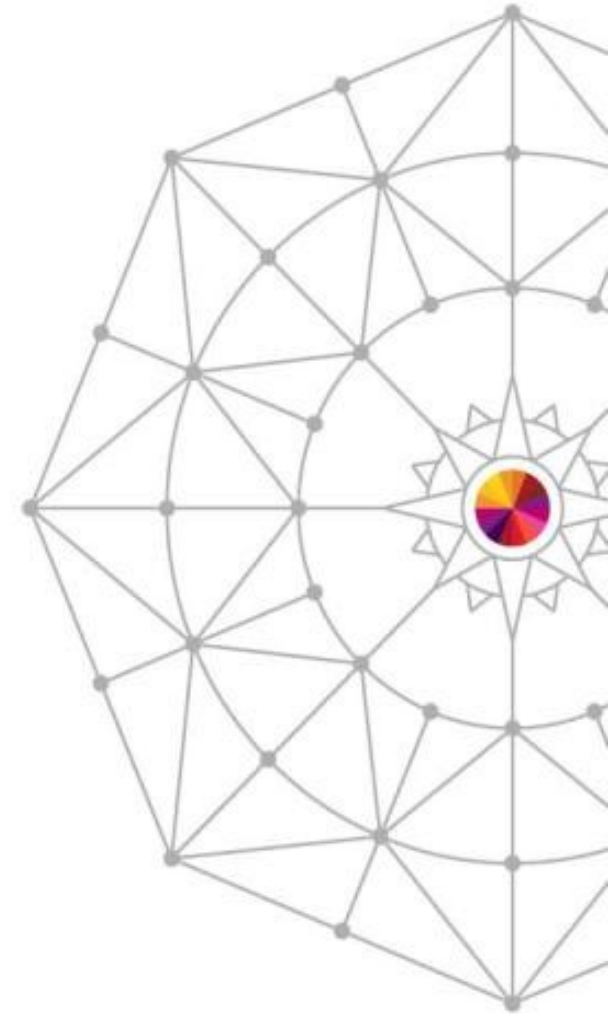




Format-2 Globals Support

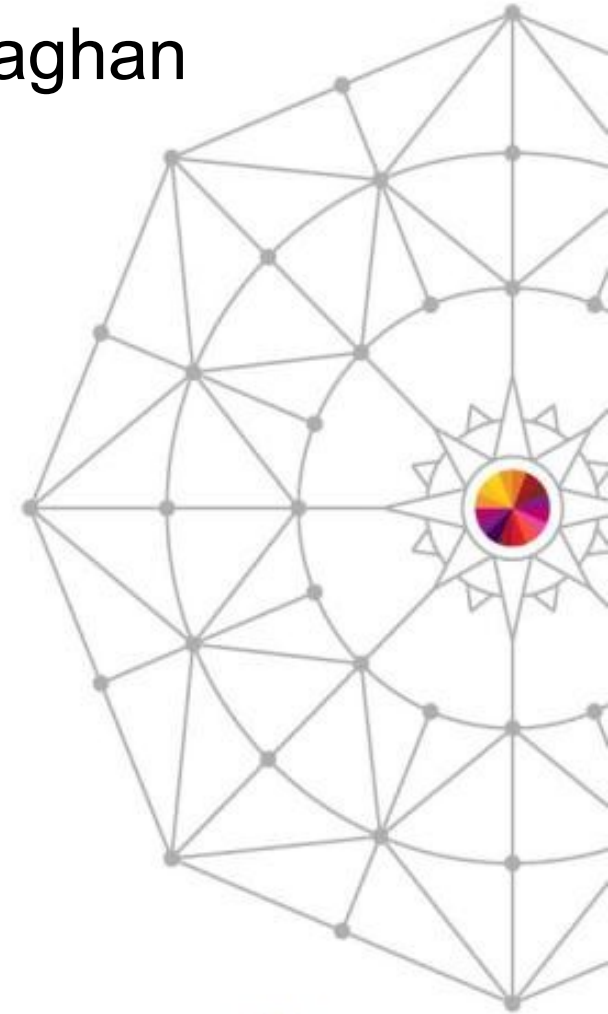
Colette A. Manoni

Education Session



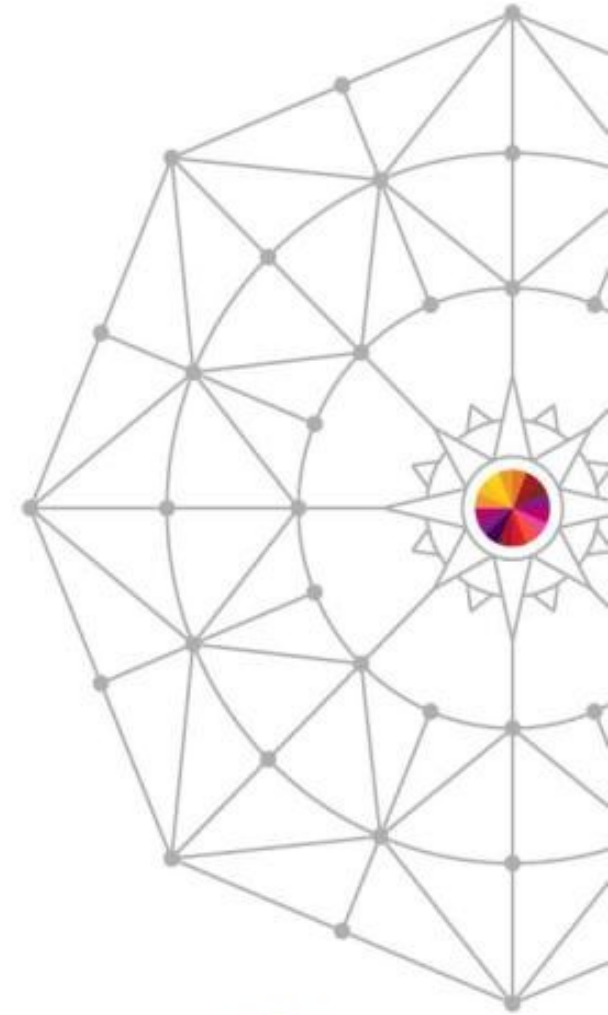
Acknowledgement

Adapted from a presentation by Jason Keenaghan

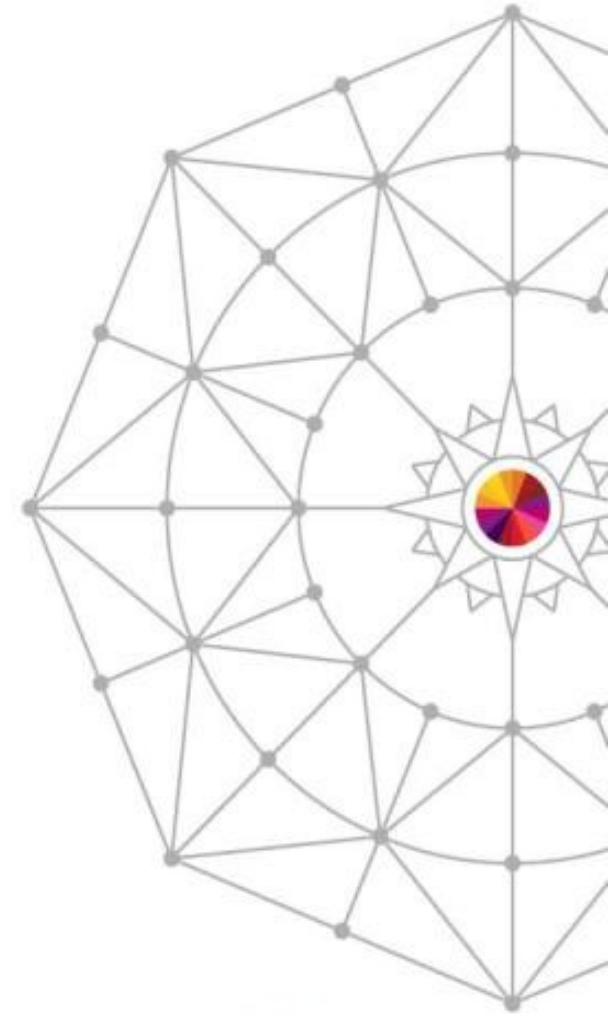


Format-2 Globals Agenda

- Overview
- Control records
- Commands
- APIs
- System Error
- User Controlled
- Migration State



Overview



Overview of Format-2 Globals Support

- New z/TPF globals terminology.
 - Format-1 globals: essentially the same as TPF 4.1 globals support.
 - Format-2 globals: new globals support for z/TPF 1.1 only; completely independent of format-1 globals support.
- Format-2 globals design initiatives.
 - Exploit 64-bit architecture and remove size limitations.
 - Make globals easy to use.
 - Provide robust functionality.
 - Allow globals to be highly extensible.
 - No impact to existing applications.
 - Provide mechanism for accessing existing format-1 globals through use of new format-2 globals APIs.

Overview of Format-2 Globals Support

- Format-2 global records share many of the same attributes as format-1 global records.
 - SSU-unique/shared
 - IS-unique/shared
 - Processor-unique/shared.
 - May be protected.
 - May be keypointable.
 - May be synchronizable.

Overview of Format-2 Globals Support

- Unique features of format-2 global records.
 - Global records only; no global fields.
 - Reside in dynamically allocated storage.
 - Can reside above or below 2-GB bar.
 - Managed dynamically using a set of commands
 - Offline definition process not required
 - No STC, pilot tapes, or online system loader required
 - Can be system controlled or user controlled
 - Extensible via set of user exits
 - Accessed by new APIs.
 - No limit on the size of records
 - Exception is keypointable and synchronizable records
 - Size limited based on the number of IOBs in the system
 - Max number of commit buffers (25% of IOBs)
 - Max size = (# of IOBs ÷ 4) × 3999
 - No limit on the number of globals defined.
 - Independent of physical position in the directory

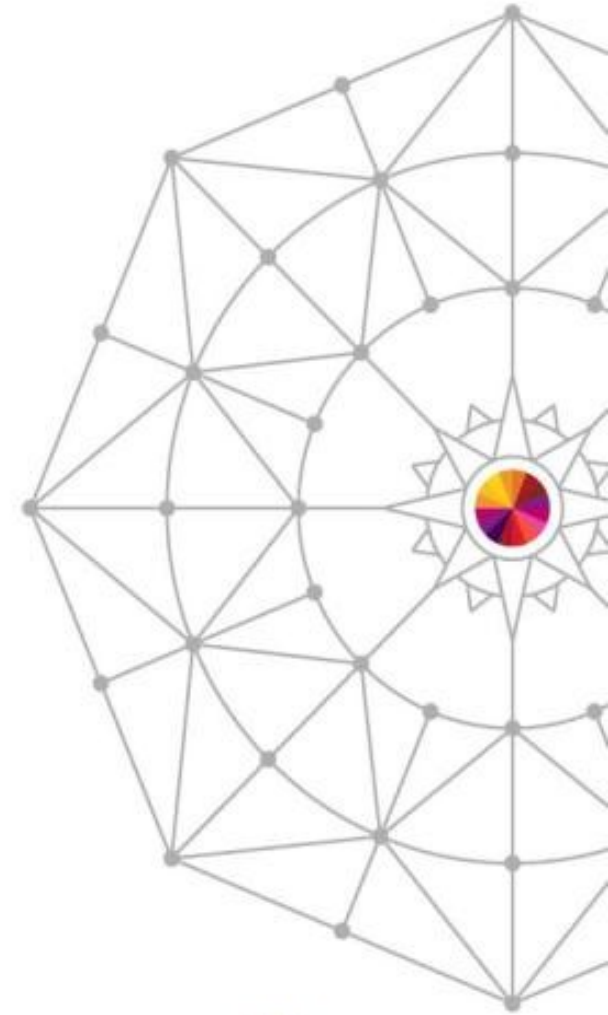
Overview of Format-2 Globals Support

- Format-2 globals commands.
 - Manage format-2 global definitions and data.
 - ZGLBL GLOBAL DEFINE.
 - ZGLBL GLOBAL ALTER.
 - ZGLBL GLOBAL INITIALIZE.
 - ZGLBL GLOBAL DELETE.
 - ZGLBL GLOBAL UNDO INITIALIZE | DELETE.
 - ZGLBL GLOBAL RELEASE.
 - Manage format-2 globals system control information.
 - ZGLBL SET.
 - ZGLBL DISPLAY.
 - Display and alter format-2 global data.
 - ZDGBL.
 - ZAGBL.

Overview of Format-2 Globals Support

- Format-2 globals APIs.
 - Abstract APIs determine behavior based on the defined attributes of the global.
 - C/C++ language APIs modeled after standard file system APIs.
 - Perform various actions on globals.
 - Open a global:
`tpf_glOpen()` and `GLOBLC FUNC=OPEN`.
 - Request updates made to a global to be written to the database:
`tpf_glWrite()` and `GLOBLC FUNC=WRITE`.
 - Change options on how a global was opened:
`tpf_glCntl()` and `GLOBLC FUNC=CNTRL`.
 - Obtain information about an open global:
`tpf_glStat()` and `GLOBLC FUNC=STAT`.
 - Close a global:
`tpf_glClose()` and `GLOBLC FUNC=CLOSE`.

Control Records



Format-2 Globals Control Structures

Fixed file records used to store format-2 globals information on DASD.

- Global control records (GCR).
 - Used to store definitions and other control information.
 - SS-unique; Processor-, SSU-, and IS-shared.
 - #IF2GCR.

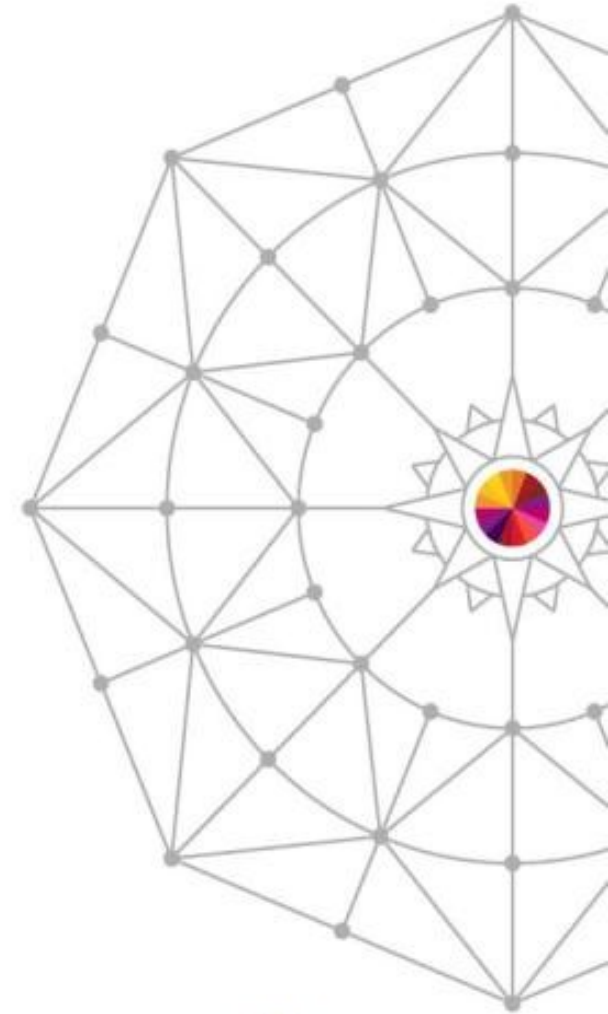
- Global data records (GDR).
 - Used to store the actual data contents of a global.
 - SS-unique.
 - 8 different types depending on uniqueness attributes of data.
 - #IF2GPSI, #IF2GPS, #IF2GPI, #IF2GP, #IF2GSI, #IF2GS, #IF2GI, #IF2G.

- Global availability records (GAR).
 - Byte arrays used to keep track of which data records are in-use.
 - SS-unique; Processor-, SSU-, and IS-shared.
 - 9 different types to correspond to each of the other 9 record types.
 - #IF2AGCR, #IF2APSI, #IF2APS, #IF2API, #IF2AP, #IF2ASI, #IF2AS, #IF2AI, #IF2A.

Determining number of records to allocate

- Global control records (GCR) should be the sum of the following:
 - number of globals defined / 20
 - number of SSU groups defined / 2
 - number of global ordinal records
 - » $((\text{sum_of_initialized_global_record_sizes} / 2,020,000) + \text{count_of_initialized_global_records})$
 - *backup global ordinal records*
 - » $((\text{sum_of_reinitialized_global_record_sizes} / 2,020,000) + \text{count_of_reinitialized_global_records})$
 - number for SSU group list per global = number of globals that include SSU groups
 - number of keypointable globals.
- Global data records (GDRs) should be the sum of the following:
 - $((\text{sum_of_initialized_global_record_sizes_that_use_this_GDR_type} / 4000) +$
 - $(\text{sum_of_reinitialized_global_record_sizes_that_use_this_GDR_type} / 4000))$
- Global availability records (GARs)
 - $((\text{number of corresponding GDR-type records}) / 4000) + 1$

Commands



Defining a New Format-2 Global Record

▪ ZGLBL GLOBAL DEFINE.

- Create a new format-2 global definition for a given SS.
- Global names are 8-characters long.
 - Valid characters: A-Z, a-z, 0-9, and '_'.
 - Padded on the right with blanks.
- Specify the attributes or copy attributes from an existing global (LIKE)
- Attributes:
 - Location: 31-bit or 64-bit.
 - Uniqueness: SSU, IS, processor.
 - IS unique option: memonly
 - Protected or not protected.
 - Keypointable, synchronizable, or neither.
 - Load: restart, cycle-up, on-demand.
 - Control: system or user.

Format-2 Globals SSU Groups

- Allow SSU-unique globals to be shared among a subset of SSUs.
- Data resides in the "owning" SSU.
- Manage SSU groups with commands.
 - ZGLBL SSUGROUP DEFINE.
 - ZGLBL SSUGROUP ALTER.
 - ZGLBL SSUGROUP DELETE.
 - ZGLBL DISPLAY SSUGROUP.
- SSU-unique globals may include as many predefined SSU groups as necessary.
 - ADDGROUP parameter on ZGLBL GLOBAL DEFINE and ZGLBL GLOBAL ALTER.
 - REMGROUP parameter on ZGLBL GLOBAL ALTER.

Format-2 Globals SSU Groups

- Format-2 globals subsystem user group table (SSUGT).
 - Memory-resident control table used to store information about the format-2 globals SSU groups that are defined.
 - There is a unique copy of the SSUGT for every SS in the z/TPF complex.
 - There is one SSUGT entry for every SSU group defined to the SS.
 - Storage for the SSUGT will be obtained dynamically from system heap, as needed.

Format-2 Globals SSU Groups

```
--> WP/ZGLBL SSUGROUP DEFINE group1 OWNER-WP2 ADDSSU-WP1.WP2

CSMP0097I 12.25.50 CPU-B SS-WP   SSU-WP1  IS-01
GLBL0700I 12.25.50 FORMAT-2 GLOBAL SSU GROUP group1 DEFINED SUCCESSFULLY

--> WP/ZGLBL GLOBAL DEFINE _globwp LOC-31 SSU-Y PROC-N IS-N KEY-N SYNC-N
      ADDGROUP-group1

CSMP0097I 12.26.43 CPU-B SS-WP   SSU-WP1  IS-01
GLBL0050I 12.26.43 FORMAT-2 GLOBAL RECORD _globwp DEFINED SUCCESSFULLY

--> WP/ZGLBL GLOBAL INIT _globwp SO-ZERO SIZE-5000 SSUGROUP-group1

CSMP0097I 12.27.24 CPU-B SS-WP   SSU-WP2  IS-01
GLBL0200I 12.27.24 FORMAT-2 GLOBAL RECORD _globwp SUCCESSFULLY
      INITIALIZED
```

Defining a New Format-2 Global Record

```
--> ZGLBL GLOBAL DEFINE _G1ob2 LOG64 SSU-N IS-N PROC-Y KEY-Y  
      SYNC-N PROT-Y LOAD-DEMAND
```

```
CSMP0097I 12.03.28 CPU-B SS-BSS SSU-HPN IS-01  
GLBL0050I 12.03.28 FORMAT-2 GLOBAL RECORD _G1ob2 DEFINED  
      SUCCESSFULLY
```

```
--> ZGLBL GL DEF _myg1ob LIKE-_G1ob2
```

```
CSMP0097I 12.03.28 CPU-B SS-BSS SSU-HPN IS-01  
GLBL0050I 12.03.28 FORMAT-2 GLOBAL RECORD _myg1ob DEFINED  
      SUCCESSFULLY
```

Defining a New Format-2 Global Record

- Upon completion of the ZGLBL GLOBAL DEFINE command:
 - The global definition is viewable by all SSUs within the SS, all processors in the L/C complex, and all I-streams.
 - The global *cannot* be opened by an application, via `tpf_glOpen()`, until it has been initialized with data.

Initializing Data for a Format-2 Global Record

▪ ZGLBL GLOBAL INITIALIZE.

- Initialize the data for a specific format-2 global using global name
- Must initialize the global before it can be used by applications.
- Specify the source from which the global is to be initialized.
 - ASDEFINED: reserves specified amount of storage only.
 - ZERO: reserves specified amount of storage and clears it.
 - GLOBAL: copies data from another format-2 global.
 - INPUTDECK: uses global data input deck created online or offline.
 - Must be in the correct format to be processed by the z/TPF system.
 - User can use any mechanism to create the input deck.
 - Create the input deck on any offline platform and transfer it to the online z/TPF system
 - GLINIT: sample tool to build global data input deck.
 - z/OS: JCL interface to create general data set (GDS), file in HFS, tape, or VRDR.
 - LINUX: command line interface creates file in HFS.
 - Create the input deck on the online z/TPF system itself.

Initializing Data for a Format-2 Global Record

Format of initialization data

Bytes	Description
0-3	Label "F2GL"
4-7	Version number (1)
8-15	Global name
16-23	Size of data to follow
24	Processor ID
25-26	I-stream number
27	Reserved (must be zero)
28-31	SSU name
32-39	SSU group name
40-127	Reserved (must be zeros)
128-*	Data

Initializing Data for a Format-2 Global Record

- Create global data input deck on LINUX, and FTP it to the z/TPF file system.

```
--> ftp 9.117.241.41
--> User:anonymous
--> Password:*****
--> cd /temp/globaldata/
--> put _myglob.data
```

- Define a temporary data definition name (DDNAME) on z/TPF, and initialize the specified copy/copies of the format-2 global record

```
--> ZDSMG DEF MYGLOBDDNAME HFS-'/temp/globaldata/_myglob.data'
DSMG0001I 07.14.09 DDNAME MYGLOBDDNAME      DEFINED
--> ZGLBL GL INITIALIZE _myglob SOURCE-INPUT INPUT-MYGLOBDDNAME PROC-ALL
GLBL0200I 07.19.14 FORMAT-2 GLOBAL RECORD _myglob SUCCESSFULLY
INITIALIZED
```

Re-initializing Data for a Format-2 Global Record

- ZGLBL GLOBAL INITIALIZE.
- Processing is nearly identical to the first initialization.
- If the global is already initialized, a backup copy of the global must be saved in the database.
 - A timestamp is kept indicating when the most recent backup was created.
 - Only one backup copy of the data can exist at a time. If a backup already exists, it will be overwritten.

User: WP1/ZGLBL GL INIT _GLBTEST SOURCE-INPUTDECK INPUT-GLOBDATA

System: GLBL0010I ENTER - ZGLBL CONTINUE - OR - ZGLBL CANCEL

User: ZGLBL CONTINUE

System: GLBL0015I CONTINUING WITH - ZGLBL GLOBAL INITIALIZE
GLBL0200I FORMAT-2 GLOBAL RECORD _GLBTEST INITIALIZED SUCCESSFULLY

Deleting Data for a Format-2 Global Record

ZGLBL GLOBAL DELETE *global*.

- When expiration time limit is reached
 - All fixed-file records reserved are marked as available for re-use
 - All main storage is released when it is no longer being referenced
 - Controlled using ECB activation numbers

User: ZGLBL GLOBAL DELETE _GLBTEST

System: GLBL0010I ENTER - ZGLBL CONTINUE - OR - ZGLBL CANCEL

User: ZGLBL CONTINUE

System: GLBL0015I CONTINUING WITH - ZGLBL GLOBAL DELETE
GLBL0150I FORMAT-2 GLOBAL _GLBTEST DELETED SUCCESSFULLY

Undoing Initialize or Delete for a Format-2 Global Record

ZGLBL GLOBAL UNDO INITIALIZE or DELETE.

- Before the expiration time limit is reached can “undo” an initialize or delete
 - Initialize will be restored using the back-up copy
 - If there is no back-up copy, global is set to uninitialized

User: ZGLBL GL DELETE _GLBTEST

System: GLBL0010I ENTER - ZGLBL CONTINUE - OR - ZGLBL CANCEL

User: ZGLBL CONTINUE

System: GLBL0015I CONTINUING WITH - ZGLBL GLOBAL DELETE
GLBL0150I FORMAT-2 GLOBAL _GLBTEST DELETED SUCCESSFULLY

User: ZGLBL GL UNDO DELETE _GLBTEST

System: GLBL0600I DELETION OF FORMAT-2 GLOBAL _GLBTEST WAS UNDONE SUCCESSFULLY

Releasing a Format-2 Global Record

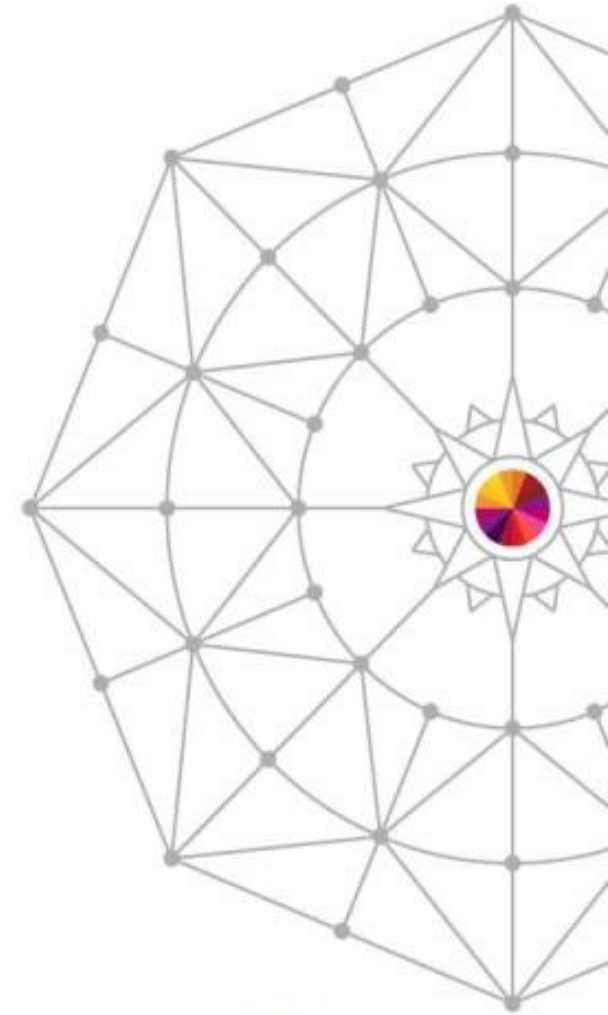
ZGLBL GLOBAL RELEASE *global name*.

- Force the release of backup storage
 - Main storage not released until there are no more ECBs referencing
- Fixed file records are made available for reuse immediately

User: WP/ZGLBL RELEASE _GLBTEST

System: GLBL0400I FORMAT-2 GLOBAL RECORD _GLBTEST BACKUP STORAGE RELEASED

APIs



Opening a Format-2 Global Record

- `tpf_glOpen(char *globalname, enum t_glopt options, void **globaladdr)`.
 - Returns main storage address of requested global for the current SSU and IS.
 - Specify options for opening a global.
 - `TPF_GLRD`: open for read-only access.
 - `TPF_GLRDWR`: open for read/write access (ie, lock global for exclusive use).
 - `TPF_GLRDFST`: open for read-only access using fast path.
 - Creates a global descriptor (non fast path only).
 - Keeps track of all globals currently open for an ECB.
 - Used as input for subsequent `tpf_gl*` API calls.
- Assembler equivalent: `GLOBLC FUNC=OPEN`.
 - `OPT=RDONLY`.
 - `OPT=RDWRITE`.
 - `OPT=RDFAST`.

Opening a Format-2 Global Record

- Processing of `tpf_glOpen()`.
 - `tpf_glOpen()` expands to a `#define` macro to allow for optimal performance when `TPF_GLRDFST` option is specified.
 - `__tpf_glOpen()` is called from within this macro to perform global open if `TPF_GLRD` or `TPF_GLRDWR` options are specified.
 - Function is located in `cglapi.cpp` in CTAD shared object.
- Processing of `GLOBLC FUNC=OPEN`.
 - `GLOBLC FUNC=OPEN` will branch to a stub routine in `FINIS` to allow for optimal performance when `OPT=RDFAST` option is specified.
 - `GLOBLC FUNC=OPEN` will perform a `CALLC` to `__tpf_glOpen()` to perform global open if `OPT=RDONLY` or `OPT=RDWRITE` options are specified.

Opening a Format-2 Global Record

- If the global is opened for read/write access, it will be locked for exclusive use by this ECB.
 - Synchronizable: perform equivalent of SYNC LOCK operation to lock the global on DASD (first GDR only) and refresh main storage copy.
 - The DECB used to lock the GDR with a FIWHC will be saved and referenced later when unlocking the global.
 - NOTE: format-2 global APIs *do not* participate in commit scopes.
 - Non-synchronizable: perform CORHC on global's main storage address.
- If this is a non-fast path open, a format-2 global descriptor will be created and chained to the ECB.
 - Global descriptor keeps track of the options with which this global was opened.
 - Keep track of all globals that an ECB currently has open.

Writing an update to a Format-2 Global Record

- Two formats:
 - `tpf_glWrite(int globaldesc, enum t_glopt options);`
 - `tpf_glWrite(int globaldesc, enum t_glopt options, long offset, long length);`
- Both require a global descriptor which was returned on the open
- Specify options
 - `TPF_GLALL`: updates the entire global record
 - `TPF_GLUPART`: specifies part of the record is to be updated.
 - Must supply offset and length.
 - Only applies to synchronizable
- Global is updated based on its attributes
 - system will request a keypointing if its keypointable
 - synchronizable will be filed but other processors not notified until global is unlocked or closed
- Assembler equivalent: `GLOBLC WRITE.`
 - `OFFSET= LENGTH=`

Change control options for Format-2 Global Record

- tpf_glCntl (int globaldesc, enum t_glopt options);
- Requires a global descriptor which was returned on the open
- Specify options
 - TPF_GLRDWR: change to read and write
 - Obtains exclusive lock if not already held
 - TPF_GLUNLK: unlock
 - If synchronized, notifications are sent to other processors
 - TPF_GLUNLKWT: unlock and wait
 - If synchronized, notifications are sent to other processors
 - Waits to return until all processors acknowledge
- Does not change storage protection
- Assembler equivalent: GLOBLC FUNC=CNTRL
 - OPT=RDWRITE
 - OPT=UNLOCK
 - OPT=UNLOCKWT

Change control options for Format-2 Global Record

- `tpf_gIStat(int globaldesc, struct iglst *stainfo);`
- Require a global descriptor which was returned on the open
- Returns a structure containing
 - Main storage address of the global
 - Size in bytes
 - SSU-unique: true or false
 - IS-unique: true or false
 - Processor unique: true or false
 - Keypointable: true or false
 - Synchronizable: true or false
 - Protected: true or false
 - Read only: true or false
- Assembler equivalent: `GLOBLC FUNC=STAT`
 `-ADDR= R14 or reg`

Closing a Format-2 Global Record with Updates

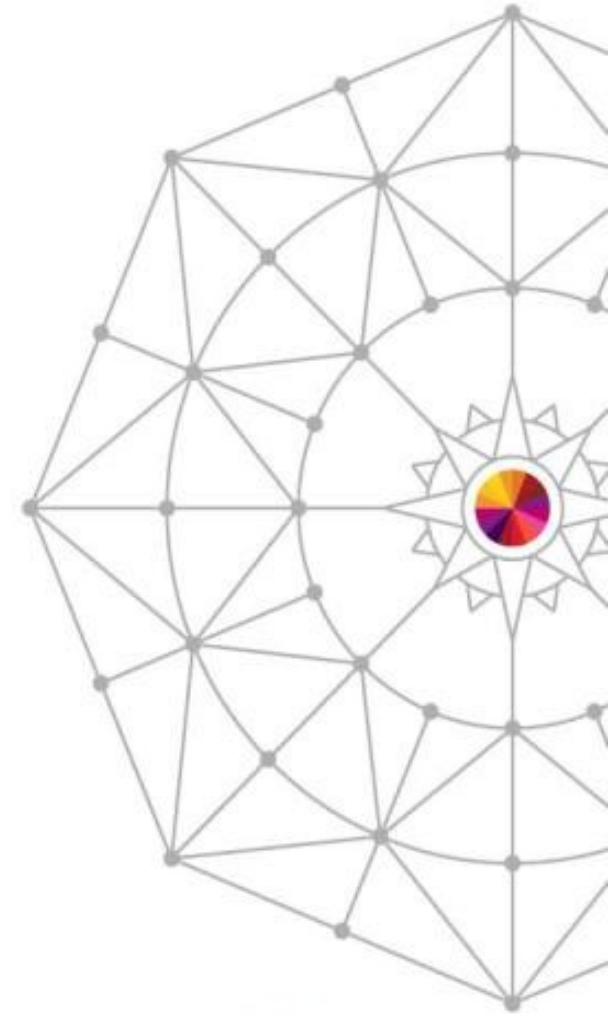
- `tpf_glClose` (int globaldesc, enum t_glopt options, long offset, long length);
 - Specify options for closing a previously opened global.
 - `TPF_GLUPD`: file out updates made to the global (ie, keypoint or synchronize).
 - `TPF_GLUPDWT`: file out updates made to the global and wait for completion (only for synchronizable).
 - `TPF_GLNOU`: close global without updating it.
 - `TPF_GLPART`: only update the specified portion of global (using offset and length parameters).
 - Remove format-2 global descriptor from ECB.
- Assembler equivalent: `GLOBLC FUNC=CLOSE.`
 - `OPT=UPDATE.`
 - `OPT=UPDATEWT.`
 - `OPT=NOUPDATE.`

Closing a Format-2 Global Record with Updates

- Processing of `tpf_glClose()` when an update is requested.
 - The global descriptor that is specified on input is used to determine which format-2 global is being closed.
 - Validation is performed on the global descriptor (eg, has the global been deleted; has the global been reinitialized; etc.).
 - Based on the attributes of the global it will be updated on DASD.
 - Keypointable.
 - Synchronizable.

- Keypointing format-2 globals.
 - 1052 state: file the updated global to the database before returning to the caller.
 - Above 1052 state: turn on the keypoint requested indicator.

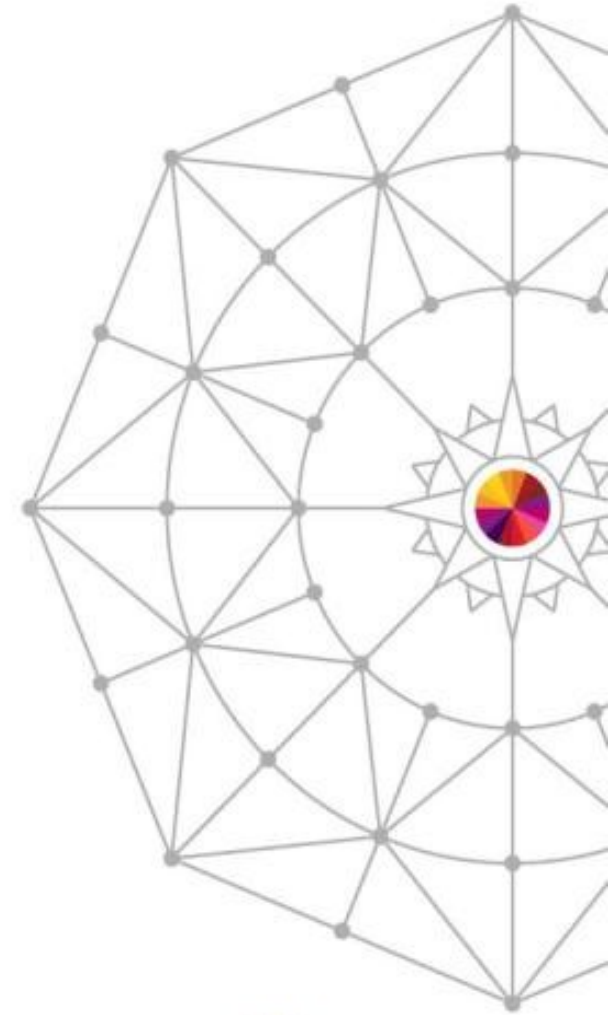
System Error Processing



System Error Processing

- Format-2 global names as dump keywords.
 - 2 ways to create a dump keyword that references a format-2 global.
 - ZIDOT CREATE command.
 - IDATG macro.
 - Use the ZIDOT INCLUDE command to specify that this global should be included for specific system errors. The copy of the global for the failing SSU and IS is dumped.
 - "IF2GLOPN": pre-defined dump keyword that will dump all format-2 globals currently opened by the failing ECB.
- Specifying format-2 globals at system error time.
 - LISTC macro.
 - Specify name of a format-2 global and maximum length to dump.
 - Can be used for SERRC and SNAPC dumps.
 - serrc_list and snapc_list structures updated for C/C++.
 - SERRC_F2GLOBAL and SNAPC_F2GLOBAL indicators tell the z/TPF system that the "tag" field is a format-2 global name.
 - "IF2GLOPN" can be used as the global name, as noted above.

User-Controlled Format-2 Global Records



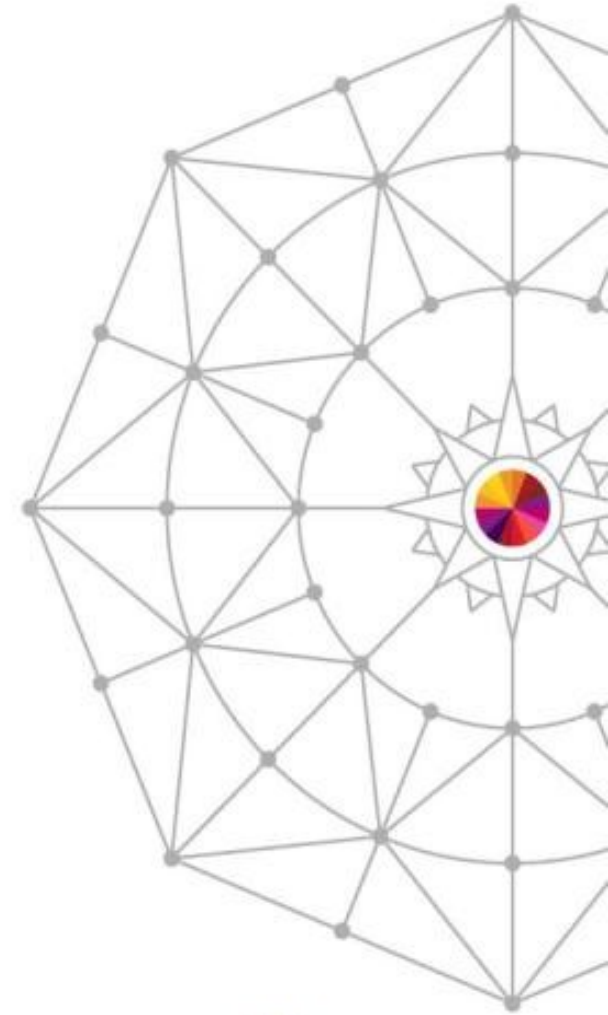
User-Controlled Format-2 Global Records

- ZGLBL GLOBAL DEFINE with CONTROL-USER.
- When the requirements for a particular global record do not fit into the definition of a system-controlled global record, it can be defined as user-controlled.
- Examples:
 - Not stored on DASD in #IF2G-type records.
 - Special loading, keypointing, synchronizing, or initialization requirements.
- Provides the ability to use format-2 globals as a pointer to dynamically built user tables.
 - Applications can use format-2 globals APIs with global record names to access user tables.
 - Tables can be dynamically replaced in NORM state and have the global address updated to point at the new table.
 - GLBLUC macro.
 - tpf_glbluc() C/C++ API.

User-Controlled Format-2 Global Records

- Easy customization for user-controlled format-2 globals with various user exits.
 - ZGLBL GLOBAL DEFINE.
 - ZGLBL GLOBAL ALTER.
 - ZGLBL GLOBAL DELETE.
 - ZGLBL GLOBAL INITIALIZE.
 - ZGLBL GLOBAL UNDO.
 - ZAGBL.
 - Format-2 global load.
 - Keypoint requests.
 - Synchronization requests.

Format-1 Globals in Migration State



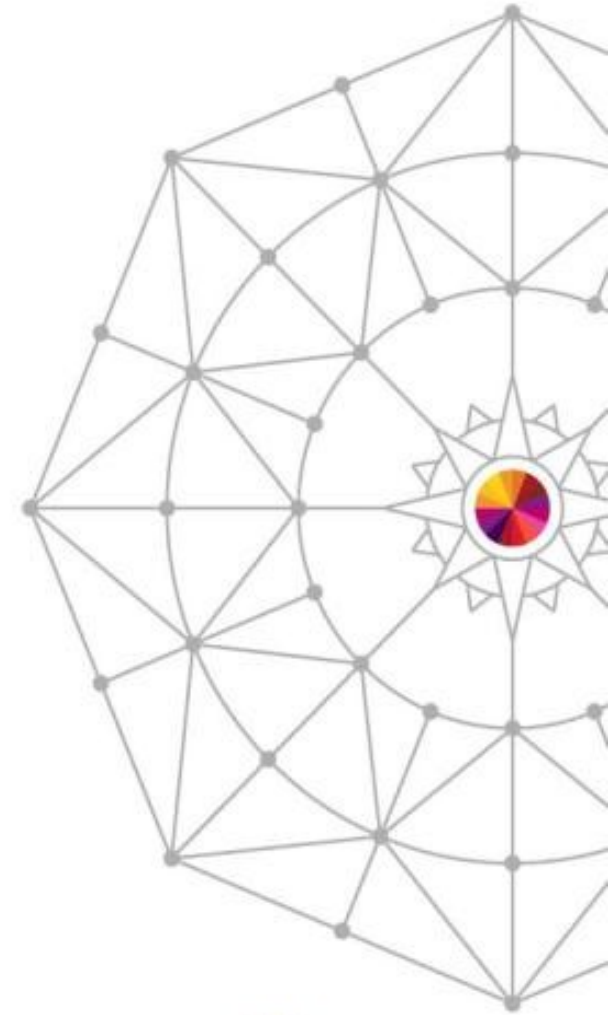
Format-1 Globals in Migration State

- ZGLBL GLOBAL DEFINE with CONTROL-FORMAT1.
- Allows user to define a format-2 global record that points at existing format-1 global data.
- Applications can be migrated one at a time to use new format-2 globals APIs to access format-1 globals data.
 - All format-2 globals APIs can be used on format-1 globals in migration state.
 - Limitations on the format-2 globals commands that can be used (eg, cannot display or alter file copy of data).
 - UGLM: migration user exit.
- ZGLBL GLOBAL MIGRATE.
 - Migrate data from format-1 to format-2.
 - Only perform once all applications have been updated to use format-2 globals APIs.
 - Global will become a system-controlled format-2 global, and will no longer reference the format-1 global data.

Format-1 Globals in Migration State

- During processing of the tpf_gl-type APIs, when the z/TPF system detects that the requested global to be acted upon is a format-1 global, the global migration user exit (UGLM) is entered.
 - UGLM must be updated to perform various actions on each format-1 global that is to be used in migration state.
 - "Load" (ie, return main storage address).
 - Keypoint the global.
 - Synchronize the global.
 - Once the global's main storage address has been stored in the FGAT, subsequent application requests to return the main storage address (eg, tpf_glOpen) will not require calls to UGLM.

Thank you !



Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.
- *(Include any special attribution statements as required – see Trademark guidelines on <https://w3-03.ibm.com/chq/legal/lis.nsf/lawdoc/5A84050DEC58FE31852576850074BB32?OpenDocument#Developing%20the%20Special%20Non-IBM%20Tr>)*

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.
- 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.
- This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.