#35 zNibbler (z/OS’ Application Integration with Release 13)
zTidBits Series

z/OS V1.13 introduces many capabilities to help you write new applications and systems programs, and extend existing programs. Businesses with applications on z/OS understand the value of its qualities of service, including availability, scalability, and security for these applications and their data on z/OS.

Extending these critical applications and expanding the access to the z/OS data hub can drive business agility, enhance usability, and provide unprecedented levels of business integration.

Batch is just such a critical business workload. According to IBM research, about 90% of respondents consider batch to be mission critical with the majority choosing to run it on System z. Central to batch processing is the COBOL programming language. COBOL is simple, efficient, robust, and scalable. With hundreds of billions of lines of code, COBOL assets are almost everywhere and capable of supporting billions of transactions a day. Top analysts agree you can take advantage of COBOL's modern interfaces to help revolutionize your batch processing.

z/OS V1.13 Batch Runtime, a new base component, and associated new function are intended to form the foundation for a powerful, integrated, and modern batch application development, deployment, and runtime environment. The z/OS Batch Runtime provides the framework for Java to COBOL interoperability, for transactional updates to DB2, and for sharing database connections between Java and COBOL, enabling you to extend your COBOL batch applications with Java. This new function is intended to help you reduce costs while improving business agility and the operational efficiency of your batch environment.
In addition, JES2 enhancements are designed to make programming JCL easier and to give you more control of your batch applications. Functions such as in-stream data in cataloged procedures, more options on reporting the return code for jobs, and the ability to stop and hold a job at the end of a step (not just at the end of the job) give much more granularity and control.

Extend the power of z/OS batch even more with a new REST API (included with z/ OSMF V1.13), which can allow you to submit, query, and retrieve output for z/OS batch workloads from both z/OS systems and other systems.

In addition to updates for your batch environment, additional application development enhancements are included for z/OS UNIX System Services, Language Environment, Program Management Binder, z/OS dbx debugger, z/OS Unicode, DFS SMB Server, and more.

Additional function, descriptions, and details in support of application integration are included for z/OS V1.13:

- z/OS V1.13 introduces many **new batch programming capabilities**:
  - The z/OS Batch Runtime environment is intended to form the foundation for resilient, highly available, secure, and scalable services for batch applications.

    > The z/OS Batch Runtime is intended to provide the framework for Java to COBOLinteroperability, for transactional updates to DB2, and for sharing database connections between Java and COBOL.

    NOTE: New Java-COBOL interoperability capabilities are designed to enable you to reuse valuable COBOL assets by developing new or enhancing existing batch applications with Java. With this capability, for example, you can share a database connection between COBOL and Java allowing you to use Java subroutines directly in lieu of Java stored procedures. The Java portions of your batch applications can be eligible for the IBM System z Application Assist Processor, or the IBM System z Integrated Information Processor, using the zAAP on zIIP capability introduced with z/OS V1.11.

    > Initially, Java, DB2, and COBOL support is provided with IBM 31-bit SDK for z/OS, Java Technology Edition, Version 6.0.1, DB2 V9.1 for z/OS, or later.

  – In z/OS V1.13, several batch enhancements are provided for JES2 environments that are intended to help simplify the development of batch applications.

> Support is provided for in-stream data sets to be used within JCL procedures and for include statements. This is designed to improve usability of JCL procedures; for example, by making it possible for you to put utility control statements in the same member as the rest of the procedure.

> Support is provided to allow jobs for which journaling is used to be stopped after a currently running step has finished and held for restart in the following step. This is intended to allow less-disruptive system shutdowns.

> Job-level return code support is provided. A new JOBRC keyword on the JOB statement is designed to allow you to specify whether a job's return code should be set to the highest return code of any step that was run, the return code of the last step, or the return code of a specific step. This new support can make it simpler to determine whether a job succeeded or failed by looking in a single place for the result.

> Support for specifying a time, interval, or number of output lines on the UNALLOC parameter of the SPIN keyword on the DD statement. This is intended to help improve availability for long-running address spaces.

- In addition to applications designed to simplify system management tasks, z/OSMF V1.13 also delivers a **new API** you can use to submit batch jobs and retrieve batch job information from z/OS using distributed systems as well as z/OS systems. This new API, which is HTTPS protocol-based and uses **Representational State Transfer (REST),** is introduced with z/OS V1.13 with z/OSMF V1.13. This is intended to make z/OS batch processing much more accessible to distributed systems and web-based processes.

- Batch concurrency can be improved. A **new FREEVOL=EOV** keyword on the JCL DD statement allows you to specify that volumes of a multivolume tape data set be made available for other processing once the processing for each volume is finished. This is intended to allow overlapped processing for multivolume data sets, which can speed batch processing.

- A number of **DFSORT programming enhancements** are now available**:**

  - These include a variety of new functions for translating ASCII, EBCDIC, hex, and binary characters; a new RESIZE operator for ICETOOL that you can use to change output record lengths, new date field arithmetic operations, a new ACCEPT option for the OUTFIL statement you can use to process subsets of output records, a new grouping function based on key changes, support for SET and PROC symbol substitution in control statements, more information in reports, and a number of other enhancements. These changes provide new functions and improved flexibility, and can help simplify the use of DFSORT.

NOTE: For more information about these new functions, see the User's Guide for DFSORT:
http://www.ibm.com/support/docview.wss?rs=114&uid=isg3T7000242
See bottom of this web page -

View a PDF version of SORTUGPH (107KB)

- z/OS **V1.13 XL C/C++** includes these performance and usability enhancements:
  - **For Metal C**, new support is provided for**:**

    > IPA and HOT options, which provide interprocedural analysis and high-order loop analysis and transformations during optimization

    > Function property blocks to help you identify the C functions and their associated properties when code scanning or dump reading

    > A DSAUSER option you can use to request that a field the size of a pointer be reserved on the stack.

    > An ARGPARSE option for automatic parsing of entry point options.

    > The qsort() function, which allows an array to be sorted using a function you supply, and is intended to

relieve Metal C programmers from having to write sort routines with similar capabilities.

- **New ARCH(9) functions** for programs running on zEnterprise System servers, including support for the interlocked storage access instructions, and for Multiply and Add in hexadecimal floating-mode with a new combination of FLOAT(MAF) and FLOAT(HEX) options.
- **Informational messages**, specified with FLAG(I), are now issued by default in z/OS UNIX System Services for consistency with batch compilations.
- TEMPLATEDEPTH option for C++ for increased template control, to specify the maximum number of recursively instantiated template specializations to be processed by the compiler.
- Additional source and binary compatibility support, including**:**
  - > Suppress warnings for text following #else and #endif
  - > Function attributes (gnu_inline, used, malloc)
  - > Temporary lifetime extensions (C++ only)
  - > Rvalue bindings to a non-constant reference (C++ only)
  - > Intrinsic complex types added to C++
  - > Addressable labels
- **New C++0x function**, trailing return type, whereby the compiler deduces the type of an auto variable from the type of its initializer expression.
- **Debugging enhancements:**
  - > In prior releases, the z/OS dbx debugger for C/C++ programs requires that the compiler insert Execute (EX) instructions (called hooks) so that the debugger can gain control during program execution to display information about the program and the data it processes. In z/OS V1.13, dbx provides support for debugging programs compiled without hooks, in addition to those compiled with hooks. This support is intended to allow you to debug programs whose sizes and performance characteristics are more closely aligned with production programs.
  - > New debugging APIs provide easier access to debug information in .mdbg and .dbg files.
  - > Debug information for inline procedures, gives the ability to set entry breakpoints at all inline instances.

- **Language Environment** supports recovery from additional abends during output and close operations for C/C++ programs, and can return to C/C++ programs, indicating that an I/O error has occurred rather than issuing an abend. This is intended to provide a more predictable recovery environment for C/C++ programs when I/O errors are encountered.

- **The Base Control Program internal interface** (**BCPii**) allows authorized programs to perform functions related to CPC Image User groups on IBM zEnterprise and IBM System z10® servers with a minimum microcode level (MCL). New function in z/OS V1.13 is designed to allow available CPC Image User groups on a particular CPC and their member images to be listed in, connected to, and disconnected from groups; to support queries for group attributes; and to support issuing most HWICMD BCPii commands to all the members of a specified group.

- **Support in z/OS UNIX System Services** for user-level file system mounts is designed to provide support for limiting overall user mounts, the ability to allow these functions for specific users and groups of users, the ability to restrict which mount points a user may use, and the ability to allow user mounts only at empty mount points. Also, improved warning and failure messages are provided for some mount and unmount operations. The ability to move these mounts from system programmers or administrators directly to users can improve usability and flexibility, and make it easier to use z/OS UNIX.

  z/OS UNIX System Services is enhanced with:

  - A script command you can use to record the output of a shell session. For example, a script command could be used to create a session log file for auditing or another output file intended to be processed later.

- Updates to the vi and ex editors that allow you to edit untagged text files and have them treated as if they contained text data using a codeset you specify. New support also allows you to override the built-in autoconversion function, making it easier to edit ASCII-encoded files under z/OS UNIX.
- A capability that allows you to specify that IPv4 UDP datagram reply packets must flow on the same interface where the request arrived when a server system has multiple home addresses with multiple routes back to the client or is using a DVIPA. This support, designed to be similar to the existing support for IPv6, is intended to allow applications to require that the response to a request be restricted to the same IPv4 address from which the request was received.

- **The Program Management Binder** supports new operations for unnamed ($PRIVATE) sections in load modules and program objects. This new design allows you to specify that all unnamed sections be deleted or that specific unnamed sections be named (using CHANGE statements) or replaced. These new functions are intended to make it easier to maintain load modules and program objects in place while avoiding unnecessary growth and virtual storage use due to an accumulation of unnamed sections.
- Support is provided to allow tasks in a **WLM enclave** that have subtasks to leave that enclave and for subtasks of a task joining a WLM enclave to be joined automatically to that enclave. In prior releases, enclave tasks with subtasks are not allowed to leave an enclave. Any subtasks created by a task that has joined an enclave are automatically added to the same enclave. This new support is designed to allow a task to leave an enclave along with its subtasks. This new support also allows a task joining an enclave to have its non-enclave subtasks joined to that enclave. This support is also available on z/OS V1.11 and z/OS V1.12.
- The **DFS SMB Server** supports clients running both the 32- and 64-bit versions of Microsoft Windows 7 Professional, Microsoft Windows 7 Enterprise, and Microsoft Windows 7 Ultimate Editions. This support is also planned to be available for z/OS V1.11 and z/OS V1.12.
- **NFS** provides support for the 32- and 64-bit versions of Microsoft Windows 7 Professional Edition with Open Text NFS Client or Open Text NFS Server installed.
- **Language Environment** adds support for initializing multiple CEEPIPI main environments under one task control block (TCB) and to provide access to a user word for each environment both within and outside it. This is intended to help you migrate Preinitialization Compatibility Interface (PICI) environments to CEEPIPI.
- **z/OS Unicode Services** implement improved bidirectional character support for applications that process scripts such as those for the Arabic and Hebrew languages. These languages, among others, are written and read from right to left but often contain strings that read from left to right. This new function is designed to support such nested bidirectional ("Bidi") strings. Samples are supplied that show how to use these extended bidirectional services, and a sample object file is supplied that you can include with C applications to make it easier to use z/OS Unicode Services functions defined by The Open Group's Bidi interface. Also, the z/OS Unicode conversion information service provides additional information about Coded Character Set Identifiers (CCSIDs), including information to identify substitution, newline, line feed, carriage return, end-offile, and space character codes.

- **Storage management** is changed to allow tasks using subspaces to access 64-bit private and 64-bit shared virtual storage without the overhead of a Branch in Subspace Group (BSG) instruction. This is intended to help provide virtual storage constraint relief by making it easier for applications to exploit 64-bit storage and to use system services that use 64-bit storage.

- These **IBM Tivoli Directory Server for z/OS (LDAP**) enhancements are provided:
  - DB2-based TDBM back end and bulk load utility are enhanced to support 64-bit addressing. This enhanced TDBM back end, when used with DB2 9 for z/OS or a later release, is intended to improve scalability of IBM Tivoli Directory Server for z/OS for large LDAP deployments.

    In addition to this TDBM enhancement, a bulk load utility capable of executing in 64-bit addressing mode facilitates loading large LDAP directory databases.
  - Support is extended to enable Kerberos binds to be processed by Microsoft's Active Directory Server. This support is intended to improve the interoperability between z/OS applications that utilize the IBM Tivoli Directory Server for z/OS client services and Kerberos authentication in environments where Active

Directory is being utilized.

- In z/OS V1.12, a **Java API was implemented for SDSF** to allow Java applications access to the data available through SDSF panels. That API is a set of classes, with each class corresponding to a panel, with each instance representing a row, with methods to perform operations similar to action characters and overtypes, and with support for filtering. In z/OS V1.13, support is provided for returning a subset of the requested data. For example, if a set of filters match a large number of SDSF instances normally all returned in a list, a sublist can be requested in a similar manner where the caller can specify relative starting and ending indexes to limit the number of instances to be returned. This new support is intended to allow Java applications to request smaller amounts of data from SDSF.

- In z/OS V1.13, **SDSF REXX** support is provided for reading the sysplex-wide operations log (OPERLOG), in addition to the single system log (SYSLOG). The support for OPERLOG is designed to be very similar to that for SYSLOG, allowing records to be selected by start time and date and providing the ability to specify a maximum number of records to be returned. Additionally, the OPERLOG display is designed to show messages in the same colors in which they would be displayed on a console. An SDSF Java interface also supports access to OPERLOG. These enhancements are intended to make it easier for you to perform complex repetitive functions programatically using SDSF.

- In z/OS V1.13, **z/OS XML System Services** supports a binary XML format, Extensible Dynamic Binary XML (XDBX). XDBX supports a subset of XML constructs in a new binary form, and appropriate use of XDBX is expected to provide performance improvements for validating parsing operations compared to conventional XML text documents. This function is planned to be enabled on z/OS V1.13.

More specific ISPF R13 enhancements:

ISPF enhancements

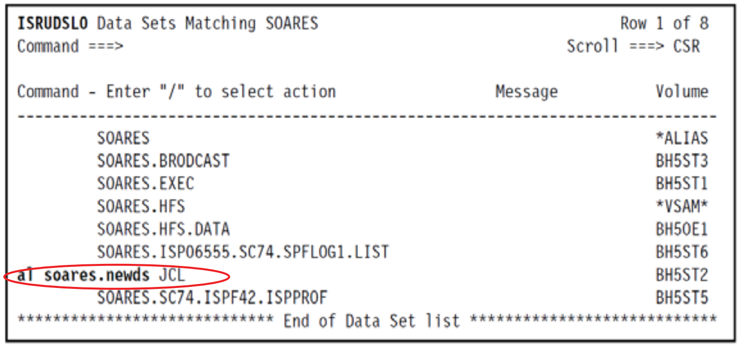
ISPF is a development tool set for the z/OS operating system. Since 1975, programmers have used ISPF for host-based application development productivity. ISPF forms the basis of many TSO and CMS applications and also provides extensive programmer-oriented facilities. System Display and Search Facility (SDSF) is a utility that allows you to monitor, control, and view the output of jobs in the system. After submitting a job, it is common to use SDSF to review the output for successful completion or to review and correct JCL errors. SDSF allows you to display printed output held in the JES spool area. Much of the printed output sent to JES by batch jobs and other jobs is never printed. Instead, it is inspected using SDSF and deleted or used as needed.

ISPF editor support for line command macros

In z/OS V1R13, ISPF provides support for line command-level Edit macros, in addition to the existing Edit macro support. This new design is intended to allow you to write macros to be used as line commands, in addition to those you might have already written for use as Edit primary or initial processing commands.

Support for new data set allocation commands from ISPF Data Set List

In z/OS V1R13, ISPF provides a new `AL` line command on the Data Set List panel (option 3.4) command that you can use to allocate a new data set using a specified data set as a model for the new data set's attributes.

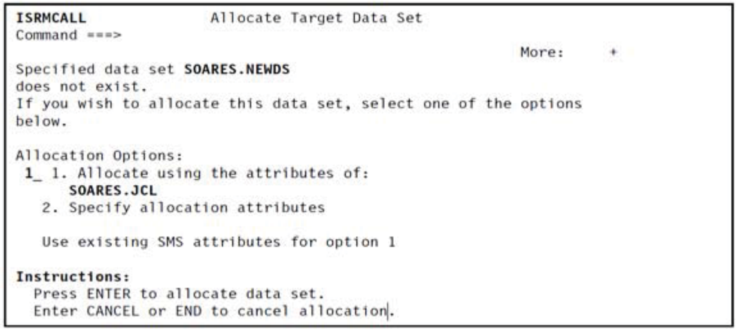


```
ISRUDSLO Data Sets Matching SOARES                               Row 1 of 8
Command ===>                                                 Scroll ===> CSR

Command - Enter "/" to select action                  Message           Volume
-------------------------------------------------------------------------------
         SOARES                                                         *ALIAS
         SOARES.BRODCAST                                                BH5ST3
         SOARES.EXEC                                                    BH5ST1
         SOARES.HFS                                                     *VSAM*
         SOARES.HFS.DATA                                                BH50E1
         SOARES.ISP06555.SC74.SPFLOG1.LIST                              BH5ST6
al soares.newds JCL                                                     BH5ST2
         SOARES.SC74.ISPF42.ISPPROF                                     BH5ST5
**************************** End of Data Set list ****************************
```

When the **AL** line command is entered against an existing data set, a panel is displayed offering the following options:

- Create the data set using the attributes of the existing data set
- Specify the attributes of the data set on the Allocate New Data Set panel



```
ISRMCALL              Allocate Target Data Set
Command ===>
                                                   More:     +
Specified data set SOARES.NEWDS
does not exist.
If you wish to allocate this data set, select one of the options
below.

Allocation Options:
 1_ 1. Allocate using the attributes of:
       SOARES.JCL
    2. Specify allocation attributes

    Use existing SMS attributes for option 1

Instructions:
  Press ENTER to allocate data set.
  Enter CANCEL or END to cancel allocation.
```

Using the AL command for a deleted data set

You do not need to provide a data set name if the **AL** command is issued against a data set that has just been deleted. The name of the deleted data set will be used for the new data set.

ISPF: z/OS UNIX Directory List enhancements

In z/OS V1R13, ISPF adds a new option to the z/OS UNIX Directory List panel to allow authorized users to update access control lists (ACLs) from within ISPF.

SDSF EAV support for output data sets

In z/OS V1AR13, SDSF supports extended format sequential (DSNTYPE=LARGE) print files, and print files that are placed in the extended addressing space (EAS) of an extended address volume (EAV).

Eliminate MQ requirement for sysplex requests - SDSF

In z/OS V1R13, SDSF provides sysplex-scope displays, including those for the system log (SYSLOG), devices, and initiators, without the use of MQSeries after all systems in the sysplex are at the z/OS V1R13 SDSF level. This new function is designed to use SDSF server address spaces and new XCF services, and to help simplify SDSF setup and operation.

Sundry

INFO command to display PDS member extended statistics

In z/OS V1R11, ISPF provided extended statistics support for partitioned data set members with more than 32,756 lines. In z/OS V1R13, display supports extended statistics with a new INFO command to display extended line counts when the Extended PDS statistics function has been enabled.

Display data set create jobname and stepname

In z/OS V1R13, ISPF provides support for the job and step names stored by the system for data sets that are eligible to reside in the extended addressing space (EAS) on an extended address volume (EAV) in the Data Set List utility (option 3.4), and in the DSLIST and LMDLIST services. This is intended to allow you to find the creating job and step names easily.

New IPCS subcommand DOCPU to perform tasks on specified CPUs
As z/OS supports more and more CPUs, problem determination becomes more complex. In z/OS V1R13, a new IPCS subcommand, DOCPU, is created to repeat the requested tasks on specified CPUs when diagnosing problems using a stand-alone dump.

Provide qsort function within Metal C
In z/OS V1R13, Language Environment adds the qsort() function to the z/OS Metal C Runtime Library. This function allows an array to be sorted using a function you supply. It is intended to relieve Metal C programmers from having to write sort routines with similar capabilities.

zFS automatic takeover of disabled aggregates
In z/OS V1R13, zFS can automatically recover disabled aggregates when possible in both single-system and in sysplex environments when multiple systems are running in zFS sysplex-aware mode. This is intended to eliminate the need to recover the file system manually before applications close and reopen the files to regain access to them.

zFS refresh
In z/OS V1R13, zFS will maintain existing connections to file systems while recovering from internal errors when possible. This is intended to provide less-disruptive recovery from most internal zFS problems, and designed to allow applications with open files to retry file system operations successfully after zFS recovery has been completed.

zFS client direct I/O
In z/OS V1R13, zFS processing has been redesigned to allow all members of a sysplex to perform zFS file system read and write I/O operations. This is expected to yield substantial performance gains for systems that would not have been zFS owning systems in the prior design, without performance impacts to systems that would have been zFS owning systems.

Mount/unmount granularity to prevent/warn mount overlay on a subtree
In z/OS V1R13, support in z/OS UNIX System Services is for user-level file system mounts. This provides support for limiting overall user mounts, the ability to allow these functions for specific users and groups of users, the ability to restrict which mount points a user may use, and the ability to allow user mounts only at empty mount points. Also, improved warning and failure messages are planned for some mount and unmount operations. The ability to move these mounts from systems programmers or administrators directly to users can improve usability and flexibility, and make it easier to use z/OS UNIX.

Shell and Utilities Add option on vi to edit ASCII files
In z/OS V1R13, z/OS UNIX System Services provides enhancements to the vi and ex editors to allow you to edit untagged text files and have them treated as if they contained text data using a codeset you specify. This new support allows you to override the built-in auto conversion function, making it easier to edit ASCII-encoded files under z/OS UNIX.

Provide script utility in z/OS UNIX Shells and Utilities
In z/OS V1R13, z/OS UNIX System Services provides a script command you can use to record the output of a shell session. For example, a script command could be used to create a session log file for auditing, or another output file intended to be processed later.

SMB server support for Windows 7 clients
In z/OS V1R13, the DFS SMB Server supports clients running both the 32-bit and 64-bitversions of Microsoft Windows 7 Professional, Microsoft Windows 7 Enterprise, and Microsoft Windows 7 Ultimate.

NFS Server Windows 7 support

In z/OS V1R13, NFS supports Microsoft Windows 7 with Open Text NFS Client or Open Text NFS Server installed.

LISTCAT support of catalog CSI
In z/OS V1R13, Access Method Services (IDCAMS) supports a new option for the `LISTCAT LEVEL` command. This new option is designed to allow you to specify whether related component names be listed when a data set entry is listed based on the pattern specified by LEVEL. For example, if a cluster name is listed, the new option will allow you to specify whether or not the DATA and INDEX entries are also listed.

– – –