



IEBCOPY

– Teaching an Old Dog New Tricks

Cecilia Carranza Lewis, IBM
STSM - z/OS DFSMS Architecture, Design and Development

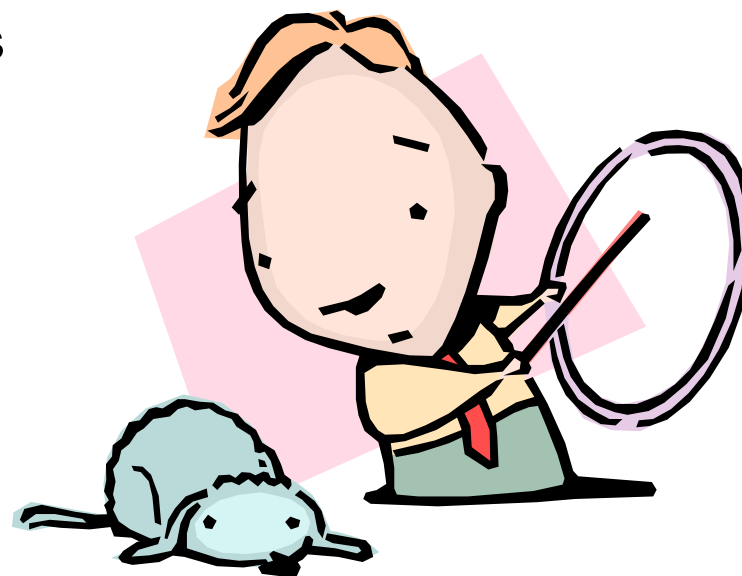
August 11, 2011
Session 9940

Disclaimer

The information on the new product is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information on the new product is for informational purposes only and may not be incorporated into any contract. The information on the new product is not a commitment, promise, or legal obligation to deliver any material, code or functionality. The development, release, and timing of any features or functionality described for our products remains at our sole discretion.

Agenda

- **IEBCOPY – Teaching an Old Dog New Tricks**
 - Basics
 - What's New with z/OS® R13
 - z/OS R13 Performance Results
 - Requirements



IEBCOPY - A Partitioned Data Set's Best Friend



- **History**

- IEBCOPY was originally part of OS/360 (1960's timeframe)
 - PDSEs support added with their introduction in 1989
 - Provided a data set utility to copy or merge members between one or more PDSs or PDSEs, in full or in part
 - Can also be used to create a backup of a PDS or PDSE into a sequential data set (aka unload data set or PDSU)

IEBCOPY Basics



• Typical Tasks

- Make a copy of a PDS or PDSE
 - Select or exclude specific members to be copied, loaded, or unloaded
 - Rename selected members
- Convert a PDS to PDSE or PDSE to PDS
- Merge PDSs or PDSEs
- Replace members of a PDS or PDSE
- Create a sequential form of a PDS or PDSE for backup (aka load/unload)
 - Load/unload one or more members from a PDS or PDSE into a PS data set
- Compress a PDS in place
- Copy to or from a PDSE data set, a member and its aliases together as a group (aka COPYGRP)

• Typical Tasks

- Copy and reblock load modules
- Convert load modules to program objects or the reverse

• Note:

- PDSs contain data members, load modules or a mix
- PDSEs contain data members OR program objects

IEBCOPY - A Partitioned Data Set's Best Friend



- **Basics**

- Make a copy of a PDS or PDSE
 - Copy to different volume or the same volume
 - For PDS, members are not physically reordered but remain in the same order they occur in the original data set
- Select / Exclude specific members of PDS or PDSE to be copied, loaded, or unloaded
 - **Max of 8 characters can be given for the member or alias name**
 - *Must specify every name you want included or excluded, including their aliases*
 - Cannot use both a SELECT and EXCLUDE statement in the same copy operation
 - Selected members are searched for in low-to-high sequence, and once found, **no** search is made for it on any subsequent input data set
 - Excluded member is searched for on **every** input set in the operation and is always omitted when found.

IEBCOPY - A Partitioned Data Set's Best Friend



- **Basics**

- Merge PDS or PDSE (except when unloading)
 - Ensure there is sufficient directory space for a PDS before beginning the merge
 - *Cannot dynamically increase the number of directory blocks for a PDS; must reallocate the PDS with enough space and directory blocks before the merge*
 - *PDSE directory automatically expands as needed*
- Replace members of a PDS or PDSE
 - Can specify replacement at the data set level or member level
 - *At the data set level, all members whose names already exist in the output data set are replaced by members copied or loaded from the input data set*
 - *At the member level, also possible to rename an input member*
 - *Selected member are not renamed in the input data set; they are just added to the output data set with the new name*

IEBCOPY - A Partitioned Data Set's Best Friend



- **Basics**

- Create a sequential form of a PDS or PDSE for backup
 - Create a backup copy by “*unloading*” it to a sequential data set (DSORG=PS) on DASD, tape, or other device supported by QSAM
 - Do not change the DCB parameters of an unload data set or IEBCOPY might not be able to reload it
- Load one or more members from a PDSU into a PDS or PDSE
 - It is possible to create a single PDS/PDSE from multiple input sequential data sets
 - PDS can be loaded from an unloaded PDSE as long as it does not contain program objects
 - PDSE may be loaded from an unloaded PDS as long as it does not contain load modules
- For both unload and load operations, you can chose to process specific members and rename them.

IEBCOPY - A Partitioned Data Set's Best Friend



- **Basics**

- Compress a PDS in place

- Method to remove the unused space in a PDS without copying it to a new data set
- Does not release extents (ie not equivalent to partial release)
- Cannot include, exclude, or rename selected members
- Operation is ignored for PDSE

- **BEWARE**

- *A PDS can be destroyed if the compress-in-place operation is interrupted during processing; recommended to make a backup before doing the compress*
- *Do not compress a PDS being used by more than one user; other users will see the data set as damaged, even though it is not*

IEBCOPY - A Partitioned Data Set's Best Friend



- **Basics**

- **COPYGRP**

- Recommended when copying program objects and their aliases from or to a PDSE
 - ***Ensures that any aliases which are longer than 8 characters are copied along with their member***
 - Valid when either the input, output or both data sets are PDSE
 - *PDSE to PDSE*
 - *PDSE to PDS*
 - *PDS to PDSE*
 - Can also be used for unloading or loading groups:
 - *PDSE to PS*
 - *PS to PDSE*
 - The EXCLUDE statement is not supported

IEBCOPY - A Partitioned Data Set's Best Friend



- **Basics**

- Copy and reblock load modules

- COPYMOD allows you to COPY and reblock the load modules to a block size appropriate for the device to which you are copying
- Maximum block size produced by COPYMOD is 32760 bytes

IEBCOPY - A Partitioned Data Set's Best Friend



- **Basics**

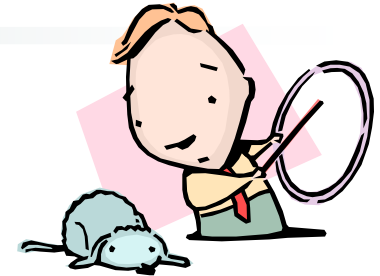
- Convert load modules to program objects or the reverse
 - PDS (load modules) -> PDSE (program objects)
 - *If the PDS contains both load modules and data members, you must separate into two separate PDSEs – one for program objects and one for data members*
 - *Cannot mix program objects and non-program objects in the same PDSE*
 - PDSE (program objects) -> PDS (load modules)
 - *Some program objects cannot be converted if they use features that do not exist in load modules*
 - Cannot directly convert between program objects and load modules when loading or unloading a PDSE or PDS
 - *PDS to PS – a unload operation can only “load” load modules*
 - *PDSE to PS – a unload operation can only “load” program objects*
 - Converting is a multi-step process

IEBCOPY - A Partitioned Data Set's Best Friend



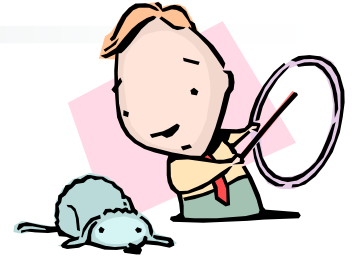
- **Basics – Where to Find More Information**
 - For detailed information on IEBCOPY and how to invoke it, refer to the following publications
 - z/OS DFSMSdfp Utilities (SC26-7414-07)
 - z/OS DFSMS Using Data Sets (SC26-7410-10)
 - IBM Redbook - Partitioned Data Set Extended Usage Guide
 - <http://www.redbooks.ibm.com/abstracts/sg246106.html?Open>

IEBCOPY – Teaching an Old Dog New Tricks



- **What's New with z/OS R13**
 - Customer pain points
 - IEBCOPY should not require its caller (a program) to have APF (authorized program facility) authorization
 - IEBCOPY performance

IEBCOPY – Teaching an Old Dog New Tricks

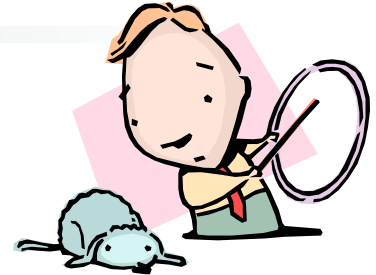


- **APF Authorization**

- Today, IEBCOPY uses channel programming which require it to be run from an authorized library
 - If another program calls IEBCOPY, that program must also be APF-authorized
 - Creates a potential system integrity exposure by requiring existing programs to get authorization although they were not designed with system integrity in mind
- ***New enhancement:*** IEBCOPY removes the requirement for APF authorization
 - If your program calls IEBCOPY, you probably can remove APF authorization from your program, making it safer

? Why it Matters: System integrity is less exposed

IEBCOPY – Teaching an Old Dog New Tricks



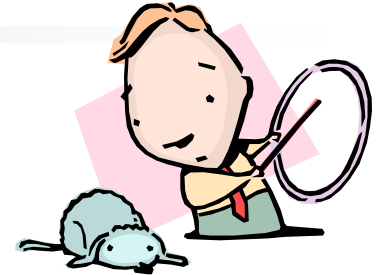
- **IEBCOPY Performance**

- Today, IEBCOPY
 - Uses older channel programming techniques which can impact performance
 - Uses 24-bit addressing which can inhibit use of virtual storage
- ***New enhancements:*** Improved channel programs and use of 31-bit buffers for PDS read operations
 - There is nothing to do to get the improved performance

? Why it Matters: PDS copy performance is enhanced in some cases; we measured elapsed time reductions from 19-70%*

* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.

IEBCOPY – Teaching an Old Dog New Tricks



- **R13 Migration & Coexistence Considerations**

- An old level of IEBCOPY is available in this release by the name of IEBCOPYO (for “OLD”).
 - IEBCOPYO continues to have the APF authorization requirement.

? Why it Matters: Provide a fallback mechanism

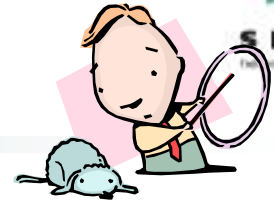
IEBCOPY – Performance Results



- **Performance Methodology**

- Generate data in PDS source data sets with several variations in block and record sizes.
 - Delete half of the data throughout, creating “holey” PDSs to simulate customer-like workload with JCL members deleted from the PDS
 - Read the PDSs in single stream and then concurrently from the same volume
 - *Copy a PDS to a PDS*
 - *Copy a PDS to a sequential data set*
 - *Compress a PDS*
 - *Copy PDS loadlib to PDS loadlib*
 - *Copy PDS loadlib to a sequential data set*
 - *Compress a PDS loadlib*

- **Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.**



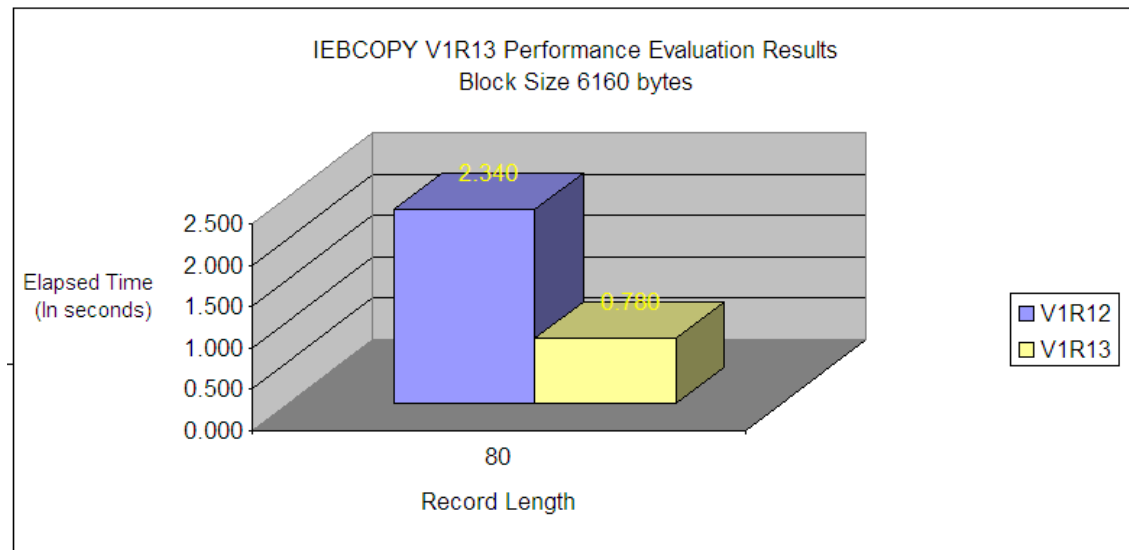
IEBCOPY – Performance Results

Copy PDS to PDS Testing results Block size 6160 Format (FB)

- **Copy PDS to PDS**
 - Copy 1500 members from PDS source to PDS target
 - **Record Format (FB)**
 - **Block size (6160)**
- ★ **~67% throughput improvement**

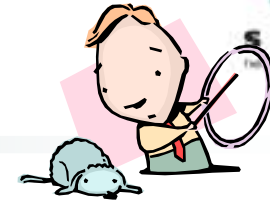
Elapsed Time measurements

LRECL	V1R12	V1R13	Delta (%)
80	2.340	0.780	-66.67
132	*	*	---
4096	*	*	---



Note: * .- Record length is inconsistent with block size for this record format. Test not executed for this variation

* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.



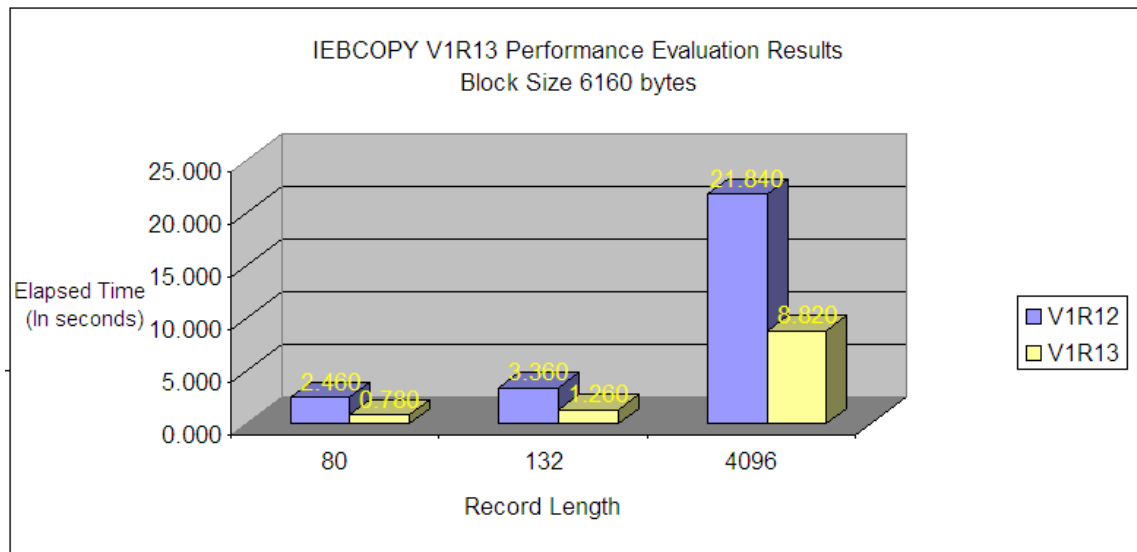
IEBCOPY – Performance Results

- **Copy PDS to PDS (VB)**
 - Copy 1500 members from PDS source to PDS target
 - **Record Format (VB)**
 - **Block size (6160)**
- ★ **~60-68% throughput improvement**

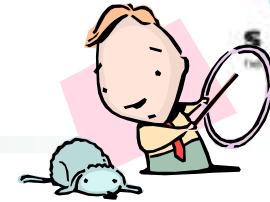
Copy PDS to PDS Testing results
Block size 6160 Format (VB)

Elapsed Time measurements

LRECL	V1R12	V1R13	Delta (%)
80	2.460	0.780	-68.29
132	3.360	1.260	-62.50
4096	21.840	8.820	-59.62



* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.



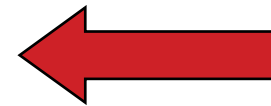
IEBCOPY – Performance Results

Copy PDS to PDS Testing results

Block size Half Track (28332 Bytes) Format (VB)

Elapsed Time measurements

LRECL	V1R12	V1R13	Delta (%)
80	1.860	0.780	-58.06
132	2.220	1.140	-48.65
4096	11.400	8.640	-24.21

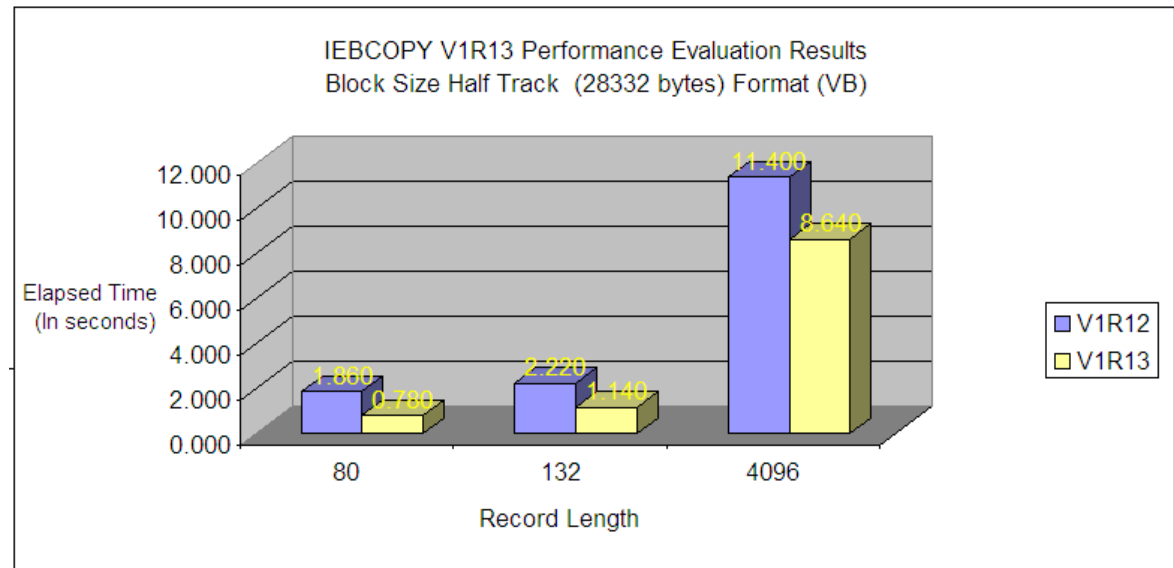


- **Copy PDS to PDS (VB)**

- Copy 1500 members from PDS source to PDS target

- **Record Format (VB)**
- **Block size Half Track (28332 bytes)**

★ **~24-58% throughput improvement**



* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.

IEBCOPY – Performance Results



Copy PDS to SEQ Testing results
Block size 6160 Bytes Format (FB)

- Copy PDS to SEQ**

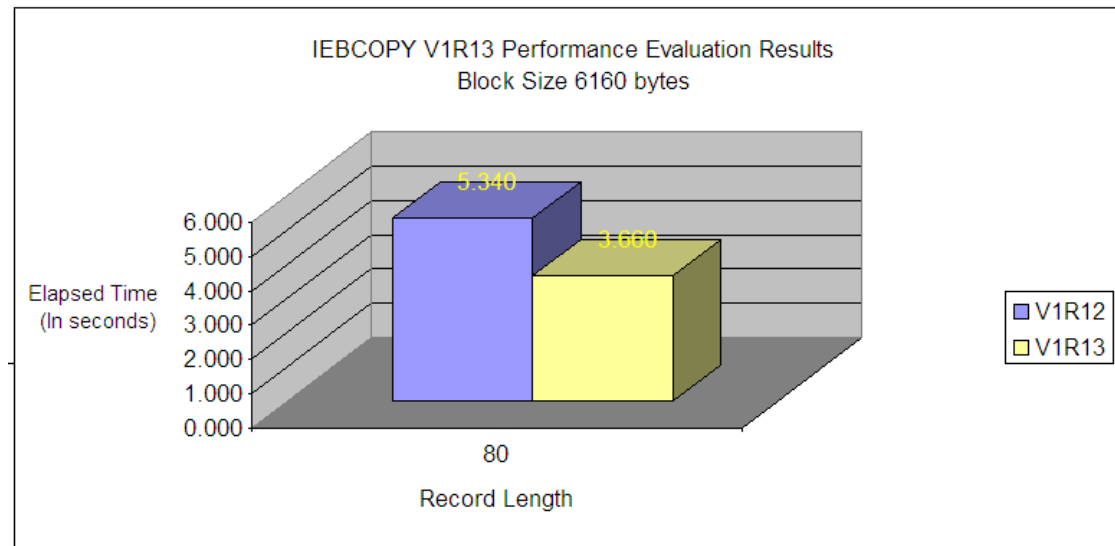
- Copy 1500 members from PDS source to SEQ target

- Record Format (FB)
- Block size (6160)

★ **~31% throughput improvement**

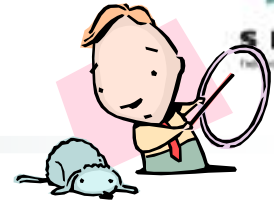
Elapsed Time measurements

LRECL	V1R12	V1R13	Delta (%)
80	5.340	3.660	-31.46
132	*	*	---
4096	*	*	---



Note: * .- Record length is inconsistent with block size for this record format. Test not executed for this variation

* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.



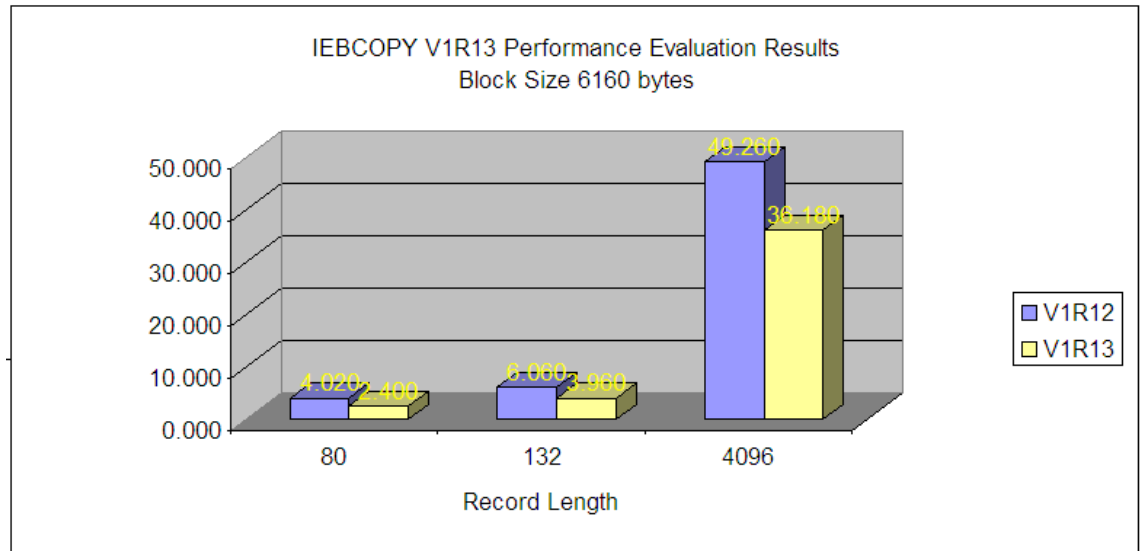
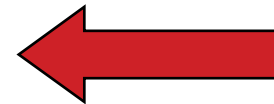
IEBCOPY – Performance Results

Copy PDS to SEQ Testing results
Block size 6160 Bytes Format (VB)

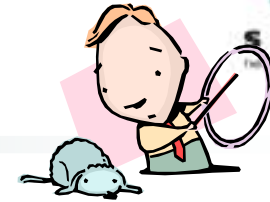
- Copy PDS to SEQ
 - Copy 1500 members from PDS source to SEQ target
 - Record Format (VB)
 - Block size (6160)
- ★ ~26-40% throughput improvement

Elapsed Time measurements

LRECL	V1R12	V1R13	Delta (%)
80	4.020	2.400	-40.30
132	6.060	3.960	-34.65
4096	49.260	36.180	-26.55



* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.



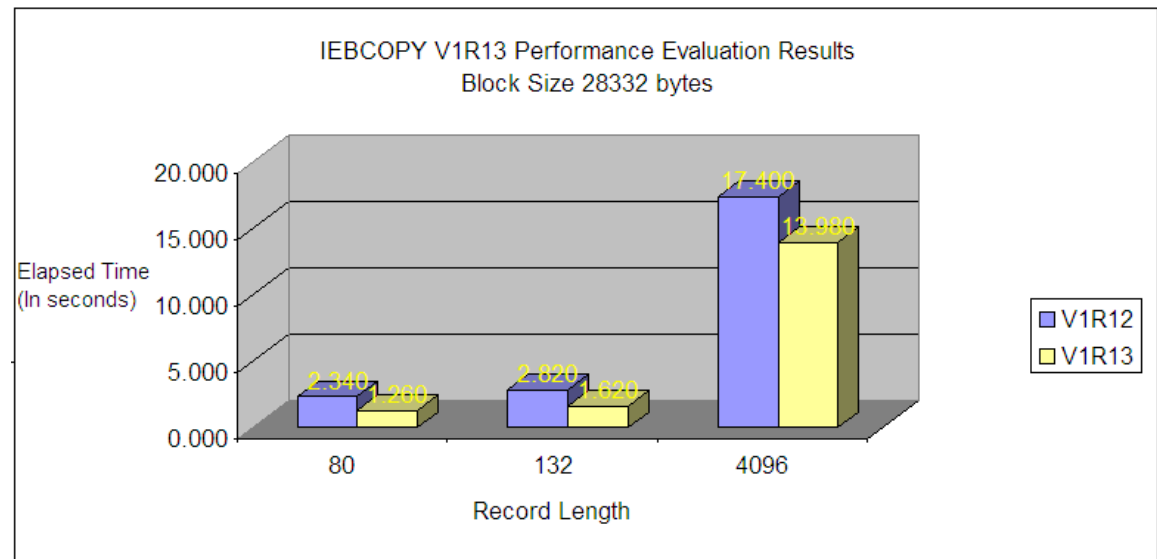
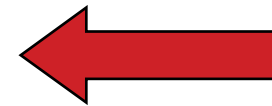
IEBCOPY – Performance Results

Copy PDS to SEQ Testing results
Block size 28332 Bytes Format (VB)

- **Copy PDS to SEQ**
 - Copy 1500 members from PDS source to SEQ target
 - **Record Format (VB)**
 - **Block size Half Track (28332 bytes)**
- ★ **~19-46% throughput improvement**

Elapsed Time measurements

LRECL	V1R12	V1R13	Delta (%)
80	2.340	1.260	-46.15
132	2.820	1.620	-42.55
4096	17.400	13.980	-19.66



* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.

IEBCOPY – Performance Results



• PDS Compress

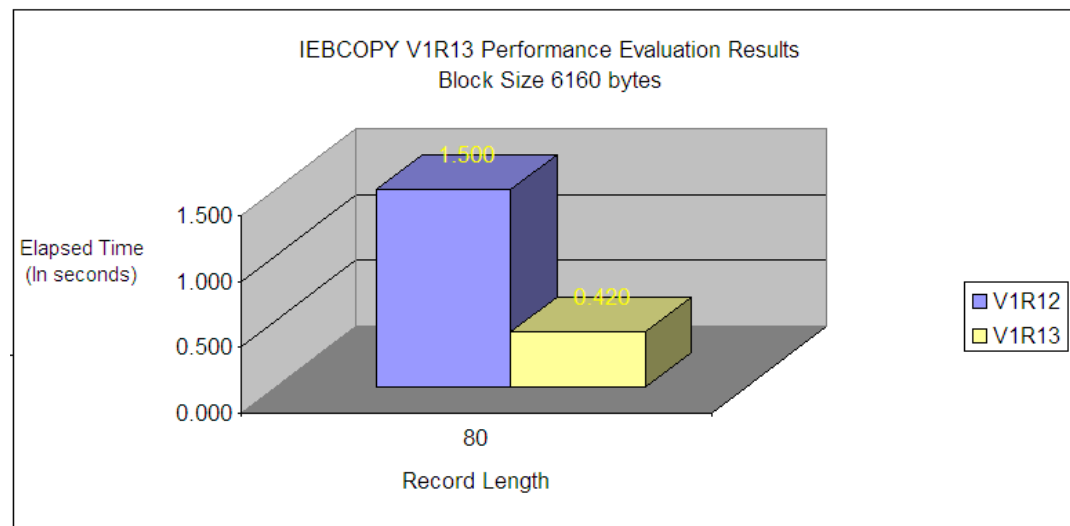
- Deleted 500 members from target PDS prior to compress
 - Prior to delete, target PDS had 1500 members
 - Record Format (FB)**
 - Block size (6160)**

★ **~72% throughput improvement**

Compress PDS Testing results
Block size 6160 Format (FB)

Elapsed Time measurements

LRECL	V1R12	V1R13	Delta (%)
80	1.500	0.420	-72.00
132	*	*	---
4096	*	*	---



Note: * .- Record length is inconsistent with block size for this record format. Test not executed for this variation

IEBCOPY – Performance Results



• PDS Compress

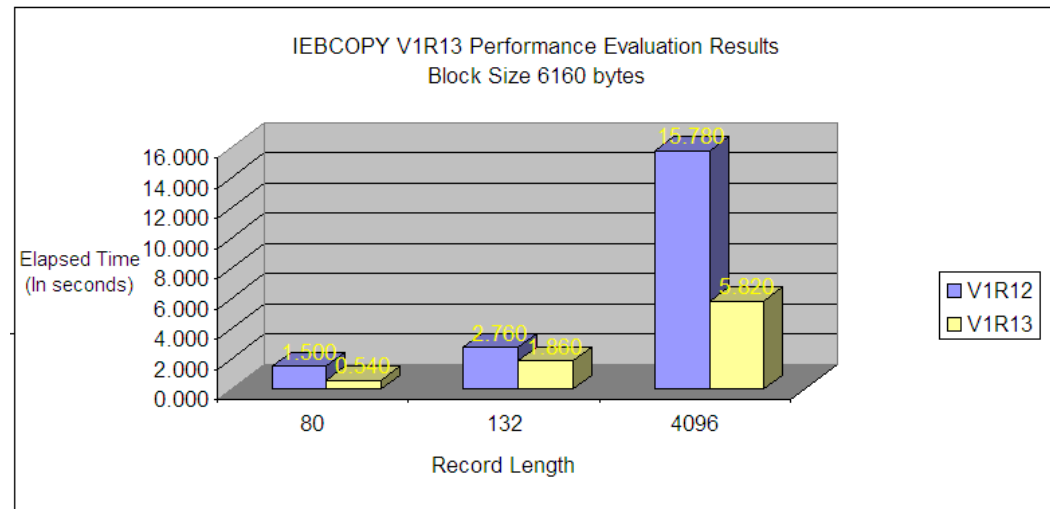
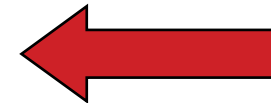
- Deleted 500 members from target PDS prior to compress
 - Prior to delete, target PDS had 1500 members
 - Record Format (VB)**
 - Block size (6160)**

★ **~32-64% throughput improvement**

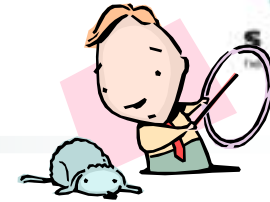
Compress PDS Testing results
Block size 6160 Format (VB)

Elapsed Time measurements

LRECL	V1R12	V1R13	Delta (%)
80	1.500	0.540	-64.00
132	2.760	1.860	-32.61
4096	15.780	5.820	-63.12



* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.

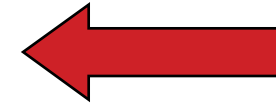


IEBCOPY – Performance Results

Compress PDS Testing results Block size 28332 Format (VB)

Elapsed Time measurements

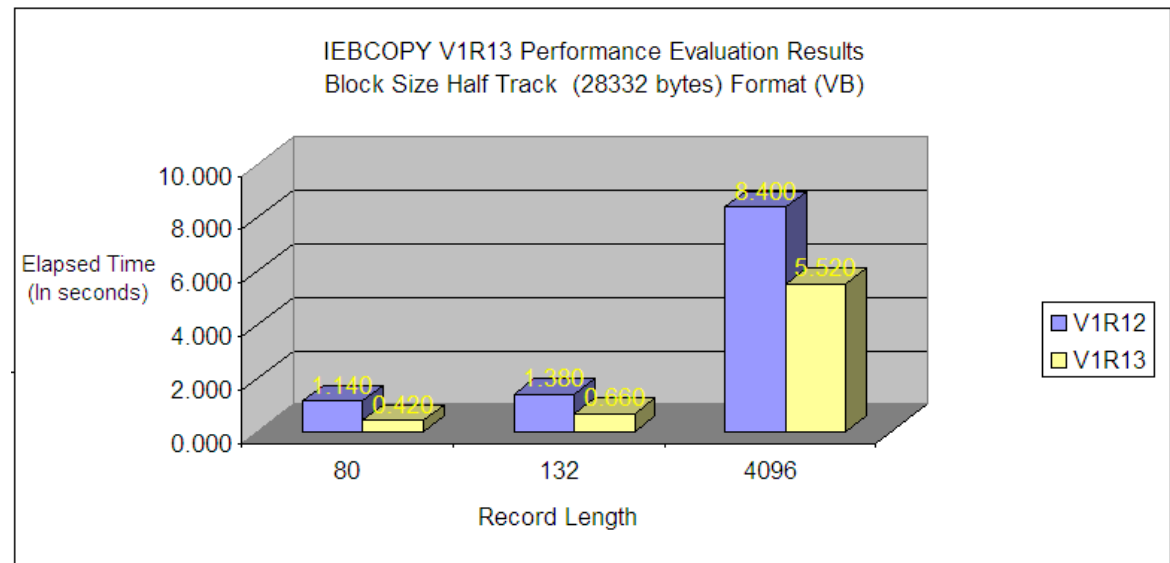
LRECL	V1R12	V1R13	Delta (%)
80	1.140	0.420	-63.16
132	1.380	0.660	-52.17
4096	8.400	5.520	-34.29

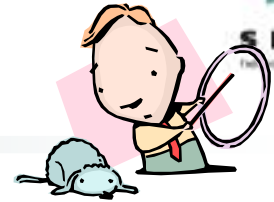


- PDS Compress**

- Deleted 500 members from target PDS prior to compress
 - Prior to delete, target PDS had 1500 members
 - Record Format (VB)**
 - Block size Half Track (28332 bytes)**

★ **~32-63% throughput improvement**





IEBCOPY – Performance Results

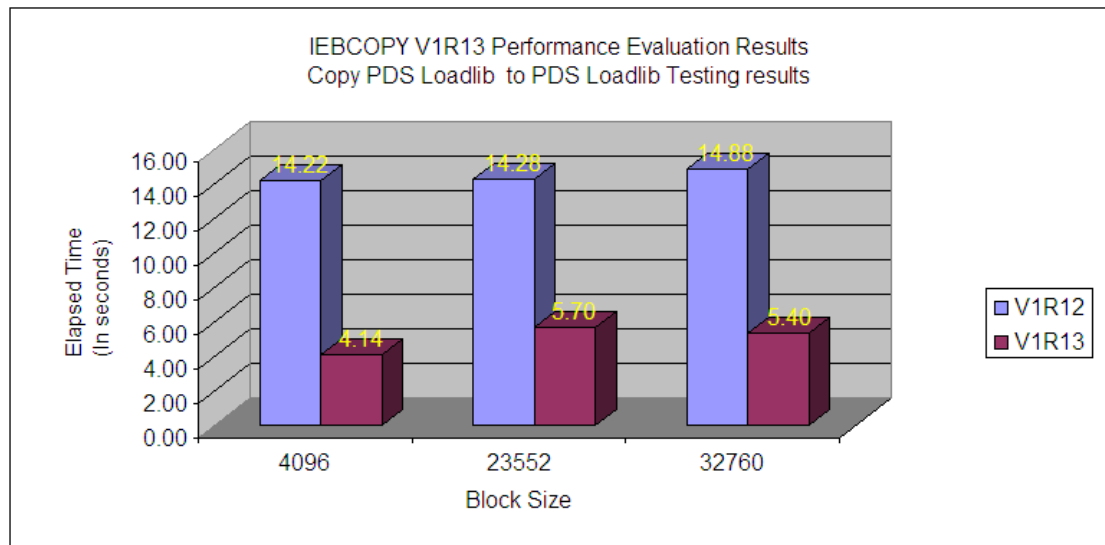
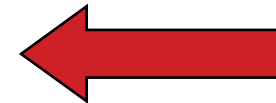
- **PDS Loadlib to PDS Loadlib**
 - Copy 1000 members from PDS source's Loadlib to PDS target's Loadlib
 - **Record Format (U)**
 - **LRECL(0)**

★ **~60-70% throughput improvement**

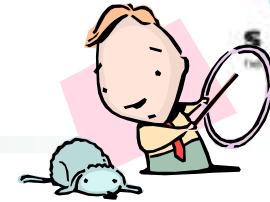
Copy PDS Loadlib to PDS Loadlib Testing results
LRECL=0 Format (U)

Elapsed Time measurements

BLKSIZE	V1R12	V1R13	Delta (%)
4096	14.22	4.14	-70.89
23552	14.28	5.70	-60.08
32760	14.88	5.40	-63.71



* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.

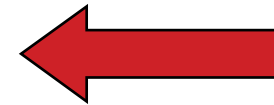


IEBCOPY – Performance Results

Copy PDS Loadlib to SEQ Testing results
LRECL=0 Format (U)

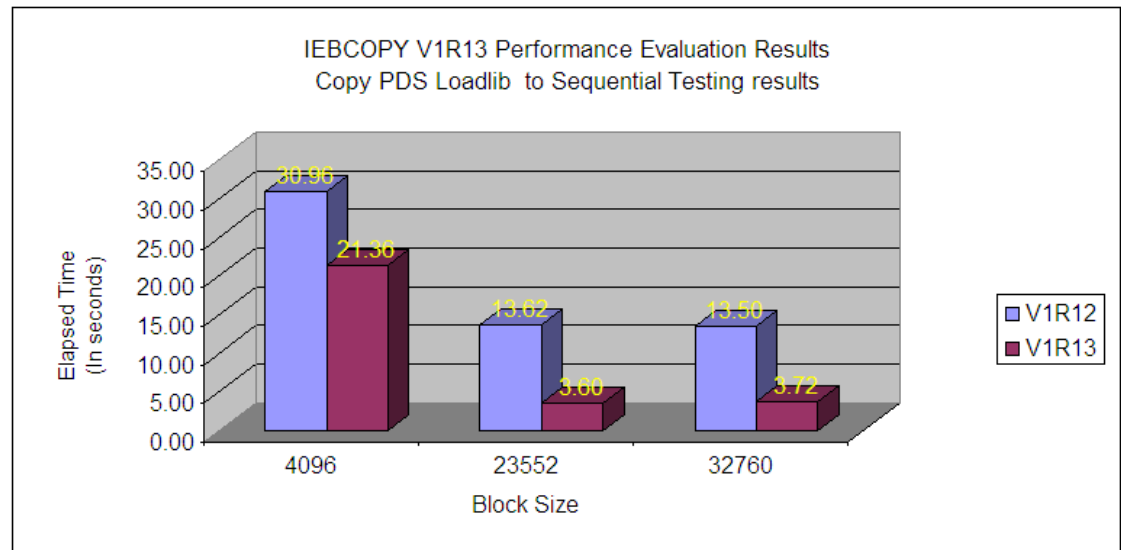
Elapsed Time measurements

BLKSIZE	V1R12	V1R13	Delta (%)
4096	30.96	21.36	-31.01
23552	13.62	3.60	-73.57
32760	13.50	3.72	-72.44



- **PDS Loadlib to SEQ Loadlib**
 - Copy 1000 members from PDS source's Loadlib to SEQ target's Loadlib
 - Record Format (U)
 - LRECL(0)

★ ~31-73% throughput improvement



* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.

IEBCOPY – Performance Results



Loadlib Compress Testing results LRECL=0 Format (U)

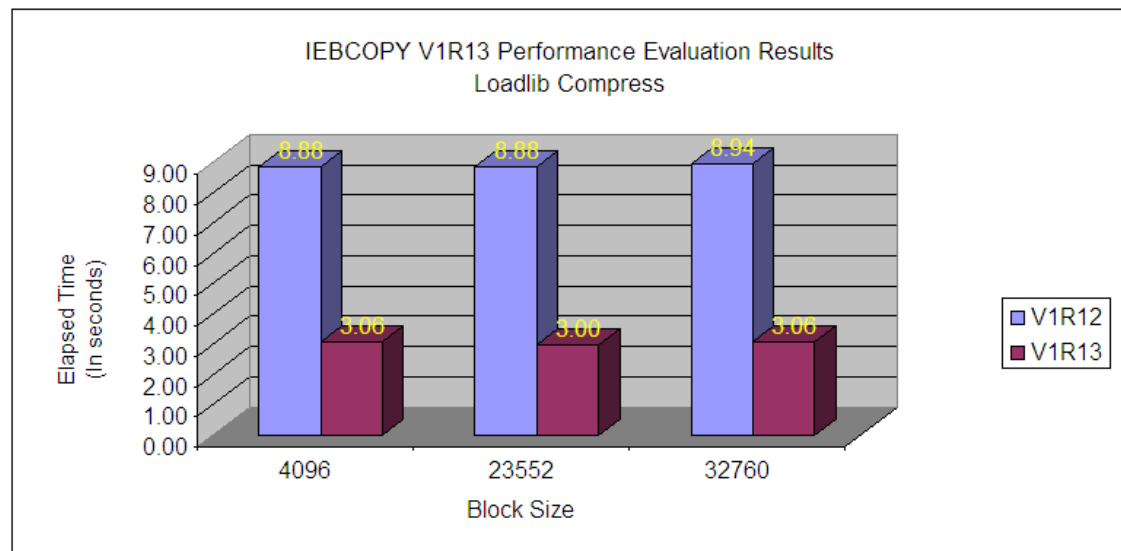
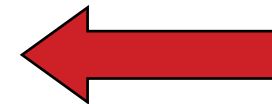
- **Loadlib Compress**

- Deleted 500 members from the target PDS prior to compress
 - Prior to delete, target PDS had 500 members
 - **Record Format (U)**
 - **LRECL(0)**

★ **~65-66% throughput improvement**

Elapsed Time measurements

BLKSIZE	V1R12	V1R13	Delta (%)
4096	8.88	3.06	-65.54
23552	8.88	3.00	-66.22
32760	8.94	3.06	-65.77



* Note: Performance improvements are based on internal IBM laboratory tests. Your results will vary. IEBCOPY improvement will depend on the amount of data being copied, the record format, the record length, and the block size.



IEBCOPY Requirements

Requirement Area	Brief Description
Performance	<ul style="list-style-type: none"> • Improve performance (wall clock time and CPU time) of IEBCOPY in the following cases: <ul style="list-style-type: none"> • Copying a PDS to a PDS, Copying a PDS to a PDSE, Copying a PDSE to a PDSE (not known to present a performance issue in the customer base. If requires changes, it would be in FAMS.) • Compressing a PDS (compressing with a threshold) • Unloading or loading (backing up or restoring) a PDS, unloading or loading (backing up or restoring) a PDSE • IEBCOPY employs inefficient channel programming and 24-bit addressing, which inhibits use of virtual storage and results in poor performance and high CPU usage • Big problems of performance with the IEBCopy coping from pdse to pds • Modify the way IEBCOPY uses PDSE directory when copying a member. Other utilities as linkedit(binder) do it better and then much faster. • Need a faster IEBCOPY. It should run in a half of of the runtime it currently does at the longest. • Explicitly addresses PDSE to PDS performance and not PDS to PDSE performance.
Data Integrity	<ul style="list-style-type: none"> • IEBCOPY needs to be enhanced to make additional data integrity checks to detect corrupted members. This could also be an option for IEBCOPY to set the importance on integrity.
IEBCOPY Replace	<ul style="list-style-type: none"> • IEBCOPY needs to a have a keyword to force an ALIAS to replace a primary member. Products like DB2 can change many modules and they are replacing primary members with aliases, IEBCOPY needs to have an option to the REPLACE keyword to force the member to be replace by an alias. • IEBCOPY processing should allow this copy of a member of a partitioned dataset to itself. • IEBCOPY needs to a have a keyword to force an ALIAS to replace a primary member. • IEBCOPY should show when a member is REPLACED during a copy operation.



IEBCOPY Requirements

Requirement Area	Brief Description
Special Characters / Matching	<ul style="list-style-type: none"> •IEBCOPY's copy member name validation processing should be amended to support special characters in member names. •Member names for IEBCOPY Selection/Exclusion must be fully qualified. We should be able to partially qualify member names (e.g. IST*), instead of entering the complete name. •IEBCOPY should permit specification of members to be copied on the basis of beginning character string. •IEBCOPY to support member level copy for special character member names.
COPYGRP	<ul style="list-style-type: none"> •Customer would like to be able to copy the aliases of load modules during IEBCOPY function. They would like to be able to specify an ALIAS keyword when specifying the main load module name in the IEBCOPY job. That way, they can more easily copy IBM load modules which can have up to a dozen alias names. •When using IEBCOPY to copy PDS LOAD MODULES, any aliases have to be specified explicitly. If they are not they do not get included in the copy. •IEBCOPY COPYMOD function no longer automatically pulls in aliases. This is inefficient as it requires extra legwork to identify module aliases and include them in the Select list. Functionality should be restored to IEBCOPY COPYMOD, or IEBCOPY COPYGRP should allow a copy of PDS to PDS, as this function does automatically pull in the aliases.
Duplicate Names	<ul style="list-style-type: none"> •Allow IEBCOPY to use duplicate member names; Copy the same member multiple times in one operation.



IEBCOPY Requirements

Requirement Area	Brief Description
Messages	<ul style="list-style-type: none">•Customer is requesting that IEBCOPY issues an informational or warning messages to advise that information is being passed to theSYSUT2 dataset from the SYSUT1 dataset.•Amend IEBCOPY to give consistency in COND CODE settings when processing a copy that selects zero members. At present IEBCOPY returns inconsistent COND CODE values when performing a various copies resulting in zero members being selected.•When IEBCOPY is invoked to copy an unloaded PDS containing load modules to a PDSE RECFM=U, the COPY correctly ends in error. Need clearer error messages to document this condition.•IEBCOPY warning messages - Issue W and/or E suffix messages for severity 4 errors. Show return code for each COPY/COPYMOD.•IEBCOPY behavior should be coherent between PDS and PDSE usage for the same request. Specific to messages and return codes.
Space Mgmt	<ul style="list-style-type: none">•Automatically manage and reuse PDS library space.•Enhance FAMS Storage Manager to more effectively manage FAMS storage to avoid ABEND878 abends during IEBCOPY of large PDSE data sets. For very large PDSE data sets like those containing 500K members, these control blocks could end up exhausting subpool 229, key 5 storage thus resulting in an ABEND878 RC10. The actual maximum number of members that can be unloaded successfully will vary depending on the amount of private area virtual storage that is available within the address space and this in turn is affected by the overall system configuration. Reference apar OA36575.•Have IEBCOPY copy what it can before quitting or, have a new parameter such as 'attempt partial copy' or, automatically perform what the user had to do manually; copy a small number of members at a time.

Thank You!!



philipmartin.com

Trademarks and Disclaimers



The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

CICS*	FICON*	Lotus*	System Storage
DB2*	FlashCopy*	MQSeries*	Tivoli*
DFSMS	GDDM*	Multiprise*	TotalStorage*
DFSMSdfp	GDPS*	OMEGAMON*	Virtualization Engine
DFSMSdss	geoManager*	OS/390*	VisualAge*
DFSMSHsm	HiperSockets	Parallel Sysplex*	VM/ESA*
DFSMSmm	HyperSwap	PR/SM	VSE/ESA
DFSORT	IBM*	QMF	VTAM*
DFSMS	IBM logo*	RACF*	WebSphere*
DS4000	ImagePlus*	Rational*	z/Architecture*
DS6000	IMS	RMF	z/OS*
DS8000	Intelligent Miner	System i	z/VM*
Enterprise Storage Server*	Language Environment*	System z	z/VSE
ESCON*		System z9	zEnterprise
			zSeries*
			zSeries Entry License Charge

The following are trademarks or registered trademarks of other companies:

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries or both

Microsoft, Windows, Windows NT and the Windows logo are registered 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 or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

NOTES:

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Users of this document should verify the applicable data for their specific environment.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Information is provided "AS IS" without warranty of any kind.



Trademarks and Disclaimers (continued)



NOTES:

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 are suggested US list prices and are subject to change without notice. Starting price may not include a hard drive, operating system or other features. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM makes no representation or warranty regarding third-party products or services including those designated as ServerProven, ClusterProven or BladeCenter Interoperability Program products. Support for these third-party (non-IBM) products is provided by non-IBM Manufacturers.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.