IBM Software Group

TPF Users Group Spring 2005

Update on Single Source APARs & Other Application Migration Tips

Name : Sue Pavlakis
Venue : Languages / Applications Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Single Source Concept

- There are some changes which must be made to application programs in order to migrate from TPF 4.1 to z/TPF.
- Single Source support introduces changes into TPF 4.1 that are required for z/TPF.
  - Allows the changes to be made to applications now while they are running in TPF 4.1.
  - Don't need two copies of application source.
    - If you start to exploit z/Architecture you will need to address maintaining different code in TPF 4.1.
  - Enables the same application source to be built for TPF 4.1 or z/TPF without any conditional code.

Single Source Steps

1. **Apply Single Source APARs to your TPF 4.1 system.**
   - Single Source APARs do not require any changes to applications - they do not break any existing interfaces.
2. **Assess applications to identify where changes need to be made.**
   - For most cases scans will identify where changes need to be made.
3. **Change applications to make them compatible with z/TPF.**
   - The changes do not have to be made at one time.
   - Can start today to facilitate your z/TPF migration.

Single Source APARs Covered at Fall 2004 TPFUG

- PJ29218 (TPF4.1 PUT18) - Adds support for the LBASEC, LREGSC, LOCKC and UNLKC macros; updates the BEGIN macro for new parameters.
  - Correcting APAR: PJ30136 (TPF4.1 PUT19).
- PJ29575 (TPF4.1 PUT18) - Adds PTR32 type definitions.
- PJ29576 (TPF4.1 PUT19) - Provides decimal data type support.
- PJ29593 (TPF4.1 PUT19) - Adds wrappers for header file name change & tpf directory.
- PJ29630 (TPF4.1 PUT19) - Adds time_t32 & size_t32 definitions (for structures out on file).
- PJ29640 (TPF4.1 PUT19) - Adds support for the PRLGC, EPLGC, CSTKC & PBASC macros.
- PJ29691 (TPF4.1 PUT19) - Adds support to the PNAMC, DEFBC, ENTRC & ENTNC macros.

Single Source APARs Covered at Fall 2004 TPFUG (continued)

- PJ29692 (TPF4.1 PUT19) - Adds support for the CPROC & CALLC macros.
- PJ29849 (TPF4.1 PUT19) - Adds support for the tpf_fp_htob & tpf_fp_btoh functions for floating point migration.
- PJ29937 (TPF4.1 PUT19) - Moves the gettimeofday function declaration from sysgtime.h to sys/time.h.
- PJ29948 (TPF4.1 PUT19) - Allows you to use the TLDR program to identify relocatable address constants (ADCONS).
- PJ29957 (TPF4.1 PUT19) - Adds support for the time zone (TZ) environment variable.
- PJ29969 (TPF4.1 PUT19) - Adds support for the SREGSC macro & changes to the PNAMC macro.
- PJ29980 (TPF4.1 PUT19) - Adds support for the tpf_fp_hton, tpf_fp_ntoh, tpf_fp_bton & tpf_fp_ntob functions for floating point migration.

References

- ***C/C++ Single Source APARs*** given by Sarat Vemuri in the Fall 2004 Languages Subcommittee.
- ***Coding Today For z/TPF Tomorrow, C/C++ Coding Changes for Ease of Migration to z/TPF*** given by Chris Filachek in the Fall 2004 Languages Subcommittee.
- ***What You Can Do Today To Get Ready For Tomorrow, Assembler Programs*** given by Sue Zee Wolfsie in the Fall 2004 Applications Subcommittee.

Additional Single Source APARs

- PJ29436 (TPF4.1 PUT19) - Changes to the activate_on_receipt (AOR) socket API.
- PJ29974 (TPF4.1 PUT19) - Adds an error trapping method for math functions.
- PJ29984 (TPF4.1 PUT19) - Moves definitions & declarations relating to BSD select to sys/time.h.
- PJ30089 (TPF4.1 PUT19) - Adds an error parameter to CSONC.
- PJ30189 (TPF4.1 PUT19) - Allows TPFAR applications to use sqlint32 data type.
- PQ91889 (TPFDF1.1.3 PUT20) - Add wrappers for header file name changes for TPFDF.

PJ29436 - Changes to the activate_on_receipt (AOR) Socket API.

- When an application is activated in a new ECB as a result of:
  - activate_on_receipt
  - activate_on_receipt_with_length
  - activate_on_receipt_of_TCP_message
  - activate_on_receipt_of_TCP_message2

address of data is saved in starting at EBW024 in addition to being saved starting at EBW012.

- Why?
  - An 8 byte address is required in z/TPF and EBW016-EBW019 was not available.
- What to look for:
  - Users of these AOR APIs and users of EBW012.
    - Update to use EBW024.

PJ29974 - Adds an Error Trapping Method for math Functions

- Provides new macros:
  - tpf_chk_log_dbl()
  - tpf_chk_log10_dbl()

which return nonzero values if the parameter to the function is outside the domain of the function and return 0 otherwise.

- Why?
  - The values returned by the log() and log10() functions under TPF 4.1 & z/TPF are different.
  - A new method for determining if the parameter is outside the domain is needed to give consistent results in both TPF 4.1 & z/TPF.
- What to look for:
  - Callers of log() and log10().

PJ29984 - Moves Definitions & Declarations Relating to BSD select to sys/time.h

- These BSD select related definitions & declarations have been moved from sysgtime.h to time.h:
  - FD_SET
  - FD_CLR
  - FD_ISSET
  - FD_COPY
  - FD_ZERO
  - tpf_select_bsd
  - struct fd_set
  - long fd_mask
  - NBBY
  - NFDBITS
  - howmany

PJ29984 - Moves Definitions & Declarations Relating to BSD select to sys/time.h (continued)

- Why?
  - These definitions & declarations are standard & have been moved to the standard header file (sys/time.h) in z/TPF.
- What to look for:
  - Scan for include of <sysgtime.h>
    - Replace it with include of sys/time.h.

PJ30089 - Adds an Error Parameter to the CSONC Macro

- An ERROR= parameter has been added to the CSONC macro.
- Why?
  - When there was an error processing CSONC, R14 was negative. When CSONC is successful, R14 contains the module (MM) & cylinder (CC) numbers.
  - With 40,000 modules supported in z/TPF a successful CSONC can appear to have a negative R14 value.
  - In z/TPF, CSONC was changed to return a -1 in R14 when an error occurs and the ERROR= parameter was added.
- What to look for:
  - Checking for a negative value in R14 following a CSONC call.
    - Update to use ERROR= parameter instead.

PJ30189 - Allows TPFAR Applications to use sqlint32 data type

- The DB2 precompiler defines a host variable with a data type of sqlint32 as a 4-byte integer for DB2. The sqlint32 data type is now allowed on TPF.
- Why?
  - Not all versions of the DB2 precompiler handle the long data type the same way (8 bytes versus 4 bytes).
  - The sqlint32 data type is handled the same way (4 bytes) across all versions & all platforms (z/OS or Linux) of the DB2 precompiler.
- What to look for:
  - Host variables defined as long data type.
    - Convert to sqlint32 data type.
- Reference:
  - TPF/AR Update given by John Tarby in the Spring 2005 Database Subcommittee.

PQ91889 - Adds Wrappers for Header File Name Changes for TPFDF

- Creates new external TPFDF header files using "_" instead of "$".
  - All these header files still exist with old names on TPFDF1.1.3 but not on z/TPFDF.
- Why?
  - Removed "$" from file names to make name compatible with GCC compiler and Linux rules.

PQ91889 - Adds Wrappers for Header File Name Changes for TPFDF (continued)

- What to look for:
  - #include statements for these headers: c$cdfapi, c$cdfeq, c$cdferr, c$cdflnk, c$cdfmac, c$sw00sr, c$sw01sr, c$sw02sr
    - change "$" to "_".
    - The convert_hdr.sh & convert_src.sh tools can be used to make this change (see TPF web site for tool information).
    - These tools also change #include "..." to #include <...>.
      - You can modify the tools to avoid this change if needed.
- Reference:
  - TPFDF Status/Update given by Kevin Jones in the Spring 2005 Database / TPFDF Subcommittee.

Enumerated Data Types

- Here is what we said at the Fall 2004 TPFUG:
  - TPF4.1: Enumerations are 1, 2, or 4 bytes
    - Size depends on values of the enumerations
  - z/TPF: Enumerations are always 4 bytes
  - For structures mapped by assembler DSECTs or written to file, change enumerations as follows
    - Change 1 byte enumerations to unsigned char
    - Change 2 byte enumerations to unsigned short
    - Leave 4 byte enumerations alone

Enumerated Data Types (continued)

- Updated to say:
  - z/TPF:
    - Enumerations are 1, 2 or 4 bytes if **-fshort-enums** compiler option is used (specifies smallest integral type should be used).
    - Enumerations are 4 bytes if **-fshort-enums** compiler option is not used.
  - For structures mapped by assembler DSECTs or written to file, make either of the following changes:
    1. Use the **-fshort-enums** compiler option to ensure the size of the enumerations are the same between TPF 4.1 and z/TPF.
    2. Update application source code:
       - Change 1 byte enumerations to unsigned char
       - Change 2 byte enumerations to unsigned short
       - Leave 4 byte enumerations alone

Trademarks

IBM, DB2, z/Architecture, and z/OS are trademarks of International Business Machines Corporation in the United States, 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.

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.

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.