/OS 3.1 upgrade as an interactive, step-by-step process. Depending on your z/OS 3.1 upgrade path, you select the file you will need.

In z/OS 3.1 (continuing what was introduced in V2.2) the z/OS Upgrade Workflow has the capability to invoke IBM health checks directly from the step, and also provides the optional capability to give feedback on your upgrade experience. The z/OS 3.1 Upgrade Workflow is supported by the IBM Service organization, and provided in PTF(s). We also do welcome suggestions or comments to email zosmig@us.ibm.com.

The z/OS 3.1 Upgrade Workflow has the ability to discover used z/OS priced features (continuing what was introduced in V2.3), and some other features, on your system. This is very helpful, because if you are not using a certain feature, then why have to manually skip those steps yourself? The z/OSMF Workflow can identify many features that you might not be using, and automatically skip them in the Workflow for you, giving you less steps to perform. Added in z/OS 3.1, is the ability to reduce the number of applicable steps by seeing if PTFs that included HOLD ACTIONs were already performed. In addition, coexistence PTF verification can be done from the Workflow, via an included SMP/E REPORT MISSINGFIX step.

If you would like to see a short demo on using the z/OS V2.1 migration workflow, visit the IBM Media Center for an older, but yet still valuable, video on what the migration workflow looks like, and how to use it.

Upgrade Workflow Usage Tips

1. If you have performed part of any Workflow, and there has been an updated version of that Workflow that has been issued, you can update your partially completed Workflow by using the action "Create new based on existing". This can "merge" the two Workflows together such that unchanged steps that you have completed, stay completed.
2. The URL links to the documentation in the workflow cannot go to an anchor in the web page. The URLs will just bring you to the web page, not content that may be further down in the page. You may have to scroll down on the web page to find the information that you need.
3. For each upgrade action and for the entire upgrade, you can optionally provide your feedback to IBM. Just follow the instructions you see in z/OSMF. You do not need to provide feedback to complete each step of the workflow.
4. When searching for something in the Workflow, use the Search capability in the upper right corner. This strong capability can look "inside" all the steps to find the string you are looking for.

z/OS Upgrade Workflow
Starting in z/OS V2.4, IBM no longer provide the z/OS Migration publication, GA32-0889, in its current format. Since z/OS V2.2, the preferred method for learning about upgrade actions has been the z/OS Upgrade Workflow. Discovering, performing, and verifying many upgrade actions through the z/OSMF Workflow function instead of a more traditional book format allows for a tailored and specific upgrade path associated with a particular system.

Starting with the z/OS V2.4 release and later, IBM provides upgrade tasks in a z/OSMF Workflow, as well as a single exported file. By providing the z/OS upgrade materials in both formats, users still can enjoy the advantages of a z/OSMF Workflow as well as being able to search, browse, and print in a more traditional format. Notice that the

exported format of the z/OS upgrade materials that can be easily read or printed for those without any z/OSMF capabilities will not be tailored for any environment.

z/OS 3.1 Upgrade Workflows…new enhancements!

- z/OS 3.1 Upgrade Workflow is critical to learn about the necessary changes in a release.
- Using the z/OS 3.1 Upgrade Workflow via z/OSMF provides the following assistance:
  - Steps are reordered to be in time sequence (vs. sorted by element).
  - New step for looking at APARs/PTFs previously installed, to determine if steps can be skipped
    - Because you would have done the HOLD ACTION during the PTF installation already.
    - Many of the steps are of this type!
  - Step “Install coexistence and fallback PTFs”, runs an SMP/E REPORT MISSINGIX for the z/OS 3.1 coexistence PTFs.

29

z/OS Upgrade Workflows…existing enhancements

z/OS 3.1 Upgrade Workflow continues to build on the existing enhancements that prior release incorporated

- Discovery of used priced features, and other functions that skips steps right away, that you don’t need.
- Invocation of health checks to determine upgrade applicability.
  - An exception will mark the workflow step as failed, until re-run is successful, to ensure you ok.
- Ability to optionally provide feedback on your upgrade experience.

Remember, if you choose to use the “Exported” z/OS 3.1 Upgrade Workflow provided on IBM Documentation, none of the above z/OSMF assistance available.

30

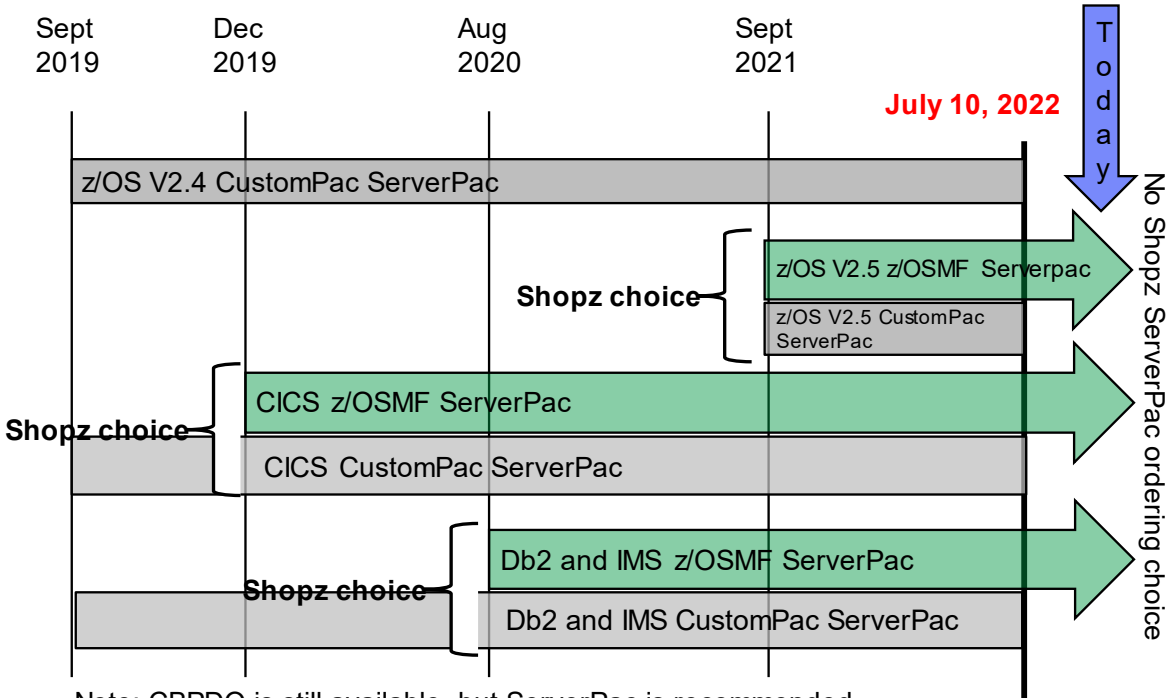
General z/OSMF Portable Software Instance (ServerPac) Reminder

- Since July 10, 2022, every IBM product available on Shopz orderable as a ServerPac is packaged as a portable software instance, and must be installed with z/OSMF.
  - This is for all IBM products: z/OS, MQ, CICS, Db2, and IMS, …
- **CBPDO remains available for those who do not wish to use z/OSMF.**

- *What this means to you right now:*
  - Learn your way around z/OSMF Software Management so you can successfully install a ServerPac packaged as a portable software instance.
    - Try out a sample package at https://www.ibm.com/support/z-content-solutions/serverpac-install-zosmf/
  - If you can't get z/OSMF active on your driving system, then use the COD, which activates z/OSMF.

31

ServerPac Portable Software Instance Timeline



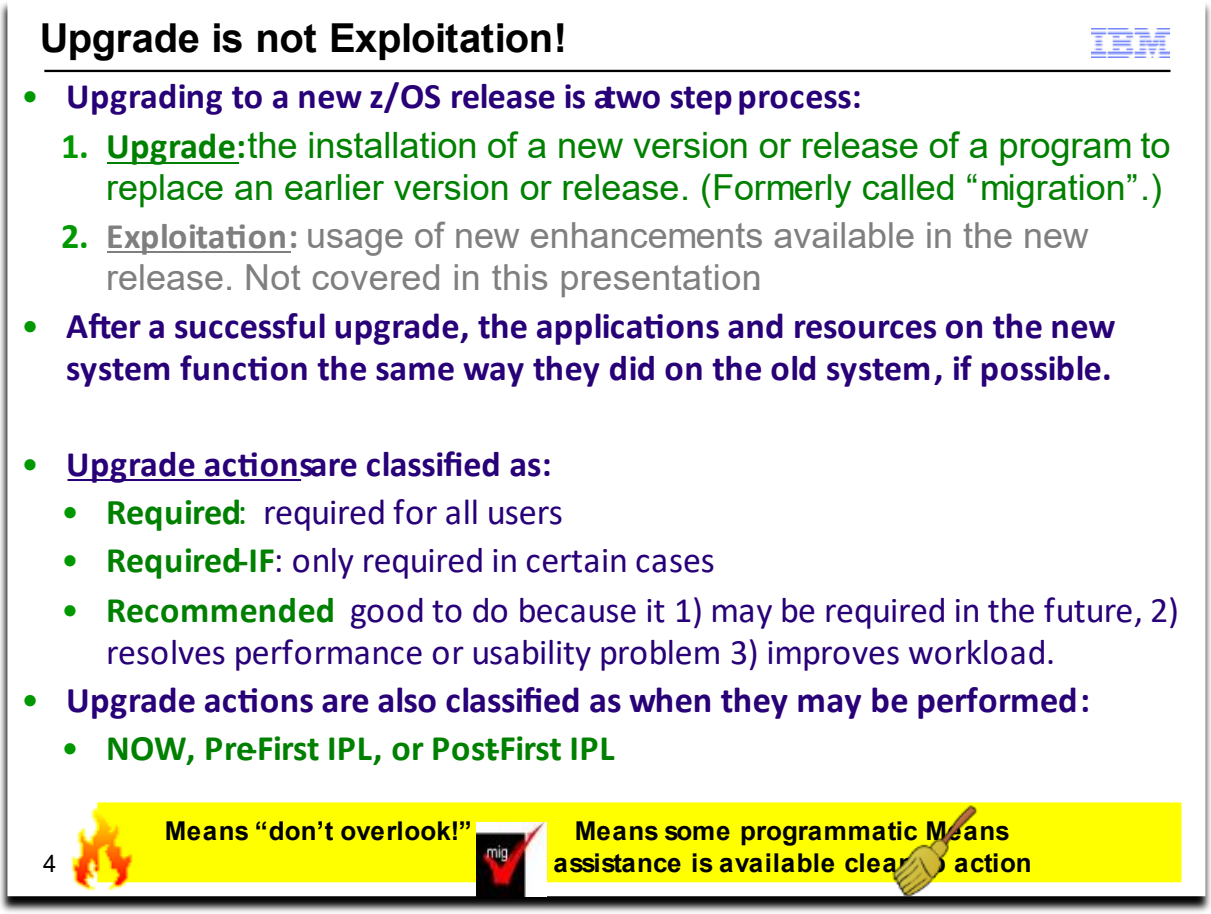
Note: CBPDO is still available, but ServerPac is recommended.
Withdrawal extension of CustomPac Dialog for ServerPac was done to accommodate addition of data set merge capability in z/OSMF.

32

Upgrading to z/OS 3.1: Technical Actions Topics

- **Definition of a "upgrade action"**
- **Overview of upgrade actions for z/OS 3.1 from z/OS V2.5 or V2.4:**
  - General Upgrade Actions
  - BCP
  - JES2
  - HCD
  - RMF
  - z/OSMF
  - zCX
  - SDSF
  - z/OS OpenSSH
  - Provided in handout
    - Security Server – RACF
    - Communications Server

34



Upgrade Definitions and Classifications

Upgrade (formerly, migration) is the first of two stages in upgrading to a new release of z/OS. The two stages are:

- **Stage 1: Upgrade.** During this stage you install your new system with the objective of making it functionally compatible with the previous system. After a successful upgrade, the applications and resources on the new system function the same way (or similar to the way) they did on the old system or, if that is not possible, in a way that accommodates the new system differences so that existing workloads can continue to run. Upgrade does not include exploitation of new functions except for new functions that are now required.
- **Stage 2: Exploitation.** During this stage you do whatever customizing and programming are necessary to take advantage of (exploit) the enhancements available in the new release. Exploitation follows upgrade.

Upgrade Requirement Classification and Timing

The upgrade actions are classified as to their requirement status:

- ***Required.*** The upgrade action is required in all cases.
- ***Required-IF.*** The upgrade action is required only in a certain case. Most of the actions in this presentation are in this category.
- ***Recommended.*** The upgrade action is not required but is recommended because it is a good programming practice, because it will be required in the future, or because it resolves unacceptable system behavior (such as poor usability or poor performance) even though resolution might require a change in behavior.

To identify the timing of upgrade actions, this presentation uses three types of headings:

- ***Now.*** These are upgrade actions that you perform on your current system, either because they require the current system or because they are possible on the current system. You don't need the z/OS 3.1 level of code to make these changes, and the changes don't require the z/OS 3.1 level of code to run once they are made. Examples are installing coexistence and fallback PTFs on your current system, discontinuing use of hardware or software that will no longer be supported, and starting to use existing functions that were optional on prior releases but required in z/OS 3.1.

- ***Pre-First IPL.*** These are upgrade actions that you perform after you've installed z/OS 3.1 but before the first time you IPL. These actions require the z/OS 3.1 level of code to be installed but don't require it to be active. That is, you need the z/OS 3.1 programs, utilities, and samples in order to perform the upgrade actions, but the z/OS 3.1 system does not have to be IPLed in order for the programs to run. Examples are running sysplex utilities and updating the RACF data base templates.

  It is possible to perform some of the upgrade actions in this category even earlier. If you prepare a system on which you will install z/OS 3.1 by making a clone of your old system, you can perform upgrade actions that involve customization data on this newly prepared system before installing z/OS 3.1 on it. Examples of such upgrade actions are updating configuration files and updating automation scripts.

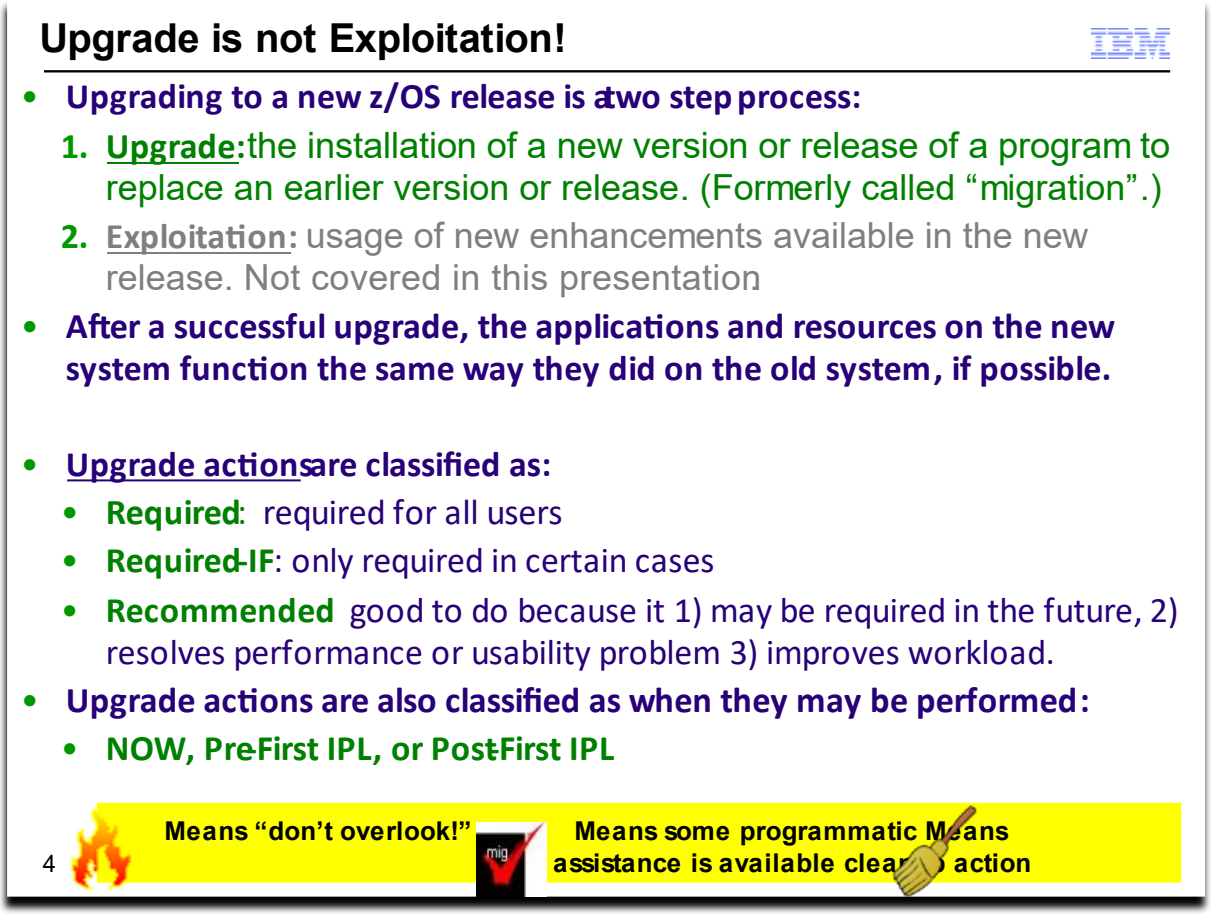
- ***Post-First IPL.*** These are upgrade actions that you can perform only after you've IPLed z/OS 3.1. You need a running z/OS 3.1 system to perform these actions. An example is issuing RACF commands related to new functions. Note that the term "first IPL" does not mean that you have to perform these actions after the very first IPL, but rather that you need z/OS 3.1 to be active to perform the task. You might perform the task quite a while after the first IPL.

Icons used in this presentation:

 means that you shouldn't overlook this upgrade action.

means that an IBM Health Check (using the IBM Health Checker for z/OS function) can help you with this upgrade action.

 means that this is a cleanup item or contains a portion that is a cleanup item. It is associated with something that is obsolete. It may cause confusion if someone thinks it does something. It is best to perform this upgrade action to avoid any confusion, since it is not needed anymore.



To use the latest version of the *z/OS 3.1 Upgrade Workflow* install PTFs marked with the FIXCAT IBM.Coexistence.z/OS.3.1. Once the PTF is installed, you will find the workflows in your `/usr/lpp/bcp/upgrade` path.

IBM strongly recommends you use the z/OS Upgrade Workflow from z/OSMF in order to have a customized upgrade of your applicable steps to take. However, if you wish to use the exported version of any z/OS Upgrade Workflow, you can see it at the bottom of the page on the IBM Documentation for z/OS 3.1 Web page, under "z/OS System Level", and then "z/OS Upgrade Workflow".

Elements with Upgrade Actions for z/OS 3.1

These elements have V2.5→ 3.1 upgrade actions :

- ➢BCP
- ➢BDT
- ➢Communications Server
- • Cryptographic Services – System SSL
- ➢DFSMS
- ➢HCD
- ▪ Infoprint Server
- ➢ JES2
- ➢ JES3
- ▪ Language Environment
- ➢ RMF
- ➢ SDSF
- ➢ Security Server (RACF)
- ▪ XL C/C++
- ➢ z/OS Container Extensions
- ➢ z/OS Management Facility

➢means that some of that element's upgrade actions are discussed in these presentations.

6

Upgrade Actions for Elements in z/OS 3.1

When upgrading from z/OS V2.5 to z/OS 3.1, the specified elements in the slide above have new or usual upgrade actions.

If you are upgrading from z/OS V2.4, use the *z/OS 3.1 Upgrade Workflow* for the z/OS V2.4 path to see the upgrade actions which were introduced in V2.5. Alternatively, if you wanted to see the upgrade actions, you can also see the exported workflow on the IBM Documentation website.

Some upgrade actions for selected elements follow in this presentation. This presentation does not cover all possible upgrade actions.

General Upgrade Actions for z/OS 3.1

Upgrade and **Exploitation**

- **Upgrade Actions Pre -First IPL:**
  - **Accommodate new address spaces (Recommended)**
    - **No new address spaces in z/OS 3.1 for upgrade or exploitation.**
    - New in V2.5:
      - **IBM z/OS Change Tracker (CYGSTC):**
        - For helping system programmers manage their z/OS configuration data sets.

7

General Upgrade Actions for z/OS 3.1

Upgrade Actions Pre-First IPL:

- **Remove references to deletedsyslib data sets and paths (Required)**
  - Removed in 3.1: **IBM JES3 and BDT** data sets, **zDNN** path /usr/lpp/IBM/ aie/IBM, and **IBM KC4Z** data sets,
    - and **SDSF: ISF.SISFLINK**, and **ISF.SISFMIG** has parmlib updates to do.
  - Removed in V2.5: **OCSF, EIM,** and some **ISPF, NFS,** and **Infoprint Server** paths. *(See RMF restructure later .)*
- **Add references to new syslib data sets and paths (Required)**
  - New in 3.1: **XML Toolkit** target, dlibs, and path **, zDNN path** /usr/lpp/aie/IBM, **Data Gatherer** dlib and path.
  - New in V2R5: **Infoprint Server** path *(See RMF restructure later.),* and **Change Tracker for z/OS.**
- **Update your health check customization for modified checks (Recommended)**
  - New in V2.5: 9 checks 3.1: 3 checks
  - Changed in V2.5: 2 checks 3.1: 1 check
  - No checks deleted in V2.5 or 3.1

8

General Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

General Upgrade Actions You Can Do Now

Install coexistence and fallback PTFs (Required)
Upgrade action: Install coexistence and fallback PTFs on your systems to allow those systems to coexist with z/OS 3.1 systems during your upgrade, and allow back out from z/OS 3.1 if necessary. Use the SMP/E REPORT MISSINGFIX command in conjunction with the FIXCAT type of HOLDDATA as follows:

1. Acquire and RECEIVE the latest HOLDDATA onto your pre-z/OS 3.1 systems. Use your normal service acquisition portals (recommended) or download the HOLDDATA directly from http://service.software.ibm.com/holdata/390holddata.html. Ensure you select **Full** from the Download NOW column to receive the FIXCAT HOLDDATA, as the other files do not contain FIXCATs.
2. Run the SMP/E REPORT MISSINGFIX command on your pre-z/OS 3.1 systems and specify a Fix Category (FIXCAT) value of **"IBM.Coexistence.z/OS.3.1**". The report will identify any missing coexistence and fallback PTFs for that system. For complete information about the REPORT MISSINGFIX command, see *SMP/E Commands*.
3. Periodically, you might want to acquire the latest HOLDDATA and rerun the REPORT MISSINGFIX command to find out if there are any new coexistence and fallback PTFs.

Use SOFTCAP to identify the effect of capacity changes (Recommended)
Not required, but is recommended to help in assessing processor capacity and available resources when upgrading to new software levels, and when upgrading to z/Architecture.
Upgrade action:

- Download SoftCap from one of the following Web sites:
  - Customers: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS268
  - Business partners: http://partners.boulder.ibm.com/src/atsmastr.nsf/Web/Techdocs. Note that this requires an ID on PartnerWorld®.Run SoftCap to determine your expected increase in CPU utilization (if any) and to identify your storage requirements, such as how much storage is needed to IPL.

Reference information: *SoftCap User's Guide*, which is provided with the tool.

General Upgrade Actions Pre-First IPL

Migrate /etc /global, and /var system control files (Required)
Upgrade action: The /etc, /global, and /var directories contain system control files: the /etc directory contains customization data that you maintain and the /global and /var directory contains customization data that IBM maintains. During installation, subdirectories of /etc , /global, and /var are created. If you install z/OS using ServerPac, some files are loaded into /etc /global, and /var due to the customization performed in ServerPac. You have to merge the files in /etc /global, and /var with those on your previous system. If you install z/OS using CBPDO, you should copy the files from your old system to the z/OS 3.1 /etc and /var subdirectories.

Copy files from your old system to the z/OS 3.1 /etc and /var subdirectories, and then modify the files as necessary to reflect z/OS 3.1 requirements. If you have other files under your existing /var directory, then you will have to merge the old and new files under /var. The easiest way to do this is to create a copy of your current /var files and then copy the new /var files into the copy.

The following elements and features use /etc:

- BCP (Predictive Failure Analysis).
- CIM.
- Communications Server (IP Services component).
- Cryptographic Services (PKI Services and System SSL components).
- DFSMSrmm.
- IBM HTTP Server.
- IBM Tivoli Directory Server (TDS). The LDAP server component uses /etc/ldap.
- Infoprint Server.
- Integrated Security Services. The Network Authentication Service component uses /etc/skrb.
- z/OS UNIX.

The following elements and features use /global:

- IBM Knowledge Center for z/OS
- IBM z/OS Management Facility (z/OSMF).

The following elements and features use /var:

- DFSMSrmm.
- IBM Tivoli Directory Server (TDS). The LDAP server component uses /var/ldap.
- Infoprint Server.
- Integrated Security Services. The Network Authentication Service component uses /var/skrb.

Back virtual storage with real and auxiliary storage (Required)

Upgrade action: As you exploit additional virtual storage by defining additional address spaces or by exploiting memory objects, ensure that you have defined sufficient real and auxiliary storage. Review real storage concentration indicators via an RMF report to evaluate if additional real or auxiliary storage is needed:

- Check UIC and average available frames.
- Check demand page rates.
- Check the percentage of auxiliary slots in use.

Reference information: For more information about memory objects, see *z/OS MVS Programming: Extended Addressability Guide* and Washington Systems Center flash 10165 at http://www.ibm.com/support/techdocs. (Search for "flash10165".)

Remove references to deleted data sets and path (Required)

Upgrade action: Using the tables in *z/OS Upgrade Workflow* as a guide, remove references to data sets and paths that no longer exist. Remove the references from the following places:

- Parmlib
- Proclib
- Logon procedures
- Catalogs
- Security definitions, including program control definitions
- DFSMS ACS routines
- /etc/profile
- SMP/E DDDEF entry (if you installed with CBPDO)
- Backup and recovery procedures, as well as any references to them in the table, the high-level qualifiers in the data set names are the default qualifiers.

Note: Do not remove any data sets, paths, or references that are needed by earlier-level systems until those systems no longer need them, and you are sure you won't need them for fallback.

Reference information: *z/OS Upgrade Workflow* contains the list of all removed data sets and paths in z/OS 3.1 and V2.5.

Add references to new data sets (Required)

Upgrade action: For z/OS V2.5, RMF and z/OS Data Gatherer, had several data sets restructured. Follow the RMF upgrade action to make the necessary parmlib and SYSPROC changes.

For z/OS 3.1, zDNN changed the installation path from /usr/lpp/IBM/aie/IBM to /usr/lpp/aie/IBM.

Accommodate new address spaces (Recommended)

Not required, but recommended to keep interested personnel aware of changes in the system and to ensure that your MAXUSER value in parmlib member IEASYSxx is adequate.

The following element adds a new address space for z/OS V2.5, which is exploitation support, and is not an upgrade action:

- **IBM z/OS Change Tracker** This new priced feature, added post-GA of z/OS V2.5 adds one new address space, CYGSTC. IBM z/OS Change Tracker can help z/OS system programmers track and control their z/OS configuration data sets.

The MAXUSER value in parmlib member IEASYS*xx* specifies a value that the system uses to limit the number of jobs and started tasks that can run concurrently during a given IPL. You might want to increase your MAXUSER value to take new address spaces into account. (A modest overspecification of MAXUSER should not hurt system performance. The number of total address spaces is the sum of M/S, TS USERS, SYSAS, and INITS. If you change your MAXUSER value, you must re-IPL to make the change effective.)

Update your check customization for modified IBM Health Checker for z/OS checks (Recommend)
Not required, but recommended to ensure that your checks continue to work as you intend them to work.
Changes that IBM makes to the checks provided by IBM Health Checker for z/OS can affect any updates you might have made.

The following health checks are new in z/OS V2R5:

- VSM_CheckRegionLoss
- RACF_ADDRESS_SPACE
- RACF_ERASE_ON_SCRATCH
- RACF_PROTECTALL_FAIL
- RACF_PTKTDATA_CLASS
- RACF_SYSPLEX_COMMUNICATION
- IOS_ENDPOINT_SECURITY_LCUPATHS
- ZOSMIGV2R5_NEXT_CS_OSADLH
- ZOSMIGV2R5_NEXT_CS_LSA

The following health checks are changed in z/OS V2R5:

- RACF_SENSITIVE_RESOURCES
- XCF_TCLASS_CLASSLEN

The following health checks were added by IBM in z/OS 3.1:

- ICSF_STATUS
- ICSF_CLEAR_KEYS
- SUP_ASVT_ABOVE_16M

The following health checks were changed by IBM in z/OS 3.1:

- RACF_PASSWORD_CONTROLS (added password phrase interval)

No health checks were deleted by IBM in z/OS 3.1 or V2.5.

Upgrade action:

1. Look at the updated checks in *IBM Health Checker for z/OS: User's Guide*.
2. Review changes you made for those checks, in HZSPRM*xx* parmlib members, for example.
3. Make any further updates for the checks to ensure that they continue to work as intended.

BCP Upgrade Actions for z/OS 3.1

• **Upgrade Actions Before First -IPL:**

Ensure that the sysplex uses SSD -capable sysplex couple data sets (Required-IF, as of 3.1)

- A basic or parallel sysplex requires a couple data set formatted to support System Status Detection (SSD) protocol.
- Failure to use the required level of sysplex CDS could result in:
  - z/OS 3.1 cannot initialize a sysplex containing a downlevel sysplex CDS.
  - z/OS 3.1 cannot join a running sysplex that contains a downlevel sysplex CDS.

- Use the XCF_SYSSTATDEF_PARTITIONING health check or enter `D XCF,COUPLE,TYPE=SYSPLEX` and check that "`SYSTEM STATUS DETECTION PROTOCOL IS SUPPORTED`" for both primary and alternate sysplex CDS's.
  - If the sysplex CDS is not formatted with the SSD protocol, format two replacement SSD -capable sysplex CDS's, with the input to the utility `ITEM NAME(SSTATDET) NUMBER(1)`.
  - Introduce the primary and alternate CDS to the sysplex using the usual `SETXCF` commands (new alternate, switch, new alternate).

BCP Upgrade Actions for z/OS 3.1

• **Upgrade Actions Before First -IPL:**

Verify default change for system use of non -executable memory (Req-IF, as of 3.1)

- System uses non-executable storage for passing parameters to a program.
  - Prior releases obtained the parameter area from executable storage.
- Applies to parameters passed to a program through either:
  - `PARM` or `PARMDD` keyword, on the `EXEC` statement
  - Parameter list passed by the system, by default, when there is no `PARM` or `PARMDD` on the `EXEC` statement
- System impacts could be:
  - ABEND0C4 errors, parameter lists that are omitted when `PARM` or `PARMDD` are omitted from the `EXEC`, programs using the CHKPT macro might receive return code 08 reason code 117.
- Modify any affected program to remove the requirement for passed parameters to be executable.
  - Although not recommended, DIAGxx `CBATTR EXECUTABLE(JCLPARM)` could be used, even temporarily.
    - (The opposite is `CBATTR NONEXECUTABLE(JCLPARM)`)
  - This is a system-wide setting.

BCP Upgrade Actions for z/OS 3.1

- **Upgrade Actions Before First -IPL:**

Verify default change for SVC dump processing (Required-IF, as of 3.1)

- SDUMP optimization will now default to `OPTIMIZE=YES` for the CHNGDUMP command
  - Previously it was `OPTIMIZE=NO`.
- With `OPTIMIZE=YES` and sufficient CPU and free real memory, SDUMP will attempt to capture data using additional parallelism and advanced in-memory capture processing.
  - Below the resource availability level, `OPTIMIZE=NO` behavior is used.
  - You will see if optimization was used, via message IEA794I "...`DUMP CAPTURED USING OPTIMIZE=YES`"

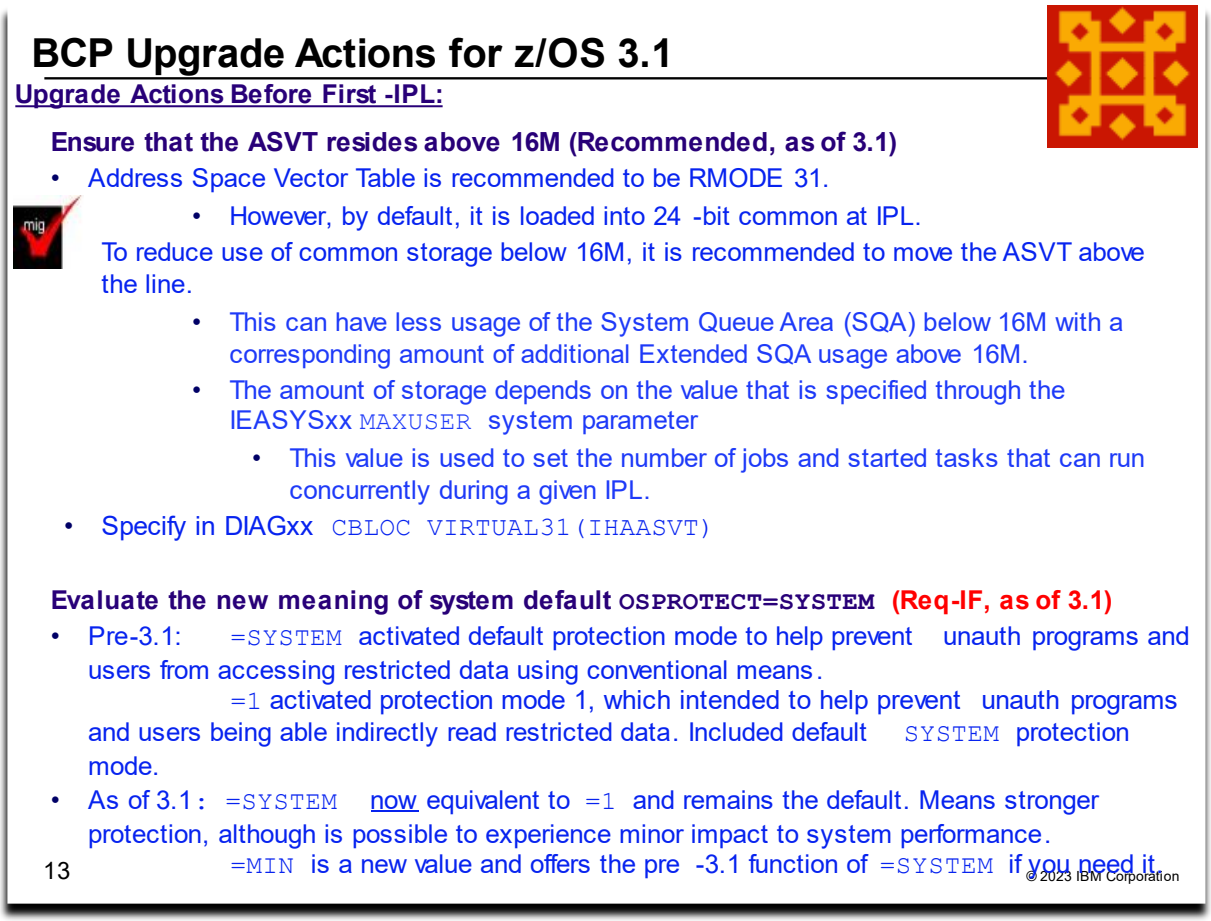
- When using the new default, clean up any specifications of `OPTIMIZE=YES`
  - On CHNGDUMP SET command, or through COMMNDxx
  - Take notice if COMMNDxx is shared pre-3.1.

BCP Upgrade Actions for z/OS 3.1

- **Upgrade Actions Before First -IPL:**

Verify default change for tape library requests (Required-IF, as of 3.1)

- ALLOCxx statement `SYSTEM TAPELIB_PREF` specifies the policy for balancing non-specific tape library requests across multiple tape libraries.
  - As of 3.1, default is `BYDEVICES`.
    - Allocations are randomized across all eligible devices.
    - Example over time with two tape libraries:
      - One library with 256 devices will have approx. 2/3 scratch allocations,
      - Other library has 128 devices will have approx. 1/3 scratch allocations
  - Pre-3.1, default was `EQUAL`.
    - Allocations to tape libraries are treated equally. Might have led to unbalancing across subsystems or with TS7700.
    - Same example:
      - One library with 256 devices will have approx. 1/2 scratch allocations,
      - Other library has 128 devices will have approx. 1/2 scratch allocations
- `SETALLOC SYSTEM,TAPELIB PREF=` can be used to change dynamically.



BCP Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

BCP Upgrade Actions Pre-First IPL

Ensure that the sysplex uses SSD-capable sysplex couple data sets (Required-IF, as of 3.1)

Required if you run z/OS in a sysplex. If so, you must upgrade to an SSD-capable sysplex CDS before introducing the first z/OS 3.1 system into the sysplex.
Starting in zOS 3.1, a basic or parallel sysplex requires a sysplex couple data set (CDS) that is formatted to support the System Status Detection (SSD) protocol. Over time, new function will depend on information that is stored in this CDS level and not available in earlier levels.

With this change, observe the following considerations:

- A zOS 3.1 system cannot initialize a sysplex by using a COUPLExx parmlib member that specifies or resolves to a sysplex CDS configuration that contains a down-level sysplex CDS.
- A zOS 3.1 system cannot join a running sysplex that is using a sysplex CDS configuration that contains a down-level sysplex CDS.

In the first item above, the phrase "resolves to" refers to system processing that attempts to determine the sysplex CDS configuration that was most recently used by a running sysplex. If the COUPLExx member specifies a CDS configuration that was previously used by a running sysplex, but was not the last configuration used by that sysplex, the system attempts to analyze the sequence of configurations used to find the most recent configuration.
This resolution process continues even if the interim configurations include down-level sysplex couple data sets. The requirement for SSD-capable couple data sets does not apply until the final, last-used CDS configuration is identified.

Upgrade action:

Determine whether you are using the required level of sysplex CDS. To do so, enter the MVS command: `D XCF,COUPLE,TYPE=SYSPLEX` and check the message "SYSTEM STATUS DETECTION PROTOCOL IS SUPPORTED" for both the primary and alternate sysplex couple data sets. If this line is absent, the CDS is down-level and must be upgraded to an SSD-capable sysplex couple data set.

To upgrade to SSD-capable sysplex couple data sets, or if a zOS 3.1 system encounters the IXC255I / IXC207A failure, proceed as follows:

1. Format two replacement SSD-capable sysplex couple data sets: One for primary and one for alternate. To format an SSD-capable sysplex CDS, include the following line in your input to the IXCL1DSU format utility: `ITEM NAME(SSTATDET) NUMBER(1)`
2. Update or create a COUPLExx parmlib member with the replacement CDS names.
3. If the zOS 3.1 system will join an existing sysplex, or if you are proactively upgrading an existing sysplex, introduce the replacement primary and alternate couple data sets into the sysplex, as follows:
    a. Specify one of the newly-formatted couple data sets as the alternate CDS by using the `SETXCF COUPLE,ACOUPLE` command
    b. Switch the new alternate CDS to the primary CDS by using the `SETXCF COUPLE,PSWITCH` command
    c. Specify another newly-formatted couple data set as the alternate by using the `SETXCF COUPLE,ACOUPLE` command.
4. On IPLing the zOS 3.1 system (if applicable), respond to message IXC207A with the updated or new COUPLExx member.

Verify the default change for system use of non-executable memory (Required-IF, as of 3.1)

Required if you need executable storage for parameters or parameter lists.
Starting in z/OS 3.1, the system uses non-executable storage for passing parameters to a program. In previous releases, the system obtained the parameter area from executable storage.
This change applies to parameters that are passed to a program through either of the following methods:

- Parameters passed through the PARM keyword or the PARMDD keyword on the EXEC statement.
- Parameter list that is passed by the system, by default, when PARM and PARMDD are omitted from the EXEC statement.

Though not recommended, it is possible to override this change by specifying the following option in the DIAGxx parmlib member: CBATTR EXECUTABLE(JCLPARM).

System impacts:

- Programs that are invoked from JCL and require the passed parameter area to be executable might receive ABEND0C4 errors.
- Parameter lists that are passed by the system, by default, when PARM and PARMDD are omitted from the EXEC statement.
- Programs that use the CHKPT macro might receive return code 08, reason code 117, from the CHKPT macro.

Upgrade action:
If you have programs that are invoked from JCL and require the passed parameter area to be executable, modify the programs to remove this requirement.
If you have programs that use the CHKPT macro, ensure that the programs can tolerate return code 08, reason code 117. Either change the programs to tolerate this return and reason code or change the programs to remove the use of the CHKPT macro.
The following options in parmlib member DIAGxx can be used to enable or disable the function:

- **CBATTR EXECUTABLE(JCLPARM)**. This option causes the system to obtain the parameter list storage using the EXECUTABLE=YES option.
- **CBATTR NONEXECUTABLE(JCLPARM).** This option causes the system to obtain the parameter list storage using the EXECUTABLE=NO option.

The DIAGxx options enable or disable the behavior on a system-wide basis. It is not possible to make program-specific exceptions for these settings.

Though not recommended, you might consider using the CBATTR EXECUTABLE(JCLPARM) specification in DIAGxx as a temporary workaround until the affected programs can be updated.

Verify the default change for SVC dump processing (Required-IF, as of 3.1)

Required if you were using the previous default and you do not want SDUMP data capture to use parallelism.

In z/OS 3.1, the default mode for SVC dump processing is changed to OPTIMIZE=YES. In previous releases, the default mode was OPTIMIZE=NO.

Prior to z/OS 3.1, if SDUMP optimization was required, it was necessary to request it explicitly by using the CHNGDUMP command. In z/OS 3.1, the default mode is changed to OPTIMIZE=YES, to improve SDUMP capture times.

When OPTIMIZE=YES is in effect, and sufficient CPUs and free real memory are available, SDUMP processing attempts to capture data using additional parallelism and advanced in-memory capture processing. Below certain levels of resource availability, or if OPTIMIZE=NO is specified, SDUMP captures the dump without using this additional parallelism.

Note: During SDUMP processing, if OPTIMIZE=YES is in effect, the technique chosen for dump capture is identified in message IEA794I. This text is not seen with OPTIMIZE=NO.

```
IEA794I SVC DUMP HAS CAPTURED: 948
DUMPID=003 REQUESTED BY JOB (TESTDUMP)
DUMP TITLE=TESTDUMP WITH SUMLIST64
DUMP CAPTURED USING OPTIMIZE=YES
```

Steps to take:

To check your current setting for SDUMP optimization, enter the command `DISPLAY D,O` at the operations console and examine the output from message IEE857I.

If the new default OPTIMIZE=YES is acceptable, consider whether the following clean-up actions are needed:

- In previous releases, if you were specifying OPTIMIZE=YES explicitly through the CHNGDUMP SET command or through the COMMNDxx parmlib member, you can safely remove these invocations.
- If your COMMNDxx parmlib member is shared with systems running prior z/OS releases, evaluate whether the differing defaults are acceptable in your environment.

If you require the previous default behavior (OPTIMIZE=NO), you must now explicitly request it through the CHNGDUMP SET,SDUMP, OPTIMIZE=NO command. You can enter this command at the operations console or include it in the active COMMNDxx parmlib member for your system.

Verify the default change for tape library requests (Required-IF, as of 3.1)

Required if you prefer to use the older algorithm EQUAL for balancing non-specific tape library requests across tape libraries.

In z/OS 3.1, In parmlib member ALLOCxx, the statement SYSTEM TAPELIB_PREF specifies the policy for balancing non-specific tape library requests, such as scratch tape requests, across multiple tape libraries. In z/OS 3.1, the default value for parameter SYSTEM TAPELIB_PREF in parmlib member ALLOCxx is changed to BYDEVICES. In previous releases, the default was EQUAL.

When SYSTEM TAPELIB_PREF(BYDEVICES) is in effect, non-specific (scratch) allocations are randomized across all eligible devices. For example, if two tape libraries are eligible for a scratch allocation and each library has 128 devices, over time, each library will receive approximately half of the scratch allocations. However, if one of the libraries has 128 devices and the other library has 256 devices, over time, the library that has 128 devices will receive approximately 1/3 of the scratch allocations and the library that has 256 devices will receive approximately 2/3 of the scratch allocations.

In contrast, the previous default, SYSTEM TAPELIB_PREF(EQUAL), indicates that for nonspecific tape library requests, all tape libraries must be treated equally, and receive an equal share of the requests. For example, if two libraries are eligible for a scratch allocation and each library has 128 devices, over time, each library will receive approximately half of the scratch allocations. Likewise, if one of the libraries has 128 devices and the other library has 256 devices, over time, each of the libraries will still receive approximately half of the scratch allocations, regardless of the number of devices in the library.

With the previous default, SYSTEM TAPELIB_PREF(EQUAL), there might be times when device randomization within the selected library (or composite library) appears unbalanced either across subsystems or across clusters in a TS7700 Virtualization Engine (multi-cluster grid configuration running in balanced mode). As the number of eligible

subsystems increases, the likelihood of this imbalance occurring also increases as the number of subsystems increases.
SYSTEM TAPELIB_PREF(BYDEVICES) is the IBM recommended setting because it allows MVS™ Device Allocation to better balance the requests across tape libraries.
Steps to take:
Check the current SYSTEM TAPELIB_PREF setting in effect for your system. To do so, run the health check that is associated with this workflow step: IBMALLOC,ALLOC_TAPELIB_PREF. Alternatively, you can enter the DISPLAY ALLOC,OPTIONS command at the operations console and examine the output from message IEFA003I.

If you require the previous default behavior SYSTEM TAPELIB_PREF(EQUAL), you must now specify this setting in the ALLOCxx parmlib member. Or you can enable it dynamically (after an IPL) through the following operator command: SETALLOC SYSTEM,TAPELIB_PREF=BYDEVICES

Verify the default change for ALLOCxx UNIT UNITAFF (Required-IF, as of 3.1)
Required if you do not have any tape devices defined for z/OS, and you do not want the default UNIT UNITAFF value to be SYSALLDA.
In parmlib member ALLOCxx, the UNIT parameter specifies the installation default for the device on which the system is to place data sets. The subparameter UNIT UNITAFF specifies the installation default for the unit name on which the system is to place data sets when the following conditions are true:

- The data set for the referencing DD, that is, the DD that specifies UNIT=AFF, DISP=NEW or DISP=MOD (MOD treated as NEW) is not SMS-managed.
- The data set for the referenced DD, that is, the DD statement pointed to by the UNIT=AFF subparameter, is SMS-managed.
- The allocation is not part of a data set collection involving data set stacking.
- The system cannot obtain a unit name from the primary DD statement in the unit affinity chain.

If the UNIT UNITAFF subparameter is not specified, the default unit name is the tape generic that is highest in the device preference table. In z/OS 3.1, if no tape generic is highest in the device preference table because no tape devices are defined to the system, the default value is SYSALLDA, which contains all DASD defined to the system.
Steps to take:
If you do not specify the UNIT UNITAFF value in the ALLOCxx parmlib member, and you do not have any tape devices defined to the system, be aware that the default value is now SYSALLDA, which contains all DASD defined to the system. If you prefer to use a different value, you must specify that value in the ALLOCxx parmlib member.
To check the current UNIT UNITAFF setting in effect for your system, enter the DISPLAY ALLOC,OPTIONS command at the operations console and examine the output from message IEFA003I.
If you prefer to use a value other than SYSALLDA for UNIT UNITAFF, specify this setting in the ALLOCxx parmlib member.

Evaluate the new meaning of system default OSPROTECT=SYSTEM (Required-IF, as of 3.1)
Required if you need the old default behavior of OSPROTECT=SYSTEM.
The z/OS operating system provides controls that are intended to help to prevent unauthorized programs and users from being able to access restricted data using conventional means. Malicious or compromised unauthorized programs and users might use a security exploit to attempt to circumvent these controls and access restricted data.
Introduced with APAR OA54807 for z/OS V2R3, the OSPROTECT system parameter specifies the operating system protection mode for unauthorized programs and users. In previous releases, either of two settings were possible for OSPROTECT:

- **OSPROTECT=SYSTEM**. Activates the default protection mode, which is intended to help prevent unauthorized programs and users from accessing restricted data using conventional means. OSPROTECT=SYSTEM is the default.

- **OSPROTECT=1**. Activates protection mode 1, which is intended to help prevent unauthorized programs and users from being able to indirectly read restricted data. This mode also includes the default protection mode level of protection (see SYSTEM). Though OSPROTECT=SYSTEM is the default, IBM recommends that you activate the stronger protection mode by using OSPROTECT=1.

In z/OS 3.1, the meaning of OSPROTECT=SYSTEM is changed. This setting, which remains the default, is now equivalent to specifying OSPROTECT=1. With the new meaning of OSPROTECT=SYSTEM, stronger protection is in effect by default. However, the system might experience a minor impact to system performance, workload execution, or both.
In z/OS 3.1, a new value is added: OSPROTECT=MIN. If you need to obtain the pre-z/OS 3.1 functionality of OSPROTECT=SYSTEM, you can specify OSPROTECT=MIN in your active IEASYSxx member.

If you currently specify OSPROTECT=1, you do not need to change it.
Note: If you use an IBM System z® Personal Development Tool (zPDT®) emulation system, OSPROTECT=MIN and OSPROTECT=SYSTEM are supported, with OSPROTECT=SYSTEM meaning OSPROTECT=MIN. That is, on a zPDT system, the OSPROTECT=SYSTEM setting remains equivalent to its original meaning.
Steps to take:
Determine the current setting of OSPROTECT for your system. To do so, use the **DISPLAY IPLINFO,OSPROTECT** command to display the value specified or defaulted for OSPROTECT. The output is message IEE255I. For example, assume that OSPROTECT=1 is specified in IEASYSxx. Here, in response to the command D IPLINFO,OSPROTECT, the system returns the following output:

```
IEE255I SYSTEM PARAMETER 'OSPROTECT': 1
```

If your current setting is OSPROTECT=1, you have no action to take.
If your current setting is OSPROTECT=SYSTEM, evaluate whether the new default behavior for OSPROTECT=SYSTEM is acceptable. If so, you have no action to take.
Otherwise, if you require the lesser protection of the pre-z/OS 3.1 OSPROTECT=SYSTEM behavior, you must specify OSPROTECT=MIN in your active IEASYSxx member.

In APAR OA55954 for z/OS V2R3 and later, the ECVT_OSProtect field was added to the extended communication vector table (ECVT) at offset x'2C3'. This field contains a value that corresponds to the OSPROTECT system parameter specification.
Starting in z/OS 3.1, If you have programs that check the **ECVT_OSProtect** field in the ECVT, you can no longer assume that any non-zero value indicates "more than minimum" protection (protection mode 1).
As of z/OS 3.1:

- ECVT_OSProtect = x'FF' means that OSPROTECT=MIN is in effect.
- ECVT_OSProtect = '0' means that OSPROTECT=SYSTEM is in effect, with its new meaning, which is equivalent to OSPROTECT=1.
- ECVT_OSProtect = Any other value means that more than minimum protection (OSPROTECT=1) is in effect.

Note: z/OS 3.1 adds the field **ECVT_OSProtect_WhenSystem** to the ECVT. This field indicates the level of protection in effect when OSPROTECT=SYSTEM is specified or defaulted to:

- ECVT_OSProtect_WhenSystem = x'FF' means OSPROTECT=MIN, for a zPDT system.
- ECVT_OSProtect_WhenSystem = '1' means OSPROTECT=1, for a system running on an IBM zSystems processor.

Ensure that the ASVT resides above 16M (Recommended, as of 3.1)

Not required, but recommended for some virtual storage constraint relief.
The recommended residency mode for the Address Space Vector Table (ASVT) control block is RMODE 31. However, by default, the ASVT is loaded into 24-bit common storage at IPL. To reduce the use of common storage below 16M, it is recommended that you ensure that the ASVT resides above 16M.
System impact: Less usage of the System Queue Area (SQA) below 16M with a corresponding amount of additional Extended SQA usage above 16M. The amount of storage depends on the value that is specified through the MAXUSER system parameter, which is used to set the number of jobs and started tasks that can run concurrently during a given IPL.
Steps to take:
The ASVT can be placed above 16M by using the CBLOC VIRTUAL31(IHAASVT) parameter of the DIAGxx parmlib member.
Use health check IBMSUP,SUP_ASVT_ABOVE_16M determine whether the ASVT resides above 16M.

WLM CPU Critical option is automatically assigned to importance 1 work (Required-IF, as of 3.1)

Required if you do want the default behavior.
When you assign long-term CPU protection to critical work, you ensure that less important work will generally have a lower dispatch priority. Doing so is essential for CPU-sensitive work and protects your business-critical workloads. Currently, you must explicitly specify the CPU Critical option in the WLM policy, and the option can only be assigned to single-period service classes. Furthermore, with System Recovery Boost, WLM automatically assigns CPU Critical to any importance 1 and 2 single-period service class while the boost is in effect.

Starting with z/OS 3.1, WLM even goes further. Long-term CPU protection is also automatically assigned when no boost is in effect. That is, the CPU Critical option is assigned:

- Implicitly to any work of importance 1 (or importance 2 when a boost is in effect) for the first period of any service class (no matter if this is a single or multiperiod service class). This is also valid while boost is in effect.
- For any work in single-period service classes which have the CPU Critical option explicitly set in the WLM service definition (base definition or service class overrides).

If you want to modify implicit CPU Critical, or even disable it, use the new CCImp and CCImpBoost parameters in your IEAOPTxx member, as follows:

- **CCImp=0|1|2** specifies the importance level up to which CPU-protection is assigned when no System Recovery Boost is active. CCImp=1 is the default. CCImp=0 suppresses implicit CPU-protection.

- **CCImpBoost=0|1|2** specifies the importance level up to which CPU-protection is assigned during boost periods. CCImpBoost=2 is the default. CCImpBoost=0 suppresses implicit CPU-Critical when a boost is effect.

With this change in z/OS 3.1, your business-critical work (importance 1) is better protected from CPU constraints. However, if you do not want the CPU Critical option to be applied automatically, you can choose to disable this functionality.

Steps to take:

Implicit CPU Critical for importance 1 work can impact the CPU distribution to lower importance work. Ensure that the goals are appropriate given their importance level. Evaluate the current distribution of CPU at different importance levels, especially those that are covered by CPU Critical, to ensure that they have consistent CPU demands.

If you want to modify or even disable implicit CPU Critical, copy the contents of your active IEAOPTxx member to a new IEAOPTyy member for z/OS 3.1 and add the CCImp or CCImpBoost parameters according to your needs. Valid values for CCImp and CCImpBoost are 0, 1, or 2.

CCImp=1 and CCImpBoost=2 are the default values.

CCImp=0 and CCImpBoost=0 disable automatic CPU protection. The CPU Critical option explicitly set in your WLM policy stays in effect.

Accommodate the new DSLIMITNUM default (Required-IF, as of V2.5)

Required if the default is not acceptable on your system.

In the SMFLIMxx parmlib member, the parameter DSLIMITNUM is used to override the maximum number of data spaces and hiperspaces that can be created by a user-key program. In z/OS V2R5, the default for DSLIMITNUM is changed to 4096. In previous releases, the default was 4294967295.

Applications that invoke DSPSERV CREATE, especially those applications that loop erroneously, might fail with the more restrictive DSLIMITNUM default in effect.

Upgrade action: SMF 30 record field SMF30NumberOfDataSpacesHWM indicates the high-water mark of the number of data spaces that are owned by the unauthorized (problem state and user key) tasks that are associated with a job. This field is added when you apply APAR OA59137 and APAR OA59126.

If your installation runs jobs that rely on the default for SMFLIMxx DSLIMITNUM or IEFUSI word 7 subword 3, inspect the SMF 30 record field SMF30NumberOfDataSpacesHWM fields for values that exceed 4096. For jobs that exceed 4096, update SMFLIMxx or IEFUSI to allow the maximum number of data spaces that are required.

ASCB and WEB are backed in 64-bit real storage by default (Required-IF, as of V2.5)

Required if you have an application that relies on the ASCB or WEB to be backed in 31-bit real storage.

In z/OS 2.5, the address space control block (which is mapped by IHAASCB) and the work element block (which is mapped by IHAWEB) are backed in 64-bit real storage by default. Previously, these data structures were backed in 31-bit storage, unless your DIAGxx parmlib member specified CBLOC REAL31(IHAASCB,IHAWEB).

In z/OS 2.4, the keywords REAL31 and REAL64 were added to the CBLOC statement of parmlib member DIAGxx. With these keywords, you can specify which data structures are backed in 31-bit real storage or 64-bit real storage.

Upgrade action: Check for programs that issue the load real address (LRA) instruction in 31-bit addressing mode for the ASCB or WEB data structures. The LRA instruction cannot be used to obtain the real address of locations backed by real frames above 2 gigabytes in 24-bit or 31-bit addressing mode. For those situations, use the LRAG instruction instead of LRA. The TPROT instruction can be used to replace the LRA instruction when a program is using it to verify that the virtual address is translatable and the page backing it is in real storage. If you have programs that do not tolerate 64-bit real storage backing for the ASCB or WEB data structures, update the DIAGxx parmlib member to specify CBLOC REAL31(IHAASCB,IHAWEB).

Accommodate the new CHECKREGIONLOSS default (Recommended, as of V2.5)

Recommended. Enabling the CHECKREGIONLOSS option causes initiator address spaces to be recycled when the maximum obtainable region size is reduced below a threshold. This change is expected to have a positive impact on the system without requiring any intervention

In the DIAGxx parmlib member, the parameter VSM CHECKREGIONLOSS specifies the amount of region size loss that can be tolerated in an initiator address space. The initiator remembers the initial maximum available region size (below and above 16 MB) before it selects its first job. Whenever a job ends in the initiator, if the maximum available region size (below or above 16MB) is decreased from the initial value by more than the CHECKREGIONLOSS specification, the initiator ends with message IEF0931 or IEF094A, depending on whether the subsystem automatically restarts the initiator.When CHECKREGIONLOSS is enabled, your installation can avoid encountering 822 abends or 878 abends in subsequent jobs that are selected by the initiator. These abends can occur when the available region size decreases because of storage fragmentation or problems that prevent storage from being freed.

In z/OS V2R5, CHECKREGIONLOSS is enabled by default with a value of (256K,30M). This change means that when the 24-bit region size decreases by 256K or more, or when the 31-bit region size decreases by 30M or more, the initiator is ended and restarted, with message IEF0931 or IEF094A.

Upgrade action: Verify that the CHECKREGIONLOSS option is specified in your active DIAGxx parmlib member:

- If the CHECKREGIONLOSS option is not specified in your active DIAGxx parmlib member, determine whether you want the option to be disabled on your z/OS V2R5 system. If so, you can set the CHECKREGIONLOSS to a very high value such as (16M,2046M), which will effectively disable the option.
- If the CHECKREGIONLOSS option is specified in your active DIAGxx parmlib member, and you want to continue using your current setting, you have no action to take. Consider using the IBM default setting CHECKREGIONLOSS(256K,30M).

JES2 Upgrade Actions for z/OS 3.1

Upgrade Actions Before installation:

Evaluate the new JES2-supplied default job resource limits (Required-IF, as of 3.1)

- Pre-3.1, you can see job level resources used for any single job in the system. These include SPOOL space (track groups/TGs) and job output elements (JOEs), and others.
- As of z/OS 3.1, JES2 will enforce, by default, resource limits on a per job basis for TGs and JOEs. Thi offers you the benefit of resource limit processing.
  - If a job exceeds the default resource limit for either TGs or JOEs, the default action is to put the jo a `WAIT` state.
    - This requires operator intervention to either make additional resources available or cancel/purge the job.
    - Other actions that could be taken against a job are `NONE` and `FAIL`
  - If these resource limits are not managed by your installation, JES2 will enforce default limits and actions.
    - Privileged jobs are not subject to the new JES2 default resource limits.
  - Small environment (<10,000 TGs or <600 JOEs): default resource limit is 75% of available resou
  - Large environment: 25% of available resource.
  - Available resource = total available resource – privilege space for privilege jobs.
- If you do not like the default resource limits, set your own values with JES2 JOBCLASS initialization statements, using `LIMIT` and `ACTION`.
  - Can also be set using JES2 policies for individual jobs!
- Messages are produced when a job reaches 90% of the defined resource limit and again when the j has exceeded the limit (and an action was taken).
  - Consider making more resource available when you see these messages.
- Possible actions to take for a `WAIT`ing job: add another SPOOL volume, cancel or purge the job.
- To remove this capability for new jobs (removing this benefit!):
  - `$T JOBCLASS(*),RESOURCE(*)=(LIMIT=100,ACTION=NONE)`

45

JES2 Upgrade Actions for z/OS 3.1

Upgrade Actions Before installation:

Activate z22 mode (Required-IF, as of V2.5)

- z22 was introduced in z/OS V2.2. As of z/OS V2.5, you are not able to fall back to z11 mode.
- Activate z22 mode before IPLing z/OS V2.5.
  - `$D ACTIVATE` – verify activation to z22 mode.

```
$D ACTIVATE
$HASP895 $DACTIVATE 177
$HASP895 JES2 CHECKPOINT MODE IS CURRENTLY Z11
$HASP895 THE CURRENT CHECKPOINT:
$HASP895   -- CONTAINS 1350 BERTS AND BERT UTILIZATION IS 12
$HASP895      PERCENT.
$HASP895   -- CONTAINS 234 4K RECORDS.
$HASP895 z22 CHECKPOINT MODE ACTIVATION WILL:
$HASP895   -- EXPAND CHECKPOINT SIZE TO 304 4K RECORDS.
$HASP895 z22 ACTIVATION WILL SUCCEED IF ISSUED FROM THIS MEMBER.
```

  - `$ACTIVATE,LEVEL=z22`

  - CYL_MANAGED support is required for a successful activation. You might see that z22 mode requires an extra nnn 4K records for CKPT1.

Accommodate change to truncation of blanks (Required-IF, as of V2.4 with APAR OA60605 and OA60528)

- JES2 OUTCLASS option BLNKTRNC=YES|NO is no longer used. JES2 will now pass full records th include trailing blanks, regardless of the BLNTRNC= setting. Printing functions maybe be affected!
  - IBM PSF, IBM Download for z/OS, and IBM AFP Download Plus are affected. .
- See handout for more details.

15

JES2 Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

JES2 Actions Before Installation

Activate z22 mode (Required-IF, as of V2,5)
Required if you are use the z11 level for checkpoint data sets.

Starting with z/OS V2R5, JES2 no longer supports the z11 level for checkpoint data sets. z22 mode was introduced in z/OS V2R2. Activate JES2 in z22 mode, if you have not done so. When you switch to z22 mode, the system upgrades the JES2 checkpoint. You are not able to fall back to z11 mode.

Upgrade action: Follow these steps:

- On all of the systems in your MAS, determine your z22 checkpoint activation readiness, as follows:
  - Enter the $D ACTIVATE command to verify that activation to z22 mode can succeed.
  - If you enter the $ACTIVATE,LEVEL=z22 command, activation of CYL_MANAGED support is required.
  - You might see that z22 mode requires an extra nnn 4K records for CKPT1.
- Enter the JES2 $ACTIVATE command to verify non-configuration changes that must be accommodated before you go to z22, and to activate z22 mode. See the considerations for this command in *z/OS JES2 Commands.*

By default, JES2 restarts in the same mode as the other members of the MAS (if any are active) or the mode of the last active JES2 member when it was shut down. On a cold start, JES2 starts in z22 mode, unless overridden by the OPTSDEF COLD_START_MODE or UNACT parameter.

Accommodate change to truncation of blanks (Required-IF, as of V2.4 with APAR OA60605 and OA60528)
Required if you have any affected printing product or application. Determine if you are performing blank truncation in accordance with what you require.

JES2 supports an option BLNKTRNC= YES|NO on the OUTCLASS statement to control whether or not blanks at the end of records are removed prior to being written to SPOOL. This reduces the amount of space needed to store the records. Most processes that read record (PSO, SAPI, Spool Browse) have those blanks restored. However, because of a limitation of the Functional Subsystem (FSS) interface the blanks could not be restored when the record was read (prior to z/OS 2.4).

Blanks (the hex 40 character) are interesting because if the SYSOUT data is not text data, the hex 40 value may be part of a structured field (for example when processing a PDF format SYSOUT data set) and not a blank at all. A missing hex 40 character could cause the SYSOUT data set to be processed incorrectly.

With the introduction of SPOOL data encryption and compression processing in z/OS V2.4, the method used to pass records to the FSS was changed such that the trailing blanks would be restored when records were passed to the FSS. This occurs regardless of the BLNKTRNC setting on the OUTCLASS statement and whether the SYSOUT data set is compressed or encrypted.

The FSS interface was updated to allow printing products to get records in a manner compatible with the traditional blank truncation processing. This implies it is the printing product that decides whether blanks should be truncated or not (regardless of the BLNKTRNC setting in JES2). The printing product default for blank truncation may or may not be compatible with your SYSOUT processing requirement.

With these changes, the JES2 BLNKTRNC setting on the OUTCLASS statement only affects the size of the records written to SPOOL and only when compression and encryption are not being used for a SYSOUT data set. The BLNKTRNC setting no longer has any effect on records read from SPOOL via the FSS interface (and has no effect reading records using other interfaces).

Upgrade action:
Review the your print products and applications to determine whether or not you have a dependency on truncation of blanks. Ensure that your print products or applications are configured properly to accommodate your requirements to truncate blank or not truncate blanks.

IBM Print Services Facility (PSF) , Download for z/OS, and AFP Download Plus are affected IBM products.

JES2 Actions Pre-First IPL

Evaluate the new JES2-supplied default job resource limits (Required-IF, as of 3.1)
Required if the default behavior is not desired.

Your installation can use job level resource limits to limit the amount of critical JES2 resource that can be used by any single job in the system. The JES2 resources that are protected by job level resource limits are SPOOL space (track groups/TGs) and job output elements (JOEs). If a job exceeds its assigned resource limit, the assigned action is taken against the job.

With the installation and initialization of z/OS JES2 3.1, JES2 itself will enforce default resource limits on a per job basis for the JES2 resources of SPOOL space and Job Output Elements. If a job exceeds the default resource limit for either of these resource types, JES2 invokes the default action against the job. The default action is to put the job in a WAIT state, requiring operator intervention to either make additional resources available on the system, or cancel/purge the job.

If these resource limits are not specifically managed by your installation, JES2 will enforce default system resource limits and actions. Be aware that these default resource limits are now in place and that some jobs that might have completed execution in the past might now assume WAIT states or fail during job submission.

Privileged jobs are not subject to the new JES2 default resource limits.

How resource limits are defined

Resource limits are defined by specifying a percentage of the total amount of resource available to the JES2 installation. Actions taken against the job can be defined as:

ACTION=NONE

JES2 issues a message but does not prevent the job from consuming more resources.

ACTION=WAIT

JES2 issues a message and suspends allocation of affected resource to the offending job until more resources become available to the job.

ACTION=FAIL

JES2 issues a message and fails the request for resource allocation. Depending on the state of the job and the type of resource request, this setting might cause the job to fail or the affected JES2 function to fail.

The resource limits and the related action can be configured at the job class level or can be managed at the individual job level using JES2 policies. If the resource limits are not specifically managed by the installation, JES2 will enforce default system resource limits and actions.

How the defaults are determined by JES2

The default job level resource limits applied by JES2 are dependent on the size of your installation's JES2 environment. An environment is deemed to be either "small" or "large," and those sizes are determined independently for each resource type, as follows:

"Small" is an environment with less than 10,000 track groups (TGs) in JES2 SPOOL or fewer than 600 job output elements (JOEs).

In a small environment, the default resource limit is 75% of the available resource. In a large environment, the default resource limit is 25% of the available resource.

"Available resource" indicates the resource available to be used by a job, which is the total available resource minus the privilege space set aside for privileged jobs. Privileged jobs are not subject to resource limits.

The default action taken against a job that exceeds its assigned resource limit is to place the job in a WAIT state. The job remains waiting until operator action is taken to increase the amount of available resource on the system, free some resources currently held by the job (such as purging SPIN output), or the job is cancelled or purged.

The default resource limits and actions take effect automatically after a JES2 member in the MAS is started and initializes using z/OS 3.1 code.

Setting your own job resource limits

If the default resource limits are not the desired outcomes for your installation, you can use JES2 JOBCLASS initialization statements or commands to set installation-specific values for resource limits and actions. To do so, use the new command keywords LIMIT and ACTION. For more information, see the description of the $T JOBCLASS command in z/OS JES2 Commands.

The DEFAULT value for the LIMIT keyword sets the limit back to the default value, as determined by the size of the JES2 environment, as described earlier:

75% for a "small" environment

25% for a "large" environment

Job resource limits can also be examined and set for individual jobs through a set of job attributes available for use in JES2 policies of types JobCreate, JobInput,PreConversion and JobConversion:

TGResLimit and TGResAction, which are used to control resource type TG

JOEResLimit and JOEResAction, which are used to control resource type JOE

Syntax and semantics of these attributes is similar to the respective keywords in job class commands, except that value “DEFAULT” is not supported for xxxResLimit attributes. Use 0 instead to reset these values to the system defaults.

Related message processing

Jobs that reach 90% of the defined resource limit for the job will encounter the following message, which warns about the approaching resource limit:

```
$HASP1807 Job XXX is approaching its RRR resource limit (RC=nn)
```

where "XXX" is the name of the affected job, and "RRR" is the type of resource limit being approached. If this job should not be subjected to resource limits, you can make more resource of that type available on the system (such as adding a SPOOL volume), or purge some of the resource used by the affected job (purge SPIN output, for example).

Jobs that have exceeded the resource limit will encounter the following message issued, which indicates the action that was taken against the job:

```
$HASP1806 Job XXX exceeded its RRR resource limit. ACTION=AAA (RC=nn)
```

where "XXX" is the name of the affected job, "RRR" is the type of resource limit exceeded, and "AAA" is the action taken against the job (NONE, WAIT, or FAIL). Consider making more resource of that type available on the system or purge some of that resource used by the affected job, in order for the job to come out of a WAIT and continue. In the case of the job being failed, resource adjustment or levels, or both, should be changed before re-submitting the job.

When a job exceeds its resource limit and the $HASP1806 message is issued, an action is taken against the job. The action depends on the state of the job.

Considerations for ACTION values

Observe the following considerations for ACTION values:

When ACTION=NONE, only the message is issued. The job is not prevented from consuming more of the resource.

When ACTION=FAIL, the request for resource will fail. This action might have different consequences, depending on the state of the job and the type of resource request. Some examples:

When job exceeds its TG or JOE limit, SYSOUT data set allocation fails. This situation might cause the job to fail, depending on how your application reacts to the failure of data set allocation.

If the TG limit is exceeded when job is actively writing to a SYSOUT data set, JES2 issues ABEND S722 RC=04.

For a job that has passed the execution phase, a specific JES2 action fails. For example, an output group is not created for the job. As a result, job and respective data sets will still be intact, but will not be accessible through the interfaces that involve output groups. Other interfaces, such as SPOOL browse, can be used to access the data.

When ACTION=WAIT, the result depends on the type of the request and the state of the job. Some examples:

If job is in execution phase (is actively executing), the job is suspended and JES2 waits for operator action

If job is beyond execution phase, the processing of the job is delayed until operator intervention

In general, the operator can do one of the following for a job that is waiting:

Purge some SPIN output owned by the job

Increase overall resource available to JES2, for example, add another SPOOL volume

Cancel or purge the job

In some stages of processing, JES2 cannot wait for operator intervention for technical reasons. In this case, ACTION=WAIT is promoted to ACTION=FAIL. For example, if resource limit is exceeded when job is on NJE job receiver, job will be rejected and will have to be resent after the resource configuration is addressed.

Other considerations for job resource limits

Note that job resource limit settings can be changed through policies only at early stages in JES2 job processing. After the job conversion phase, these settings can no longer be changed.

Because of the goal to prevent adverse effects on the performance of normal resource allocation code paths, resource limit detection might lag the consumption of resources. As a result, the enforcement of the limit is not precise. The job might be able to consume resources somewhat above the configured limit before JES2 code detects this situation and responds to it.

After a message about job resource limits is issued for a job, the message is not repeated unless resource consumption by the job changes or the JES2 pool of a relevant resource changes. However, code in different stages of job processing detects resource limits independently. As a result, it is possible to see HASP1807 and HASP1806 messages repeatedly for the same job as it progresses through different phases.
System impacts: Failure to adjust the job level resource limit configurations on the system job classes can result in jobs waiting or failing if the jobs exceed the default job level resource limits.

The following messages are issued:
$HASP1807 is issued when a job reaches 90% of the defined resource limit
$HASP1806 is issued when a job exceeds its defined resource limit, and an action is taken against the job.
Steps to take:
If the default resource limits are not the desired behavior for your installation, you can use JES2 JOBCLASS initialization statements or commands to set installation-specific values for resource limits and actions. Here, you can use the new keywords LIMIT and ACTION.
Or you can set job resource limits for individual jobs by using a set of job attributes in JES2 policies of types JobCreate, JobInput, PreConversion and JobConversion through the use of attributes TGResLimit and TGResAction for controlling the track group (TG) resource type, or JOEResLimit and JOEResAction for controlling the job output element (JOE) resource type.
It is also possible to disable the JES2 default job level resource limits by using the following JES2 operator command:
$T JOBCLASS(*),RESOURCE(*)=(LIMIT=100,ACTION=NONE)
This command allows unlimited resource allocation for all new jobs entering JES2. Note, however, that disabling this support means that your system will not benefit from JES2 default resource limit processing.

HCD Upgrade Actions for z/OS 3.1

Upgrade Actions Before Installing:

Remove configurations for unsupported processor types (Req-IF, as of 3.1)

- Out of service processor types are not supported by HCD :
  - 2817 and 2818: z196 and z114
  - 2097 and 2098: z10 EC and z10 BC
  - 2094 and 2096: z9 EC and z9 BC
  - 2084 and 2086: z990 and z890
- Remove these server configurations from your IODF, before upgrading to 3.1.
- HCD cannot validate the I/O configuration for unsupported processor types.
- If you are still using a processor that is out of service, the system that maintains that IODF cannot be upgraded to 3.1.

16

HCD Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

HCD Upgrade Actions You Can Do Now

Remove configurations for unsupported processor types (Required-IF, as of 3.1)

Required if you unsupported processors are still defined in your IODF.

In z/OS 3.1, HCD removes support for the following processors types because they are out of service:

- IBM z114, processor type 2818 models M05 and M10
- IBM z196, processor type 2817 models M15, M32, M49, M66, and M80

Previously, in z/OS V2R5, HCD removed support for following processors types:

- IBM z10 EC, processor type 2097, models E12, E26, E40, E56, and E64
- IBM z10 BC, processor type 2098, model E10
- IBM z9 EC, processor type 2094, models S08, S18, S28, S38 and S54
- IBM z9 BC, processor type 2096, models R07 and S07
- IBM z990, processor type 2084, models A08, B16, C24, and D32
- IBM z890, processor type 2086, model A04

You cannot build a new production IODF or modify a work IODF if an unsupported processor type is defined in the IODF. This restriction applies to the z/OS system used to maintain the IODF.

Steps to take: Check your currently active IODFs to determine whether you have any saved processor configurations for these out-of-service processors. Foillow these steps:

1. Start HCD.
2. Select option 1 “Define, modify, or view configuration data”
3. Select option 3 “Processors” to see the list of defined processor configurations.

4. Check the Type column for any of the out-of-service processor types.
5. If you still have any processor configuration for one or more of the out-of-service processor types, determine whether the processor is still in use. If not, delete the configuration.

Otherwise, if the processor is still in use, the system that maintains the IODF cannot be upgraded to z/OS 3.1.

RMF Upgrade Actions for z/OS 3.1

Upgrade Actions Pre-First IPL:

- **Perform updates for RMF structural changes (Req, as of V2.5)**

z/OS V2.3 and V2.4	z/OS V2.5	IFAPRDxx FEATURENAME
Priced feature: RMF	Priced feature: RMF	RMF
	Priced feature: Advanced Data Gatherer (ADG) (which is entitled when ordering RMF)	ADV DATA GATHER
	Base element Data Gatherer (DG)	n/a

V2.3 and V2.4 RMF	V2.5 DG or ADG (1)	V2.5 RMF (2)	PARMLIB / PROCLIB
SERBLINK	SGRBLINK	SERBLNKE	LNKLST, APF
	SGRBLPA		LPALST
SERBCLS	SGRBCLS*	SERBCLS	SYSPROC

For **z/OS V2.5 RMF**: do the customization in a(1). All else(2) remains the same.
For **z/OS V2.5 DG** or **ADG**, do customization in (1)
* Sharing consideration with pre-V2.5.
Also: update CLASS(PROGRAM) profiles for new and removed data sets.

17

RMF Upgrade Actions for z/OS 3.1

Upgrade Actions Pre-First IPL:

Remove references to deprecated ports for the RMF DDS server (Recommended, as of V2.4)

As of V2.4, the following Distributed Data Server (DDS) options are deprecated: `DM_PORT, DM_ACCEPTHOST, MAXSESSIONS_INET, SESSION_PORT`, and `TIMEOUT`

- Remove these options from GPMSRV00.
- RMF is changed to no longer use ports 8801 and 8802.
  - 8801 was the default for the DDS option SESSION_PORT.
  - 8802 was the default for the DDS option DM_PORT.
  - These are recommended to be disabled for RMF DDS.
- A warning message is issued if SESSION_PORT or DM_PORT are present in GPMSRV00.

Remove the RMF Postprocessor XML Toolkit from your workstation (Req, as of 3.1)

- Due to new browser security standards, it is no longer acceptable to load JavaScript files from a local disk using a web browser.
  - This makes the RMF Postprocess XML Toolkit no longer usable.
- Remove this workstation code. On Windows is installed into program group “IBM RMF Performance Management”.
- Use as an alterative, the RMF Data Portal Postprocessor facility.

49

RMF Upgrade Actions For z/OS 3.1
This upgrade action was taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

RMF Upgrade Actions Pre-First IPL


Remove references to deprecated ports for the RMF DDS server (Recommended, as of V2.4)

Not required, but recommend if you use RMF and the DDS server is configured to use any of the deprecated options.

In z/OS V2R4 and later, the following Distributed Data Server (DDS) options are deprecated: DM_PORT, DM_ACCEPTHOST, MAXSESSIONS_INET, SESSION_PORT, and TIMEOUT. These options should be removed from PARMLIB member GPMSRV00

RMF is changed to no longer use ports 8801 and 8802. The ports are described, as follows:

In previous releases, RMF used port 8801 as the default for the Distributed Data Server (DDS) option SESSION_PORT. This port was used by the RMF PM java client in addition to the HTTP port.

In previous releases, RMF used port 8802 as the default for the DDS option DM_PORT. This port was used as the UDP/IP port for Tivoli® DM/390 communication. This product is no longer available.

Ports 8801 and 8802 are deprecated. It is recommended that you disable these ports in the RMF DDS. A warning message is issued if either of the options SESSION_PORT or DM_PORT are present in the RMF parmlib member GPMSRVxx.
Steps to take: If you have RMF, determine whether your system is affected by this change. Check for the following conditions:

- RMF PM version 2.4.x or an earlier release failing to connect to the DDS server.
- Warning message for deprecated DDS server options SESSION_PORT or DM_PORT.
- Health check messages GPMH1004I and GPMH1005E if health checks are enabled.

Do the following:

- Verify that you are running the latest level of RMF PM.
- If your RMF parmlib member GPMSRVxx includes the settings SESSION_PORT(8801) or DM_PORT(8802), remove the settings.
- Ensure that RMF client applications refer to the HTTP_PORT.

Remove the Postprocessor XML Toolkit from your workstation (Required, as of 3,1)
The RMF Postprocessor XML Toolkit is part of the RMF™ product. With the toolkit, you can display a downloaded RMF Postprocessor XML report in a web browser locally without network access.
Due to new browser security standards, it is no longer acceptable to load JavaScript files from a local disk using a web browser. As a result, the RMF Postprocessor XML Toolkit is no longer usable.
As an alternative, IBM recommends that you use the RMF Data Portal Postprocessor facility, which provides similar functions as the RMF Postprocessor XML Toolkit**.**
Steps to take: If you use the RMF Postprocessor XML Toolkit, uninstall it from your workstation. It is installed on Windows as an MSI package of XML, JavaScript, and HTML files. It is installed into program group IBM RMF Performance Management.
As an alternative, use the RMF Data Portal Postprocessor facility, which provides similar functions.

Determine updates for RMF structural changes (Required, as of V2.5)
When the PTFs for APARs OA58281 and OA58759 are applied to z/OS V2.3 or V2.4, the RMF product is restructured into the Data Gatherer and Reporter components. In z/OS V2.5, the Data Gatherer component is packaged and delivered as separate FMID, and a priced feature of Advanced Data Gatherer is added..

With the z/OS Data Gatherer now included in the z/OS base, the RMF installation procedure and licensing model are changed. RMF consists of two components that work together to provide performance management capabilities, as follows:

z/OS Data Gatherer
: Collects performance measurements from the hardware and operating system and provides access to these measurements across the sysplex.

RMF Reporter
: Uses the collected measurements to report performance statistics in tabular and graphical reports
The term RMF refers to the RMF Reporter component. When you are entitled to RMF, you are also entitled to the Advanced Data Gatherer priced feature.

In z/OS V2.5, the Data Gatherer base z/OS component (566527401) is shipped within the same FMID as the z/OS Advanced Data Gatherer priced feature, FMID HRG77D0. The RMF Reporter component (566527404) remains in the priced RMF feature and is packaged in the existing FMIDs HRM77D0 and JRM77DJ.

The new RMF feature provides the same capabilities as the RMF feature. The RMF feature entitles you to use both RMF Reporter and z/OS Advanced Data Gatherer.

Upgrade action:

- z/OS Data Gatherer product libraries SYS1.SGRBLINK must be added to the link list and APF list. SYS1.SGRBLPA must be added to the LPA list.
- RMF users must change SERBLINK to SERBLNKE in the link list.
- IBM supplied procedures RMF and RMFGAT are installed into SYS1.PROCLIB (as in previous releases), but are part of z/OS Data Gatherer.
- IBM supplied procedures RMFCSC, RMFM3B, GPMSERVE, GPM4CIM are installed into SYS1.PROCLIB and are owned by RMF, as in previous releases.
- IBM supplied CLISTs ERBS2V, ERBV2S, and REXX execs ERBSCAN, ERBSHOW, ERBVSDEF, are installed into SYS1.SGRBCLS during z/OS Data Gatherer installation. Make it available to your SYSPROC. If you use RMF, make SERBCLS available in your SYSPROC.
- Make the follow updates to your RACF program profiles:

```
RALT PROGRAM ERB* DELMEM('SYS1.SERBLINK' //NOPADCHK)
RALT PROGRAM GPM* DELMEM('SYS1.SERBLINK' //NOPADCHK)

RALT PROGRAM ERB* ADDMEM('SYS1.SERBLNKE' //NOPADCHK)RALT PROGRAM GPM*
ADDMEM('SYS1.SERBLNKE' //NOPADCHK)
RALT PROGRAM ERB* ADDMEM('SYS1.SGRBLINK' //NOPADCHK)
RALT PROGRAM GRB* ADDMEM('SYS1.SGRBLINK' //NOPADCHK)
```

z/OSMF Upgrade Actions for z/OS 3.1

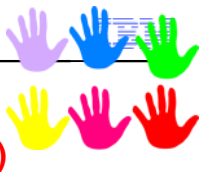
Upgrade Actions you *must* do NOW:

Check Workflow definition files for undeclared referenced entities (Required-IF, as of 3.1)

- Pre-3.1 (and using Java 8 64 -bit) the use of undeclared referenced entities was allowed if the z/OSMF workflow definition file also contained parameter entities.
- As of 3.1 (and using Semeru 11) the requirement for entity declarations is enforced.
  - Meaning, a workflow definition file that contains undeclared referenced entities **will fail validation** , regardless of whether it includes parameter entities.
- On 3.1, opening (or creating) one of this type of Workflow instances will result in an error (IZUWF0120E, entity not declared)
- **Prior to IPLing z/OS 3.1**, if you created any of your own Workflows, find and correct any such Workflow instances or definition files which have undeclared referenced entities.
  - Once you are using 3.1 z/OSMF, it will be too late to access impacted Workflow instances.
  - z/OS release fallback might be necessary if you must re -gain access to those Workflow instances.

19

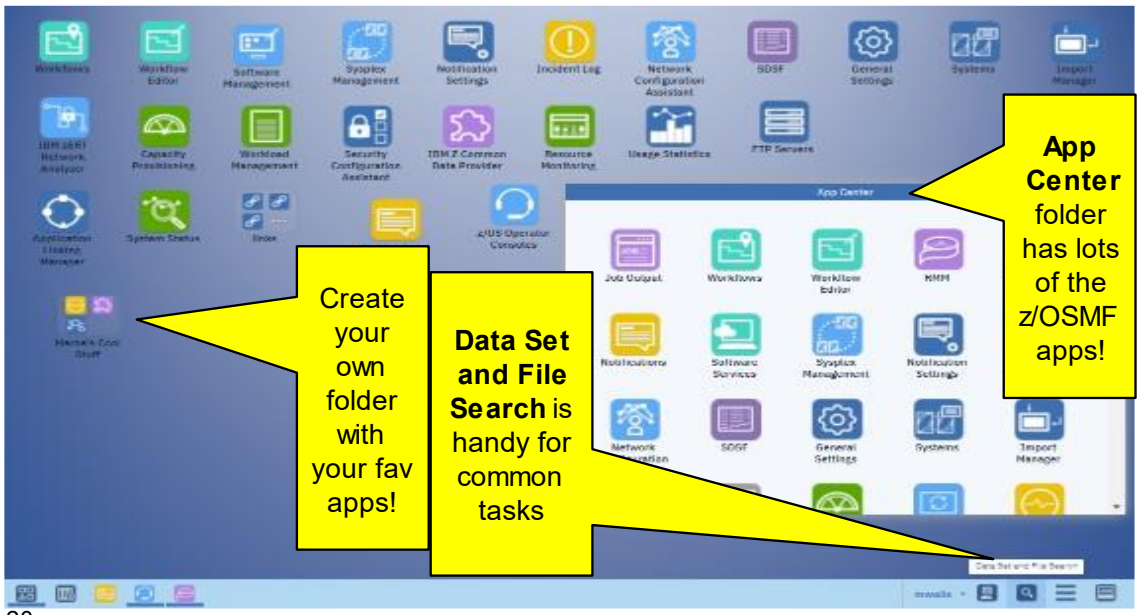
z/OSMF Upgrade Actions for z/OS 3.1



Upgrade Actions you can do NOW:

Use the z/OSMF Desktop interface (Required-IF, as of V2.5)

- More modern and personalized UI than the tree -style interface.



20

z/OSMF Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

z/OSMF Upgrade Actions You Can Do Now

Check workflow definition files for undeclared referenced entities (Req-IF, as of 3.1)

Required if your installation creates z/OSMF Workflows.
In z/OSMF Workflows, a workflow definition consists of one or more XML files and related files. The primary XML file is referred to as the workflow definition file.
Starting in z/OS 3.1, a workflow definition that contains both parameter entities and undeclared referenced entities will fail validation in the z/OSMF Workflows task and the z/OSMF Workflow Editor. In previous releases of z/OS, the use of undeclared referenced entities was allowed if the workflow definition file also contained parameter entities.
In z/OS 3.1, the requirement for entity declarations is enforced. A workflow definition file that contains undeclared referenced entities will fail validation, regardless of whether it includes parameter entities.
Attempting to open an existing workflow in z/OSMF from a workflow definition that includes undeclared referenced entities, or opening such a definition in the Workflow Editor, results in error message IZUWF0113E, which indicates the entity that is not declared.
System impact: Attempting to create a workflow instance on z/OS 3.1 in z/OSMF from a workflow definition that includes undeclared referenced entities, or opening such a definition in the Workflow Editor, results in error message IZUWF0120E, which indicates the entity that is not declared.
Steps to take: If your installation creates z/OSMF workflows, check the workflow definition files for undeclared referenced entities.
For any cases that you find, correct the workflow definition files. Either remove the undeclared referenced entities or define the entities before referencing them in the workflow definition file.
Delete any existing workflow instances that were created previously through this kind of workflow definition file, and create new instances, based on the corrected workflow definition file.
This must be completed before upgrading to z/OS 3.1, or you will encounter the system impact stated above.

Use the z/OSMF Desktop interface (Req-IF, as of V2.5)

Required because importing Policy Agent configuration files into Network Configuration Assistant is not supported in V2.5.
The z/OSMF desktop is the primary user interface for interacting with z/OSMF. In z/OS V2R5, the older classic or tree-style interface is removed from z/OSMF.

The z/OSMF desktop provides all of the functions of the classic interface in a more modern and personalized UI. The z/OSMF desktop includes task icons, a taskbar, and other desktop elements that can be customized by the user. With the z/OSMF desktop, users can interact with z/OS through a familiar interface that is similar to other operating environments. The z/OSMF desktop offers additional capabilities, such as:

- Search for z/OS data sets and files
- Ability to group tasks in a folder.

The z/OSMF desktop is displayed to users when they access the z/OSMF welcome page. If the classic interface was saved as a user preference in a previous release, the z/OSMF desktop is displayed instead.

Remove references to z/OSMF mobile notification service (Req-IF, as of V2.5)

Required if you use the z/OSMF mobile notification service.
z/OS V2R5 removes support for z/OSMF mobile notification service. The other z/OSMF notification services, including the Notifications task and the email notification services remain available, and can be used in place of the mobile notification service. Specifically, the following functions are removed from z/OSMF:

- In the z/OSMF graphical user interface (GUI), the following pages are removed from the Notification Settings task: Mobile Configuration page, and z/OSMF mobile application entry in the User page.
- REST API that was used for z/OSMF mobile notifications: POST /zosmf/notifications/new
- The following request body properties are removed from the z/OSMF notifications REST API:
  - "product"
  - “eventGroup”
  - “data”
  - “alert”

The change is related to the removal of the IBM zEvent mobile application and the Bluemix based push services.
Upgrade action: Determine whether any of your users or programs use the z/OSMF mobile notification service. If so, use the z/OSMF Notifications task or the z/OSMF email notification service as a replacement.

Stop using the policy data import function of Network Configuration Assistant (Req-IF, as of V2.5)
Required because importing Policy Agent configuration files into Network Configuration Assistant is not supported in V2.5.
z/OS V2.4 was the last release in which the Network Configuration Assistant (NCA) plug-in of z/OSMF supports the policy data import function. This function allowed the user to import existing Policy Agent configuration files into the Network Configuration Assistant.
In z/OS V2.5, it is not be possible to import policy configuration files for AT-TLS, IPSec, PBR, and IDS technologies. Import of TCP/IP profiles into Network Configuration Assistant is not affected.

Upgrade action: If you plan to import Policy Agent configuration files into the Network Configuration Assistant, perform this work prior on a pre-V2.5 release of z/OS (V2.3, or V2.4). Otherwise, you have no action to take.

z/OSMF Upgrade Actions Before First-IPL

Upgrade the IBM zERT Network Analyzer (Req-IF, as of 3.1)
Required if you use the IBM zERT Network Analyzer and you want to preserve your existing database and settings.
As part of upgrading to a new release of z/OS, you must upgrade your existing IBM zERT Network Analyzer settings to the new release. Starting with z/OS 3.1, the zERT Network Analyzer plugin includes the following types of assistance to help simplify upgrades from earlier supported releases:

- Database schema upgrade tooling for the database administrator
- Application and database settings upgrade functions for the user.

System impacts: If the upgrade action is not performed, the zERT Network Analyzer cannot connect to its database.
Steps to take:
Determine whether your installation uses the IBM zERT Network Analyzer plugin. To do so, check the IZUPRMxx member's PLUGINS statement for the ZERT_ANALYZER parameter. If specified, the IBM zERT Network Analyzer is enabled for the z/OSMF instance. Otherwise, the IBM zERT Network Analyzer is not used and no upgrade action is required.
If the IBM zERT Network Analyzer is enabled for use on your system, continue with the instructions in this section.
Upgrading the database schema:
Before starting the zERT Network Analyzer on z/OS 3.1, ask your Db2 for z/OS administrator to upgrade the network analyzer's database schema to the current 3.1 level. The 3.1 zERT Network Analyzer cannot connect to a down-level database.
Tip: You can check the database schema version and release values in the Database Information area of the IBM zERT Network Analyzer's Settings > Database Settings dialog.
The zERT Network Analyzer database schema tooling provides DDL templates to upgrade your existing V2R4 or V2R5 zERT Network Analyzer database to the current 3.1 schema, regardless of the schema version of your existing database.
The templates reside in the SYS1.SAMPLIB data set and are named as follows:

- **IZUZNADT1.** Upgrades a V2R5 database that uses fixed table names (as created by the IZUZNADT template) to the most current 3.1 schema.
- **IZUZNADT2.** Upgrades a V2R4 database that uses fixed table names (as created by the IZUZNADT template) to the most current 3.1 schema.
- **IZUZNADA1.** Upgrades a V2R5 database that uses aliased table names (as created by the IZUZNADA template) to the most current 3.1 schema.
- **IZUZNADA2.** Upgrades a V2R4 database that uses aliased table names (as created by the IZUZNADA template) to the most current 3.1 schema.

The 1 and 2 suffix in the member names represent the number of releases prior to the current release:

- IZUZNAx1 templates operate on databases that are one release behind the current release. For 3.1, the prior release is V2R5.
- IZUZNAx2 templates operate on databases that are two releases behind the current release. For 3.1, two releases prior is V2R4.

This naming scheme will be carried forward into future z/OS releases, as well.
To use the upgrade templates, do the following:

1. Compare the IZUZNADI variable substitution member of the SYS1.SAMPLIB data set to the customized variable substitution file that you used to create your zERT Network Analyzer database for any new variables. Add the new variables with suitable values to your customized file.
2. Invoke the IZUZNADG REXX exec in the z/OS 3.1 SYS1.SAMPLIB data set, specifying the template and your customized variable substitution file.

As an example, a user logged into z/OS as DBAUSER can invoke the IZUZNADG exec as shown below:

```
ex 'sys1.samplib(izuznadg)' 'zdbvars v24t120 ''sys1.samplib(izuznat2)'' dbvers(1.2.0)'
```

This invocation generates executable DDL commands for upgrading the V2R4 zERT Network Analyzer database (which uses fixed table names and schema version 1.2.0) to the latest 3.1 schema. The DDL commands are written to a data set named DBAUSER.V24T120, and the customized variable substitution values are defined in the data set DBAUSER.ZDBVARS.

Notes:

- In the command invocation, notice that two single quotes are used before and after sys1.samplib(izuznat2), rather than double quotes.

- For a description of IZUZNADG exec invocation syntax, run the 3.1 exec with the – HELP parameter.

Upgrading the Application and Database Connection Settings:

After the database schema is upgraded, you can log into the zERT Network Analyzer UI. The first time you log in, you are directed to the Database Settings panel to provide valid database connection information. The Application Settings are also set to default values.

You can use two new dialogs to import the zERT Network Analyzer database connection and application settings from an earlier supported release. Note that you must manually enter the database user ID and password or passphrase. These values are not imported from the previous release database settings.

In the zERT Network Analyzer user interface, new “Import or reset application settings” and “Import database connection settings” options are available on the Application Settings and Database Settings panels, respectively. Click these options to open a dialog to find and use the related settings from your z/OS V2R4 or V2R5 zERT Network Analyzer.

Observe the following considerations:

- If the 3.1 zERT Network Analyzer was installed in the same z/OSMF root directory as the previous release, clicking this button will open a new dialog that shows the relevant settings from the most recent supported release (V2R4 or V2R5). To use those settings, click **Import settings from a previous release**. You can also choose to cancel the dialog to retain the current settings (if you have changed them manually), and there is also an option to reset application settings to their default values if so desired.
- If no V2R4 or V2R5 settings are found within the current z/OSMF root directory structure, the dialog provides a file explorer widget through which you can find and select the V2R4 or V2R5 settings in another directory tree.

After valid database credentials are entered and saved through the Database Settings dialog, the network analyzer can connect to the upgraded database using any upgraded application or database settings.

zCX Upgrade Actions for z/OS 3.1

Upgrade Actions you can do NOW:

Prepare existing zCX workflow instances for z/OS 3.1 (Required-IF, as of V2.4 OA64231)

- zCX Workflows was impacted by the z/OSMF Workflow upgrade action ("Check workflow definition files for undeclared referenced entities").
- Correction was provided in OA64231 for the affected workflows, with workflow version 1.1.4.
  - If your zCX Workflow instances are at least 1.1.4, you are not affected.
- If you have any zCX Workflow instances that are 1.1.3 or earlier, you are impacted. Prior to IPLing 3.1, you must:
  - Complete, export, or delete any zCX workflow instances on your system
  - Install the PTF for OA64231.
- After 3.1 has been IPLed, you can no longer access the zCX workflows that are levels prior to 1.1.3.

Replace Vim usage with Nano in the IBM zCX CLI container (Required-IF, as of V2.4 OA64259)

- Vim (VI) text editor is removed from the command line interface, and replaced with the Nano text editor
- Stop using Vim and use Nano text editor instead

21

zCX Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

zCX Upgrade Actions You Can Do Now

Prepare existing zCX workflow instances for z/OS 3.1 (Req-IF, as of V2.4 with OA64231)
Required if you use the zCX z/OSMF workflows to provision zCX instances using the z/OSMF workflows task or REST APIs.
Note: zCX for OpenShift product (5655-ZCX) is also impacted and requires APAR OA64241.
Starting in z/OS 3.1, a workflow definition that contains both parameter entities and undeclared referenced entities will fail validation in the z/OSMF Workflows task. In previous releases of z/OS, the use of undeclared referenced entities was allowed if the workflow definition file also contained parameter entities.
Specifically, the z/OSMF Workflows task fails the workflow instance creation with the following error message:
`IZUWF0120E: "The validation of workflow definition file failed. The entity "name_of_the_entity" was referenced, but not declared."`
With the installation of APAR OA64231, the zCX z/OSMF workflows are enhanced to define the referenced entities in the workflow XML files that are shipped with the product. This enhancement works with all supported levels of z/OS and the z/OSMF Workflows task.
However, if your system has zCX z/OSMF workflow instances at version 1.1.3 or earlier, it does not have the maintenance applied and the workflow instances are impacted.
System impacts: Failure to perform this upgrade action can result in failure to open an existing zCX z/OSMF workflow instance after upgrade to z/OS 3.1. After the upgrade to the z/OS 3.1, users cannot open the remaining or incomplete zCX z/OSMF workflow instances in z/OSMF on a z/OS 3.1 system.
Steps to take: Prior to upgrading to z/OS 3.1, complete, export, or delete the remaining zCX z/OSMF workflows instances on your system.
Then, apply APAR OA64231 across your enterprise where you have z/OSMF servers running.
After upgrading to z/OS 3.1, users will no longer be able to open the older zCX z/OSMF workflow instances in z/OSMF in z/OS 3.1.

Replace Vim usage with Nano in the IBM zCX CLI container (Req-IF, as of V2.4 with OA64259)
Required if you use the Vim editor in zCX.
The Vim (VI) text editor is removed from the command line interface (CLI) container in IBM z/OS Container Extensions (zCX). It is replaced with the Nano text editor.
Steps to take: Stop using the Vim text editor. Replace it with the nano text editor, which is shipped with IBM z/OS Container Extensions (zCX).

SDSF Upgrade Actions for z/OS 3.1

Upgrade Actions Before Installing:

Use dynamic statements for SDSF configuration, not assembler macros (Req-IF, as of 3.1)

- As of 3.1, assembler macros for defining ISFPARMS is not allowed.
  - Usage of the parmlib member ISFPRMxx is required.
- ISFPRMxx was introduced in 1995 and is easier to use, less error prone, and more dynamic than assembling ISFPARMS.
  - In addition, when ISFPRMxx is used, the SDSF server creates a log that lists all ISFPRMxx statements processed and the values that were used
- **If you are already using ISFPRMxx, no action is necessary**
- To convert from assembler format to ISFPRMxx, use conversion tool ISFACP.

Remove dependencies on the non-scrollable main panel (Req-IF, as of 3.1)

- As of 3.1, SDSF does not provide the old -style non-scrollable main panel.
- Review your ISFPRMxx to ensure that `Panel.Main.DisableTable` is not found or is `FALSE`.
- Update any automation or pgoramming you have that depends on the the old-style format.

22

SDSF Upgrade Actions for z/OS 3.1

Upgrade Actions Before Installing:

Use only SAF-based security to protect SDSF functions (Req-IF, as of V2.5)

- As of V2.5, only SAF-based security is used to protect SDSF product functions.
  - SDSF no longer supports the use of legacy internal security, which is provided by definitions in the ISFPARMS assembler source or ISFPRMxx PARMLIB statements.
- As of V2.5, SDSF uses only SAF security profiles in classes, such as SDSF, OPERCMDS and JESSPOOL, to control the display and command authority in the product.
  - Non-SAF security decisions provided by the “Display Auth” and “Command Auth” exit points in ISFUSER are no longer supported.
- **If you are already using SAF for SDSF product security, no action is necessary.**
- To convert to SAF-base security, refer to the SDSF documentation.
  - As an alternative to the ISFACR tool, use the ISFNTCNV migration tool, which is shipped in V2.4/V2.5 APAR PH49811.
    - ISFNTCNV processes the installation ISFPRMxx member and generates RACF commands.
  - V2.5 APAR PH48846 provides the ability for z/OSMF Security Configuration Assistant to help with verifying and fixing security configuration for SDSF!
  - Supplied in SISFJCL sample job via APAR PH27387
  - *z/OS SDSF Security Migration Guide*, SC27-4942-40

54

SDSF Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

SDSF Actions You Can Do Now

Use dynamic statement for SDSF configuration, not assembler macros (Required-IF, as 3.1)

Required if are still using assembler macros for defining ISFPARMS, and are not using the parmlib member ISFPRMxx
Before z/OS 3.1, SDSF had two different methods for product configuration:

- assembling module ISFPARMS using SDSF macros
- defining statements through SYS1.PARMLIB member ISFPRMxx

ISFPARMS defined with assembler macros supported a subset of the SDSF configuration options as compared to ISFPRMxx and was no longer being enhanced. For example, custom properties and field lists for recent panels cannot be specified through ISFPARMS. The assembler based ISFPARMS cannot be shared across systems and a separate copy is required for each release of SDSF, even in a mixed environment. SDSF failures will result when ISFPARMS is assembled with incorrect macro levels for the release being run.

The ISFPRMxx parmlib member (introduced in 1995) is easier to use, less error prone, and more dynamic than assembling ISFPARMS. In addition, when the ISFPRMxx parmlib member is used, the SDSF server creates a log that lists all ISFPRMxx statements processed and the values that were used.

Given that ISFPARMS defined with assembler macros does not support all SDSF configuration options and the cumbersome process for creating it, using ISFPARMs defined with assembler macros will no longer be supported after z/OS V2.5. As of z/OS 3.1, all SDSF configuration must be done through parmlib member ISFPRMxx.
Steps to take:
SDSF provides a conversion tool (ISFACP) that can convert your existing ISFPARMS to ISFPRMxx format. This tool can be run on any SDSF release so that you can convert to ISFPRMxx before support for ISFPARMS defined with assember macros is dropped.

- ISFACP is a simple to use ISPF dialog that you invoke from ISPF option 6. After completing the dialog panel by specifying the data set name and member to convert, a corresponding ISFPRMxx will be generated that you can immediately use.
- You must complete the conversion to ISFPRMxx prior to IPLing z/OS 3.1. z/OS V2.5 SDSF is the last release of SDSF to ship the conversion tool ISFACP.

With the removal of support for ISFPARMS defined with assembler macros, z/OS V2.5 SDSF is the last SDSF release to ship the assembler macros necessary to create it. These macros are installed in ISF.SISFMAC and include, but are not limited to, the following: ISFFLD, ISFFLDM, ISFGRP, ISFGRPM, ISFNTBL, ISFNTBLM, ISFPMAC, and ISFPMACM.

Remove dependencies on the non-scrollable main panel (Required-IF, as 3.1)
Required if your installation has any automation or other programming that depends on the SDSF non-scrollable main panel.
As of z/OS 3.1, SDSF no longer provides an old-style (non-scrollable) version of the SDSF main panel.
The SDSF scrollable main panel was introduced in z/OS V2.3 SDSF when the number of SDSF options would no longer fit on a single page. At that time, SDSF provided a compatibility mode to allow fall back to the older style (non-scrollable) panel if it was needed for automation or some other process. In z/OS 3.1, the non-scrollable panel is removed from SDSF.
Your installation should update any automation or other programs that might depend on the existence of a non-scrollable SDSF main panel.
Steps to take:
Verify that the old-style main panel is not being used:
Review your ISFPRMxx definition and ensure that Panel.Main.DisableTable is not present or is set to FALSE.
Check for the special ddname ISFMIGMN allocated to the session.
Update any scripts or automation you have that depend on the old-style format to run with the scrollable main menu.

Use only SAF-based security to protect SDSF functions (Required-IF, as of V2.5)

Required if SDSF on your system uses security definitions in the ISFPARMS assembler source or ISFPRMxx PARMLIB statements.
Starting in z/OS V2R5, only SAF-based security is used to protect SDSF product functions. SDSF no longer supports the use of legacy internal security, which is provided by definitions in the ISFPARMS assembler source or ISFPRMxx PARMLIB statements.

The system authorization facility (SAF) is an interface defined by z/OS that enables programs to use system authorization services to control access to resources, such as data sets and MVS commands. SAF either processes security authorization requests directly or works with RACF, or other security managers, to process them.
As of z/OS V2R5, SDSF uses only SAF security profiles in classes, such as SDSF, OPERCMDS and JESSPOOL, to control the display and command authority in the product.
Non-SAF security decisions provided by the “Display Auth” and “Command Auth” exit points in ISFUSER are no longer supported.

Upgrade action: If you are already using SAF for SDSF product security, no action is necessary. If you are using SDSF internal security or the ISFUSER exit to enforce local security decisions, you must upgrade to SAF security for SDSF before using z/OS V2R5. Converting to SAF security for SDSF can be performed on any currently supported release of z/OS. For information about how to convert to the SAF interface, see SDSF Security Migration Guide, SC27-4942.

Update path environment variables to refer to the 64-bit JVM (Required-IF, as of V2.5)

Required if your installation uses SDSF programs that rely on 31-bit Java.
In z/OS V2R5, SDSF removes support for running an SDSF/Java application using the 31-bit Java virtual machine (JVM). SDSF/Java applications can run unchanged using the 64-bit JVM.
SDSF supplies Java classes (referred to as SDSF/Java) that allow access to SDSF panels and functions through applications written in Java. Such applications are typically invoked under the z/OS UNIX System Services shell by running Java and specifying a main class that references the SDSF/Java classes.
In previous releases, an SDSF/Java application could run using either the 31-bit or 64-bit version of the Java JVM. Effective with SDSF V2R5, the JVM must be running in 64-bit mode.
Upgrade action:

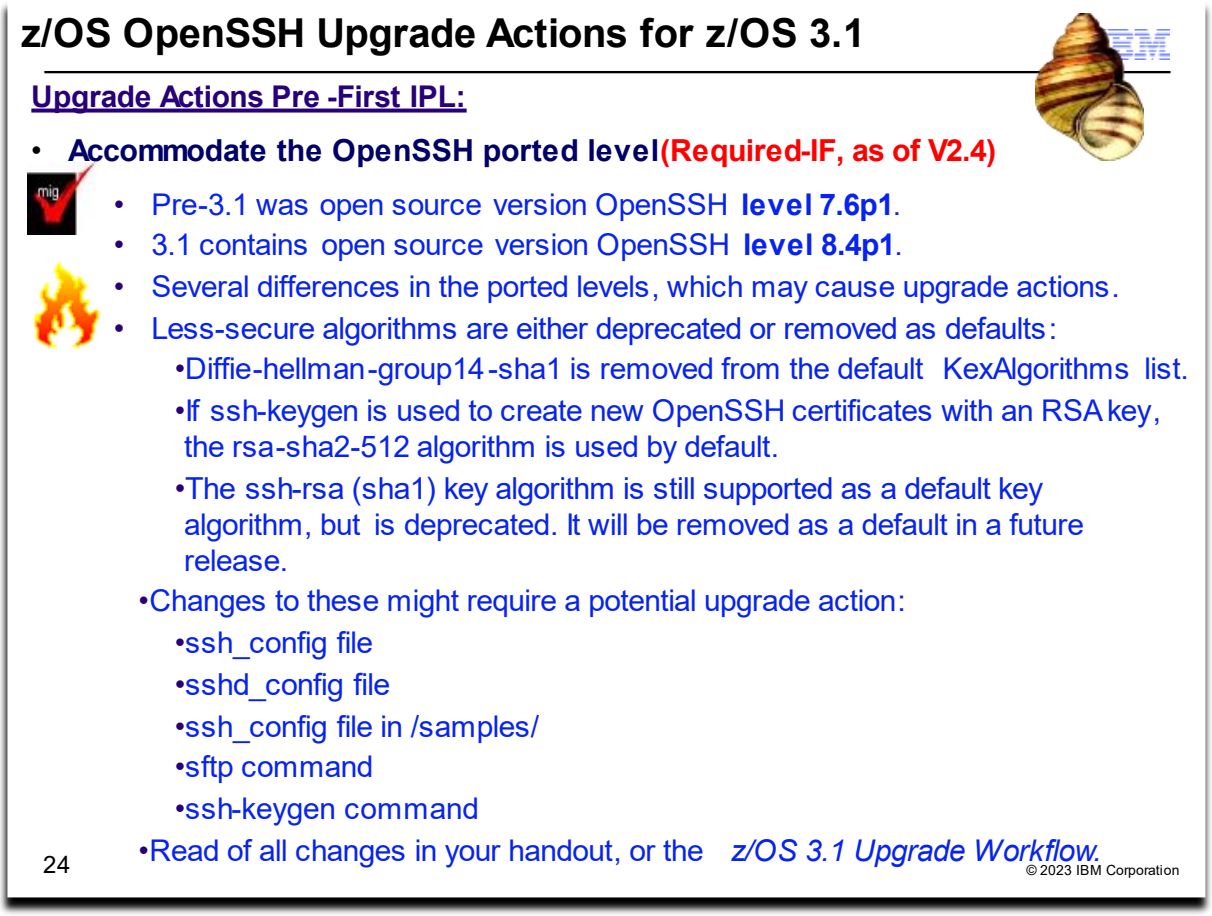
1. Check the PATH environment variable for a reference to the Java 31-bit JVM, such as the following:

export PATH=/usr/lpp/java/**J8.0**/bin:$PATH
If so, change the PATH environment variable to refer to the 64-bit JVM:
export PATH=/usr/lpp/java/**J8.0_64**/bin:$PATH

2. Check the LIBPATH environment variable for a reference to the 31-bit SDSF DLLs, such as the following:

export LIBPATH=/usr/lib/**java_runtime**:$LIBPATH
or
export LIBPATH=/usr/lpp/sdsf/java/**lib**:$LIBPATH
If so, change your LIBPATH environment variable to refer to the SDSF 64-bit DLL:
export LIBPATH=/usr/lib/**java_runtime64**:$LIBPATH
or
export LIBPATH=/usr/lpp/sdsf/java/**lib_64**:$LIBPATH

Note that no changes to your application code are necessary. The only difference is the use of the 64-bit JVM to run the application.



z/OS OpenSSH Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow*.

z/OS OpenSSH Upgrade Action Pre-First-IPL

Accommodate the OpenSSH ported level (Required-IF, as of 3.1)

Required if you reliant upon any of the changes in the newer ported level.
z/OS OpenSSH is updated to OpenSSH 8.4p1. Previously, the product was based on OpenSSH 7.6p1.
In z/OS OpenSSH V3.1, significant new features include the following:

- Support is added for FIDO/U2F key authentication, which is standardized support for user-present hardware tokens. z/OS OpenSSH supports these for verification only where the actual hardware token is not required, such as the following situations:
  - z/OS SSHD authentication of a remote user with a FIDO/U2F token.
  - z/OS ssh client verification of a host key, where the server has a FIDO/U2F token.

- Less-secure algorithms are either deprecated or removed as defaults:
  - Diffie-hellman-group14-sha1 is removed from the default KexAlgorithms list.
  - If ssh-keygen is used to create new OpenSSH certificates with an RSA key, the rsa-sha2-512 algorithm is used by default.
  - The ssh-rsa (sha1) key algorithm is still supported as a default key algorithm, but is deprecated. It will be removed as a default in a future release.

Steps to take:
The following sections describe potential upgrade actions for the z/OS OpenSSH base element:

- Changes to the ssh_config file that might require an upgrade action
- Changes to the sshd_config file that might require an upgrade action
- Changes to the sshd_config file in /samples/ that might require an upgrade action
- Changes to the sftp command that might require an upgrade action
- Changes to the ssh-keygen command that might require an upgrade action

Changes to the ssh_config file that might require an upgrade action:

What Changed	Customization action needed?
The key type default list is expanded for the following options: • HostbasedKeyTypes • HostKeyAlgorithms • PubkeyAcceptedKeyTypes For these options, the default list now includes the following key types: • sk-ecdsa-sha2-nistp256-cert-v01@openssh.com • sk-ssh-ed25519-cert-v01@openssh.com • rsa-sha2-512-cert-v01@openssh.com • rsa-sha2-256-cert-v01@openssh.com • sk-ecdsa-sha2-nistp256@openssh.com • sk-ssh-ed25519@openssh.com • rsa-sha2-512 • rsa-sha2-256	Yes. Because new key types are added to the default list, you must verify that the new key types are acceptable for your installation. Disable the key types that you do not plan to use.
KexAlgorithms option The following KEX (Key Exchange) algorithms are removed from the default list: • diffie-hellman-group-exchange-sha1 • diffie-hellman-group14-sha1	Yes. If you do not specify the option KexAlgorithms in the ssh_config file, you will use the default KexAlgorithms values. If you still need to use the previous default KEX algorithms, add the algorithms back to the default list. Ensure that the server sshd supports the new default key exchange algorithms list.

Changes to the sshd_config file that might require an upgrade action:

What Changed	Customization action needed?
The key type default list is expanded for the following options: • HostbasedAcceptedKeyTypes • HostKeyAlgorithms • PubkeyAcceptedKeyTypes For these options, the default list now includes the following key types: • sk-ecdsa-sha2-nistp256-cert-v01@openssh.com • sk-ssh-ed25519-cert-v01@openssh.com • rsa-sha2-512-cert-v01@openssh.com • rsa-sha2-256-cert-v01@openssh.com • sk-ecdsa-sha2-nistp256@openssh.com • sk-ssh-ed25519@openssh.com • rsa-sha2-512,rsa-sha2-256 To obtain the available key types, you can use the following commands: • "ssh -Q HostbasedAcceptedKeyTypes" • "ssh -Q HostKeyAlgorithms" • "ssh -Q PubkeyAcceptedKeyTypes"	Yes. Because new key types are added to the default list, you must verify that the new key types are acceptable for your installation. Disable the key types that you do not plan to use.

KexAlgorithms option The following algorithms are removed from the default list: • diffie-hellman-group-exchange-sha1 • diffie-hellman-group14-sha1	Yes. If you do not specify the KexAlgorithms option in your config file, you will use the default KexAlgorithms values. Determine whether you still need to use the previous default KEX algorithms. If so, add the algorithms back to the default list.

Changes to the sshd_config file in /samples/ that might require an upgrade action:

What Changed	Upgrade action needed?
PermitRootLogin option The setting PermitRootLogin yes is commented out, which means that prohibit-password is the default.	Yes, if your installation uses the sshd_config file in /samples/ as the default sshd_config. Be aware that the password logon for root user is prohibited by default in z/OS 3.1. If you still want to permit root user logon with password, set this option to yes.

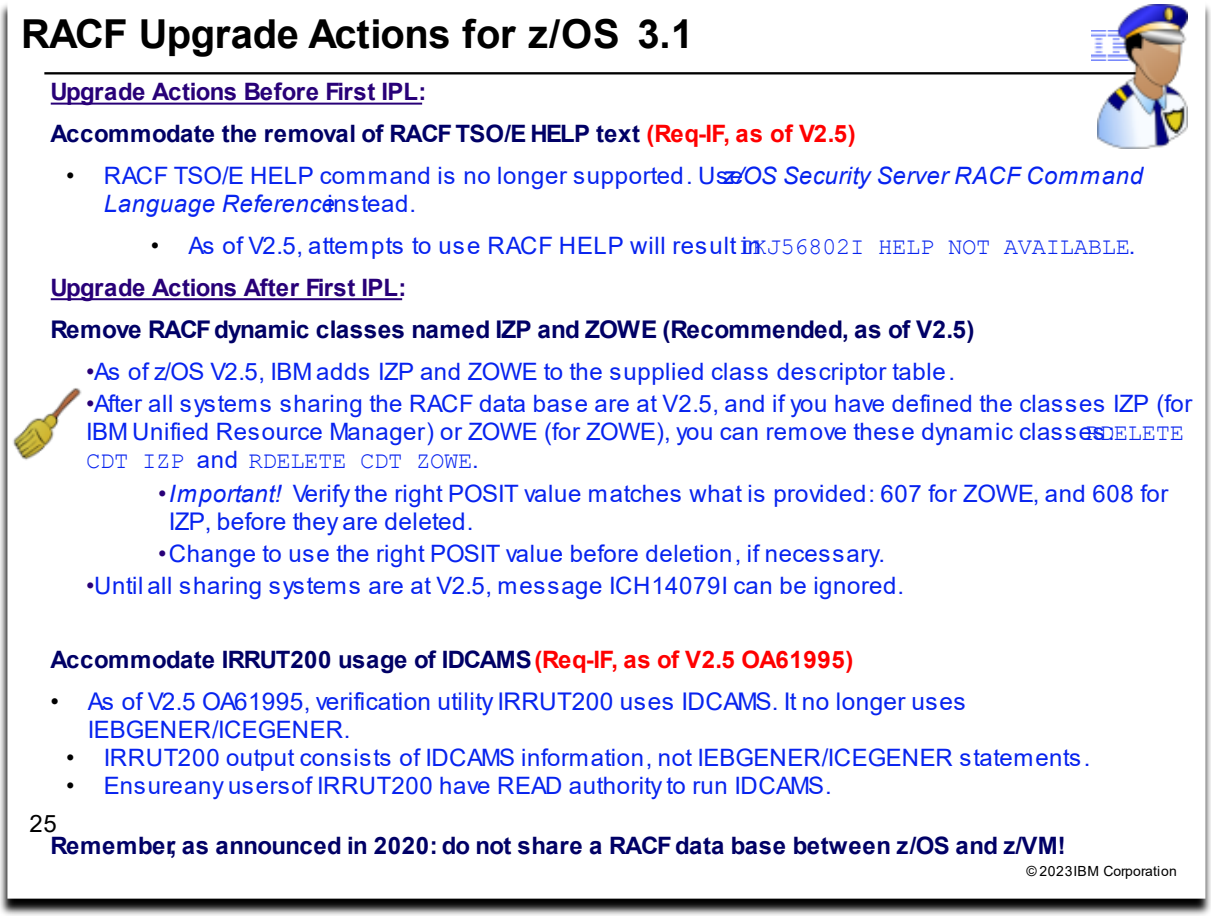
Changes to the sftp command that might require an upgrade action:

What Changed	Customization action needed?
The recommended sftp subcommand formats are changed in z/OS 3.1. Specifically, the following sftp subcommands: • get [-afPpr] remote-path [local-path] • put [-afPpr] local-path [remote-path] • reget [-Ppr] remote-path [local-path] • reput [-Ppr] [local-path] remote-path Are replaced by these sftp subcommands: • get [-afpR] remote-path [local-path] • put [-afpR] local-path [remote-path] • reget [-fpR] remote-path [local-path] • reput [-fpR] local-path [remote-path]	Yes, if your scripts use these sftp subcommands. If so, edit the scripts and replace the -r and -P subcommands with the -R and -p subcommands, respectively.

Changes to the ssh-keygen command that might require an upgrade action:

What Changed	Customization action needed?
The -b option. For RSA keys, the default bit length is changed from 2048 bits to 3072 bits.	Yes, if you are still using 2048 bits or less for RSA key generation. If so, consider changing your programs to use 3072 bits or higher. The maximum key size is 16384 bits.
For moduli file generation, the ssh-keygen options -G, -T, -J, -j, -K, -S and -W are replaced by the ssh-keygen -M option. Updated support for the ssh-keygen -M option: -M generate Generate candidate parameters for Diffie-Hellman Group Exchange (DH-GEX) for eventual use by the 'diffie-hellman-group-exchange-*' key exchange methods. The numbers generated by this operation must be further screened before use. See the MODULI GENERATION section for more information. -M screen Screen candidate parameters for Diffie-Hellman Group Exchange. This operation accepts a list	Yes, if your scripts use the ssh-keygen options -G, -T, -J, -j, -K, -S or -W. If so, update your scripts to use the -M option instead.

of candidate numbers and tests that the numbers are safe (Sophie Germain) primes with acceptable group generators. The results of this operation can be added to the /etc/moduli file. See the MODULI GENERATION section for more information.	
The options for moduli file generation are replaced by the -M option. As a result, the following options are no longer supported: • -G output_file option • -T output_file • -J num_lines • -j start_line • -K checkpt • -S start • -W generator	Yes, if your applications use the removed options for moduli file generation. If so, update your applications to use the -M option instead.



RACF Upgrade Actions For z/OS 3.1

These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

RACF Action Pre-First IPL

Accommodate IRRUT200 usage of IDCAMS (Required-IF, as of V2.5 OA61995)

Required if your installation uses a program profile for IDCAMS.

In z/OS V2R5, RACF introduced support to allow a VSAM linear data set (LDS) to be used as a RACF data set in specific configurations. As a part of this support, the RACF database verification utility program IRRUT200 was changed to use IDCAMS REPRO instead of IEBGENER or ICEGENER, if either the source or target data set is a VSAM LDS.

Now, with z/OS 3.1 (and z/OS V2R5 with APAR OA61995), IRRUT200 uses IDCAMS REPRO for copying any RACF data set, including a non-VSAM to non-VSAM copy operation. This results in a change to the output sent to the IRRUT200 SYSUT2 DD name. Previously, a non-VSAM to non-VSAM RACF data set copy operation would result in the following messages:

```
DATA SET UTILITY - GENERATE PROCESSING ENDED AT EOD IRR62065I - IEBGENER copied
SYSRACF to the work dataset SYSUT1, IEBGENER RC=0000
```

With z/OS 3.1 (and z/OS V2R5 with APAR OA61995), a RACF data set copy operation results in this message:

```
IRR62005I - IDCAMS REPRO copied SYSRACF to the work data set SYSUT1
```

In the unlikely event that your installation has program-protected the IDCAMS program, users of the IRRUT200 utility need to be granted READ authority to the profile that covers the IBM-supplied IDCAMS program.

Steps to take:

If SETROPTS WHEN(PROGRAM) is not in effect, no action is required.

If SETROPTS WHEN(PROGRAM) is in effect and there is no profile in the PROGRAM class which matches the program name "IDCAMS" and the volume on which it resides, no action is necessary.

If SETROPTS WHEN(PROGRAM) is in effect and a profile in the PROGRAM class matches the program name "IDCAMS" and the volume on which it resides, an upgrade action is required. Ensure that user IDs that use IRRUT200 to copy a RACF data set have READ authority to the PROGRAM class profile. Be sure to enter a SETROPTS WHEN(PROGRAM) REFRESH command after updating the PROGRAM class profile.

Examine your use of IRRUT200 to see if you have any dependencies on the messages produced by IEBGENER. If you have any dependencies on the IEBGENER or its replacement messages, review the new IDCAMS utility messages.

Accommodate the removal of RACF TSO/E HELP text (Required-IF, as of V2.5)
Required if you use the TSO/E HELP command for information about RACF command syntax.
As of z/OS V2R5, the RACF TSO/E HELP command is no longer supported. Entering the command "help *racf keyword*" in TSO/E results in the message "IKJ56802I HELP NOT AVAILABLE." For information about RACF command syntax, see the IBM publication z/OS Security Server RACF Command Language Reference.
Upgrade action: Stop using the TSO/E HELP command for information about RACF command syntax.

Do not share the RACF data base between z/OS and z/VM (Required-IF, as of V2.5, and enforced with OA62875)
Required if you share a RACF data base between z/OS and z/VM.

As had been previously announced, it is not possible to share RACF databases between z/VM and z/OS systems. With PTF for intended apar OA62875, z/OS RACF is now planning to check for and issue warning messages when a RACF database is shared with z/VM 7.3 (or higher). This is consistent with the behavior of RACF VM 7.3 which also intends to prevent RACF database sharing with z/OS.

Sharing of databases between z/OS systems is not affected by this statement.

.

RACF Action Post-First IPL

Remove RACF dynamic classes named IZP and ZOWE (Recommended, as of V2.5)
Not required, but recommended to avoid messages that indicate a conflict between the IBM classes and the dynamic classes.
If your installation uses IBM Zowe, its documentation directed you to add the ZOWE class to RACF. In z/OS V2R5, this class can be deleted after all of the systems that share the RACF database are upgraded to z/OS V2.5.

If your installation uses IBM Unified Resource Manager, its documentation directed you to add the IZP class to RACF. In z/OS V2R5, this class can be deleted after all of the systems that share the RACF database are upgraded to z/OS V2.5. Despite issuing message ICH14079I, RACF functions normally using the IBM-defined class.

In z/OS V2.5, RACF adds the IZP and ZOWE classes in the supplied class descriptor table.
Warning: If the RACF database is shared with any z/OS release earlier than V2R5, the deletion of the class will make the profiles in that class unusable on the downlevel system

Otherwise, after you upgrade to z/OS V2R5, RACF identifies the conflict between the dynamic class and the IBM class by issuing message ICH14079I during IPL and with every subsequent SETROPTS RACLIST(xxx) REFRESH for the class.

For a list of the new classes that are shipped with RACF, see the topic “Supplied resource classes for systems” in the IBM publication *Security Server RACF Security Administrator's Guide*.
Upgrade action: Check for message ICH14079I or the existence of the IZP and ZOWE classes in the RACF class descriptor table (CDT). For example:

- RLIST CDT IZP CDTINFO
- RLIST CDT ZOWE CDTINFO

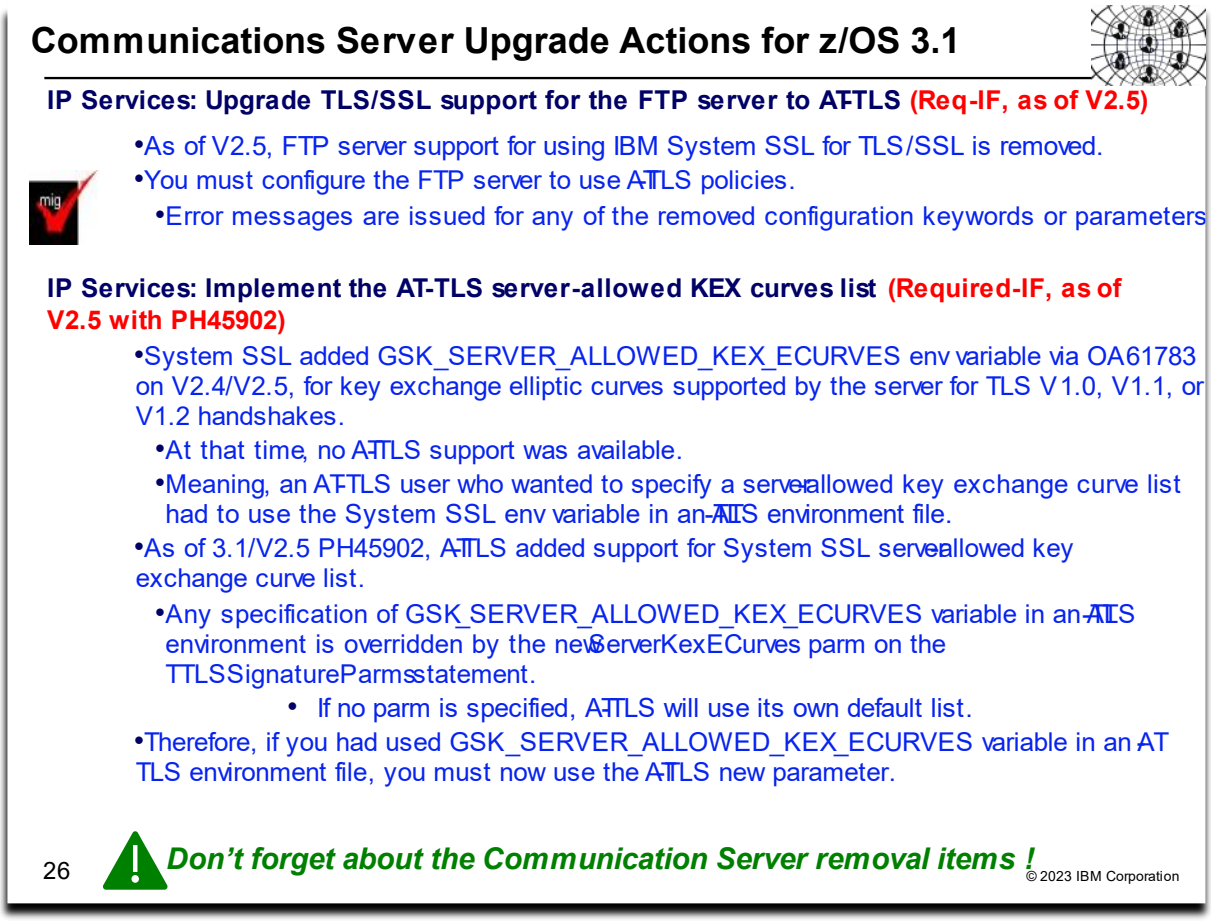
I**mportant:** Ensure that the dynamic class was defined compatibly with the IBM class, as documented by Zowe or IBM Unified Resource Manager. Specifically, verify that the POSIT value matches what is documented for the class

in *z/OS Security Server RACF Macros and Interfaces* (607 for ZOWE, 608 for IZP). If not, change the POSIT value on your current release before upgrading to V2.5. The *Security Administrator's Guide* contains instructions on changing a POSIT number in the topic titled "Changing a POSIT value for a dynamic class."

When all the systems that share the RACF database are upgraded to z/OS V2.5, delete the classes as follows:

- RDELETE CDT IZP
- RDELETE CDT ZOWE
- SETROPTS RACLIST(CDT) REFRESH

In the interim, the ICH14079I messages can be ignored. After the classes are deleted from the RACF CDT class and the RACF CDT class is refreshed, the message is no longer issued. There is no need to alter any profiles in the IZP or ZOWE class.



Communications Server Upgrade Actions For z/OS 3.1
These upgrade actions were taken from *z/OS 3.1 Upgrade Workflow.* Some descriptions and actions have been shortened for inclusion in this presentation. Not all upgrade actions have been included. For the complete descriptions and actions, refer to *z/OS 3.1 Upgrade Workflow.*

Communications Server Actions You Can Do Now

mig **IP Services: Upgrade TLS/SSL support for the FTP server to AT-TLS (Req-IF, as of V2.5)**

Required if you are using native TLS/SSL support for the FTP server (TLSMECHANISM FTP).
As of z/OS V2R5, FTP server support for using IBM System SSL for TLS/SSL is removed. You must configure the FTP server to use AT-TLS policies. FTP configuration error messages are issued if any of the removed configuration keywords or parameters are configured for FTP servers.
Upgrade action: See the *z/OS V2.5 Upgrade Workflow for details* – the overview steps are provided here:

1. Configure AT-TLS and Policy Agent.
2. Configure the FTP server to use AT-TLS by coding TLSMECHANISM ATTLS in FTP.DATA.
3. If TLSRFCLEVEL CCCNONOTIFY is configured in FTP.DATA, update TLSRFCLEVEL to have a valid value for AT-TLS. If your FTP server uses TLSRFCLEVEL CCCNONOTIFY, change it to TLSRFCLEVEL RFC4217.
4. Migrate existing FTP server configuration to AT-TLS.
5. Migrate existing ciphers coded on CIPHERSUITE statement in FTP.DATA to AT-TLS TTLSCipherParms statements.
6. ATTLS supports more secure TLS versions and ciphers. Consider enabling TLSv1.2 or TLSV1.3 on the TTLSEnvironmentAdvancedParms or TTLSConnectionAdvancedParms statement.

Communications Server Actions Before First-IPL

IP Services: Implement the AT-TLS server-allowed KEX curves list (Req-IF, as of V2.5 with PH45902)
Required if an AT-TLS policy is being used on the system and that policy specifies the use of an AT-TLS environment file that contains the System SSL GSK_SERVER_ALLOWED_KEX_ECURVES variable.

System SSL APAR OA61783, in z/OS V2R4 and z/OS V2R5, added the GSK_SERVER_ALLOWED_KEX_ECURVES environment variable. This variable was used to specify the list of key exchange elliptic curves that are supported by the server during a TLS V1.0, TLS V1.1, or TLS V1.2 handshake. When this variable was introduced, no corresponding AT-TLS support existed. Thus, if an AT-TLS user wanted to specify a server-allowed key exchange curve list, it was necessary to code the System SSL environment variable in an AT-TLS environment file.
Now, with z/OS 3.1 (and z/OS V2R5 with APAR PH45902), AT-TLS adds support for the System SSL server-allowed key exchange curve list. With this change, any specification of the GSK_SERVER_ALLOWED_KEX_ECURVES variable in an AT-TLS environment file is overridden by the new ServerKexECurves parameter of the TTLSSignatureParms statement. If no ServerKexECurves parameter is specified, AT-TLS uses its own default list.
Because of this change, configurations that specify the GSK_SERVER_ALLOWED_KEX_ECURVES variable in an AT-TLS environment file must be updated to use the new AT-TLS policy parameter.
System impact: If the action is not taken and the specified conditions exist, the server allowed KEX curves specified in the AT-TLS environment files are ignored and the AT-TLS default is used.
Steps to take: To determine whether you currently use the GSK_SERVER_ALLOWED_KEX_ECURVES variable, do the following:

1. On your z/OS V2R5 and z/OS V2R4 systems, for each affected z/OS TCP/IP stack, enter the **pasearch -t** command to list all the AT-TLS rules currently in use and redirect the command output to a z/OS UNIX file, for example: **pasearch -t > /tmp/attlsrules.txt**. The pasearch command generates a consolidated report of each installed AT-TLS rule, including each of the conditions and actions that make up the rule.
2. Search the generated file (/tmp/attlsrules.txt in the example above) for each occurrence of the label "Envfile:" to locate all the references to AT-TLS environment files or data sets.
3. For each unique AT-TLS environment file, examine its contents. If it contains a GSK_SERVER_ALLOWED_KEX_ECURVES variable, do the following:
    1. Write down the name of the environment file
    2. Write down value specified on the GSK_SERVER_ALLOWED_KEX_ECURVES variable
    3. Note whether the file contains other environment variables in addition to the GSK_SERVER_ALLOWED_KEX_ECURVES variable.
    4. Search for and write down the name of each AT-TLS rule that references the environment file. The rule name appears before the environment file specification and can be found by searching for the preceding “policyRule:” label.

    If the environment file does not contain a GSK_SERVER_ALLOWED_KEX_ECURVES variable, you can ignore it.
    If you use the z/OSMF Network Configuration Assistant to maintain your AT-TLS policy:

    5. Using the notes you took during the previous steps, log into the z/OSMF Network Configuration Assistant AT-TLS perspective. For each affected TCP/IP stack, go to the main Connectivity Rules dialog. Notice that the policyRule names you noted earlier contain the names of the Connectivity Rules for which they were generated. Note that one Connectivity Rule can result in the generation of multiple policyRules.
    6. For each affected Connectivity Rule, do the following:
        1. Add the curves specified on the GSK_SERVER_ALLOWED_KEX_ECURVES variable to the server key exchange curve list under the signature parameters for each Security Level associated with the Connectivity Rule
        2. Remove the GSK_SERVER_ALLOWED_KEX_ECURVES from the environment file. If this action renders the file empty, you can delete the file.
        3. If the environment file is empty or was deleted, deselect the use of the environment file for the Connectivity Rule.

        If you hand-code your AT-TLS policy files, using the notes you took during the previous steps, edit each affected AT-TLS policy file. For each environment file that contains a GSK_SERVER_ALLOWED_KEX_ECURVES variable, do the following:

        1. Find each TTLSGroupAction statement that references the environment file
        2. For each of those TTLSGroupAction statements, find each TTLSRule statement that references the TTLSGroupAction.

3. For each of those TTLSRule statements, find the TTLSEnvironmentAction statement it references. **Note:** You can also reference TTLSSignatureParms statements from the TTLSConnectionAction, if necessary.

   Do the following:
   - If the TTLSEnvironmentAction statement references a TTLSSignatureParms statement, add a ServerKexECurves parameter with the specified GSK_SERVER_ALLOWED_KEX_ECURVES value to that TTLSSignatureParms statement.
   - Otherwise, create a new TTLSSignatureParms statement that specifies the ServerKexECurves parameter with the specified GSK_SERVER_ALLOWED_KEX_ECURVES value. Note that if you have already created such a new TTLSSignatureParms statement for another TTLSEnvionmentAction statement, you can point this TTLSEnvironmentAction statement to that new TTLSSignatureParms statement. You do not have to create a new TTLSSignatureParms statement for each TTLSEnvironmentAction statement (multiple TTLSEnvironmentAction statements can reference the same TTLSSignatureParms statement).
   - After the above steps are complete for each of the affected TTLSEnvironmentAction statements, you can delete the GSK_SERVER_ALLOWED_KEX_ECURVES variable from the environment file. If this renders the file empty, you can delete it and remove the Envfile parameter from the TTLSGroupAction statement.

IP Services: Update /etc configuration files (Required-IF)

Required if you have customized a configuration file that IBM has changed.

Some utilities provided by Communications Server require the use of certain configuration files. You are responsible for providing these files if you expect to use the utilities. IBM provides default configuration files as samples in the /usr/lpp/tcpip/samples directory. Before the first use of any of these utilities, you should copy these IBM-provided samples to the /etc directory (in most cases). You can further customize these files to include installation-dependent information. An example is setting up the /etc/osnmpd.data file by copying the sample file from /usr/lpp/tcpip/samples/osnmpd.data to /etc/osnmpd.data and then customizing it for the installation.

If you customized any of the configuration files that have changed, then you must incorporate the customization into the new versions of the configuration files.

"Big Migs“ occurring on V2.5

Upgrade actions at V2.5 you should not overlook:

1. z/OSMF ServerPac driving system requirement.
2. HFS removal
3. Use only SAF-based security to protect SDSF functions
4. Activate JES2 z22 mode
5. Perform updates for RMF structural changes

Plus…

27

"Big Migs“ occurring on 3.1

Upgrade actions at 3.1 you should not overlook:

1. IBM JES3 removal
2. Use dynamic statements for SDSF configuration, not assembler macros.
3. Sysplex couple data sets are System Status Detection (SSD) capable
4. OpenSSH new ported level, 8.4p1
5. z/OSMF Workflow definition files for undeclared referenced entities, before 3.1 IPL.
6. zCX Workflow instances should be completed, before 3.1 IPL .

28

Upgrading to z/OS V3.1: Planning Summary

- **Changing content of z/OS 3.1**
  - •New z/OS content: XML Toolkit, Change Tracker, DFSMStvs is included in the base
  - •Removed in z/OS V2.5: HFS
  - •Removed in z/OS 3.1: JES3, ISFPARMS, Alternate Base, CommServer Security L3, …
- **Timeline of z/OS ordering and deliverables:** associated products to consider
- **z/OS Policies**
  - •Three consecutive releases (V2.4 → 3.1) for coexistence, upgrade, fallback.
- **Ensuring System Requirements are Satisfied**
  - •Driving, Target SW, Target HW, and Coexisting System Requirements
    - •z/OS 3.1 requires z14 or later
    - •Memory requirements: 8 GB "native", or 2 GB zPDT or under zVM of memory
    - •Target system requirements: general Java SDK 11 functional requirement
    - •Use FIXCATs `IBM.TargetSystem-RequiredService.z/OS.3.1`, `IBM.Coexistence.z/OS.3.1`, `IBM.Function.HealthChecker`, `IBM.TargetSystem-RequiredService.Semeru.*` …
- **Use IBM Health Checker for Upgrade Actions:** Install and activate checks
- **Use z/OSMF Workflow for z/OS 3,1 Upgrade:** Goodbye, book!
- **z/OSMF is the ServerPac driving system as of July 10, 2022!**
  - This includes all ServerPacs, including middleware.

60

Upgrading to z/OS 3.1: Technical Actions Summary

- **General:** New address space, new and old data sets, changed checks.
- **BCP:**
  - •Non-executable storage used for passing parameters, `OPTIMIZE=YES` default for SVC dump processing, ALLOC's tape library request default is `BYDEVICES`, ASVT resides above 16M, new meaning and default for `OSPROTECT`
- **JES2:**
  - •Default usage of new job level resource limits and actions, truncation of blanks which affects some printing products.
- **HCD:** Out of service processor types are removed.
- **RMF:**
  - •Remove references to deprecated ports, remove RMF Postprocess XML toolkit from workstation.

29