z/Transaction Processing Facility Enterprise Edition 1.1.0 (z/TPF)

Features, Functions, Benefits
A Business Perspective

Penny Cresswell

IBM Certified Consulting I/T Specialist

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Customer Business Drivers

- How does z/TPF enable the customer to:
  - reduce costs
  - effectively manage a skilled resource base
  - improve time to market
  - develop competitive products and solutions
  - gain market share
  - protect current investment and position for business growth
  - ease the migration effort
- What can the customer do with z/TPF, that could not be done before?

Mapping Business Drivers to Product Capabilities

- **Reduce Costs**
  - Improve performance (people, software, hardware, support, etc.)
  - Reduce complexity (people, software, hardware, support, etc.)
  - Maximize resources (people, software, hardware, support, etc.)
- **Effectively Manage a Skilled Resource Base**
  - Reduce Cost +
  - Facilitate synergies
  - Facilitate hiring individuals with new technology skills
  - Facilitate hiring at competitive rates
  - Facilitate integrating valued content knowledge with the next generation of programming talent
  - Facilitate education in new skills and technologies

Mapping Business Drivers to Product Capabilities

- **Improve Time to Market**
  - <u>Reduce Cost + Effectively Manage a Skilled Resource Base +</u>
  - Facilitate fast, easy, efficient software development
    - Ease programming constraints
    - Standard language support
    - Flexible and easy to use APIs
  - Facilitate faster, easier problem determination and resolution
  - Provide best-practice development, testing and debugging tools
  - Standardize product development environment
  - Provide for compliance and support for applicable industry standards (e.g. POSIX, W3C…) and interface standards (e.g. SOAP, XML, MQ…)
  - Provide the ability to port existing code
  - Provide the ability to access and/or re-use resources within an Enterprise

Mapping Business Drivers to Product Capabilities

- **Develop Competitive Products and Solutions**
  - Reduce Cost + Effectively Manage a Skilled Resource Base + Improve Time to Market +
  - Provide Speed, Availability, Reliability, and Scalability,
  - Frees up development sources for that which is truly unique and provides added value to the business
- **Gain Market Share**
  - Reduce Cost + Effectively Manage a Skilled Resource Base + Improve Time to Market + Competitive Products and Solutions +
  - Enable the customer to market unique functions and features that benefit the business while at the same time providing assurance that their product has a solid foundation, utilizes new technology, and is positioned for growth.

Mapping Business Drivers to Product Capabilities

- **Protect Investment**
  - Reduce Cost + Effectively Manage a Skilled Resource Base + Improve Time to Market + Competitive Products and Solutions + Gain Market Share +
  - Invest in technology that protects existing environment, utilizes new technology, and positions for the technologies of tomorrow
  - Reduce Risk
- **Ease the Migration Effort**
  - Provide conversion tools
  - Provide migration tools
  - Provide coexistence capabilities
  - Provide user modifications and/or user exits in the base product
  - Provide pre, during, and post education services and consultation

Feature – z/Architecture

- **z900, z990, z800, z890 machines have 2 architecture modes:**
  - ESA/390 mode – TPF 4.1 runs in this mode which is limited to 2 GB of storage (memory)
  - z/Architecture mode – z/TPF runs in this mode which is designed to exploit larger storage (memory) spaces
  - ❖ Largest machine today is the Z990-D32 which is 256GB…imagine the future if there was a 1 TB model! *
- **Function - z/Architecture is designed to exploit larger storage spaces; 32 GB, 64 GB, 128 GB, and so on**
  - – Benefit – z/TPF **positions** the customer **for growth** tomorrow while **protect**ing the **investment** of today

* This is not a statement of IBM direction

Feature – z/Architecture

- **Function – Provides addressability to a maximum of**

<u>16 exabytes</u> of memory or 264 = 18,446,744,073,709,551,616**

- ❖ Eighteen Quintillion, Four Hundred and Forty Six Quadrillion, Seven Hundred and Forty Four Trillion, Seventy Three Billion, Seven Hundred and Nine Million, Five Hundred and Fifty One Thousand, Six Hundred and Sixteen

- – Benefit – It is possible to have the memory and engine power (mips) to **run more messages per second in one physical processor**, which means that it might be possible for the customer to **run with fewer images**, which results in a direct **reduction in cost** as this may **reduce operational complexity** and **reduce the cost** of environmentals such as floor space.

Feature – z/Architecture

- **Function – The ability to exploit > 2GB of memory. Several existing TPF 4.1 large system tables, and most new z/TPF system tables reside above 2GB. Customer tables can reside above 2GB. Programs can reside and execute above 2GB.**
  - Benefit – Large tables in memory **reduce** the amount of **disk I/O** required, resulting in **faster throughput.** This may **improve performance** which results in a **reduction in cost** and **response time** for Uniprocessor as well as Loosely Coupled systems.
  - Benefit – Optimizes **price performance** of system services and middleware such as web servers, mail servers and MQSeries.

Expanding and Positioning for Growth

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Feature – z/TPF Lifting Constraints

- **Function – Virtual File Access (VFA) is above the 2-GB Bar. All VFA blocks will be accessed with 64-bit addressing.**
  - Benefit – Can have enormous amount of records in VFA/memory which may **reduce the amount of disk I/O** required, resulting in **faster throughput**. This may **improve performance** which results in a **reduction in cost** and response time for Uniprocessor as well as Loosely Coupled systems.
- **Function – Support for as many as 255 subsystem users (SSUs)**
  - Benefit – Allows for more customer hosted systems, and capability for **business growth**.

Feature – z/TPF Lifting Constraints

- **Function – Support for symbolic device addresses (SDAs) greater than x'7FFF' including tape addresses.**
  - Benefit – Provides greater address assignment flexibility. For example a customer could place tape addresses in the Axxx range, and DASD in the 9xxx range, thus **reducing** operational **complexity.**
- **Function – New DASD limit is 40,000 logical volumes.**

(32760 cylinders/mod) * (15 tracks/cylinder) * (12 4K records/track) * (4095 bytes/record) = 24.1 GB

(24.1 GB) * (19,999 DASD Volumes) = approximately 481 TB

  - Benefit – Provides enormous **ability to scale for business growth**.
  - Benefit – Allows more data at a single point of control which **reduces complexity.**
- **Imagine if there was the possibility of…. (65520 cylinders/mod) = approximately (48.2 GB per module) * (19,999 DASD volumes) = approximately 962 TB (almost a Petabyte of storage)**

Feature – z/TPF Lifting Constraints

- **Function – Software designed for support for as many as 86 I-Streams (Tightly-coupled)**
  - Benefit – Couple this function with the immense addressing capabilities of z/architecture, and it is possible to run extremely CPU–intensive applications that have the ability to access an enormous database. This scalability **positions** the customer for incredible **business growth**.
  - Imagine what you could do with:
    - Two+ z990-D32 256 GB machines
    - Each with 32 I-Streams
    - Accessing a combination of 19,000+ primes and 19,000+ duplicate modules.

Feature – Tightly Coupled Load Balancing Enhancement

- **Function – The Tightly Coupled scheduler has been updated to more efficiently calculate the resource use of a processor and minimize response time**

In TPF 4.1 ECBs are assigned to an I-stream by the tightly coupled scheduler and remain on that I-stream unless the application program specifically switches I-streams by using the SWISC macro.

In z/TPF you can set the I-stream affinity of a program to indicate whether the scheduler can dynamically balance entries.

- Benefit – Load balanced processes and provides a more **consistent response time at higher utilizations**.
- Benefit – Provides **operational flexibility** with the ability to set the I-stream affinity of a program via the ZAPAT command.
  - AFFINITY=PROGRAM, the scheduler will not attempt to balance the ECBs running in this program dynamically and will not move them to other I-streams.
  - AFFINITY=NONE, entries processed by the program will be balanced dynamically by the scheduler.

❖ Note: For the initial migration to z/TPF, leave your application programs defined as AFFINITY=PROGRAM. After you have migrated, examine the most frequently run programs and determine which ones can be changed to AFFINITY=NONE.

Feature – Tightly Coupled Load Balancing Enhancement

- **Function - Ability to switch to another I-Stream, and continue processing on the next sequential instruction.**
  - Benefit – **Reduces** program **complexity** and **eases programming effort.**

Before: When a programmer wanted to switch I-Streams, it was necessary to set an indicator so that the called program could recognize that this was a switched ECB. The called program then checked the indicator and branched to the appropriate logic.

Now: The programmer needs only to use an option on the SWISC macro to indicate that the next sequential instruction should be executed.

Feature – z/TPF Two Addressing Modes

- **Function – z/TPF supports 64-bit addressing mode and 31 bit addressing mode.**

  C programs will execute in 64-bit addressing mode and will mostly be a recompile of the TPF 4.1 source.

  Assembler programs can execute in 31-bit mode or 64-bit addressing mode.

  TPF 4.1 assembler programs running below the 2GB bar that call other programs below the 2GB bar will only require a reassembly.

  - Benefit – Initially keeping programs in 31-bit mode may **ease the migration effort** as well as **reduce** migration **costs**.
  - Benefit - Customers may take advantage of 64-bit addressing mode while still **protect**ing the **investment** of current 31-bit addressing mode software.
  - Benefit – Increased **flexibility, scalability** and **performance** are obtained when using 64-bit addressing.

Feature – z/TPF Virtual Addressing Advantages

- **Function – In z/TPF an SVM address does not equal a real address, in z/TPF SVM is truly virtual.**

  Do not require >2 GB of real storage to run z/TPF.

  VFA will always use 64-bit virtual SVM addresses regardless of the real storage size of the machine.

  Can have a 512 meg machine running z/TPF with SVM addresses above the 2GB bar.

  - Benefit – Customer does not need to have a test system as large as the production system in order to use 64-bit virtual addresses. This provides **greater flexibility** in system hardware resource usage.

Feature – Memory Configurations per Processor

- **Function – Ability to define as many as eight memory configurations for a processor.**
  - Benefit – Configurable using the ZCTKA PREFER command to specify a memory configuration as *preferred* so that CTIN will try to use this configuration first.
  - Benefit – Uses the best fit principle; that is, it uses the memory configuration that requires the most amount of storage after meeting the requirements of a minimum amount of both VFA and 31-bit system heap.
  - Benefit – A physical processor can run different logical processors using different memory configurations in the same Keypoint A. Before it was necessary to have multiple versions of Keypoint A, each load deck with it's own. This provides **operational flexibility** as it is possible to define the test and production systems which are of various and differed sizes.
  - Benefit – **Reduces complexity** in maintaining systems

Feature – z/TPF Virtual Addressing Advantages

- **Function – SVM and EVM are not mapped 2GB – 4GB**

  Example: If the top bit is ‘on’ in a 31-bit address 9F42769C, the PSW is in 31-bit mode and attempts to use the address, a system error (CTL-4) will take place.

  – Benefit – Intentionally designed to find problems where programs in 31-bit addressing mode pass an incorrect pointer to programs in 64-bit addressing mode. Finding these errors on the test system **eases the migration effort**, and **protects the production system**, resulting in greater **system stability** which **protects** the customer’s **investment**.

❖ **Note:**

z/TPF is not wasting real address space since these skipped addresses are virtual.

Feature – z/TPF Exploits z/Architecture Instructions

- **Function – New Relative Instructions (BRC, LARL) available for use.**

  Branch Relative Instructions allow a programmer to write programs without a base register.

  The assembler will generate a branch using the number of half words relative to the current location.

  - Benefit – **Program expansion** and **programming is easier/faster** because you are not limited to a 4K program with one base register
  - Benefit – Potentially **improves performance** as ENTER/BACK mechanism is not used as often

  Note: Every Branch Relative Instruction has an equivalent Jump instruction for example BRC and JC have the same Op code A74. The assembler will recognize BRC and JC and produce the same Op code.

Feature – z/TPF Exploits z/Architecture Instructions

- **Function – New Immediate Instructions e.g. AHI, CHI, LHI, MHI are available for use. You don’t need a base register because the compare operand is in the instruction.**
  - Benefit – **Programs can be written without a base register**, which makes them easer to write and more efficient which **reduces** development **cost**.
- **Function – New Immediate Instructions for a 64-bit register. NIxx, OIxx. Saves a register, and an instruction. Example:**

```
Before: L R1, =X’00000001’          Load R1 with mask value of 1
          OR    R2,R1               Turns on bit 31 in R2
With z/TPF: OIHL R2,=x’0001’        Or Immediate High Low to turn on bit 31 in R2
```

  - Benefit – Programmer has **more flexibility in register** usage, programs are easier to write and more efficient which **reduces** development **cost**.

Feature – z/TPF Exploits z/Architecture Instructions

- **Function – New Instructions for portions of a 64-bit register – TMHH, TMHL, TMLH, TMLL. Provides the ability to test one bit that is in a register, before there was no test under mask on a register.**
  - Benefit – Programs are easier to write and more efficient which **reduces** development **cost**.
  - Benefit – **Programs that are easier and faster to write enable the customer to effectively manage skilled resources to write software for competitive products and solutions, which in turn potentially results in a gain in market share.**

Feature – z/TPF Extensive and Enhanced Support of Open and Widely Used Protocols

- **Function – z/TPF supports Standard Protocols; FTP, HTTP, IMAP4, MATIP, POP3, SNMP, SMTP, SOAP, TCP/IP, TFTP, SSL, WebSphere MQ for z/TPF, XML.**
  - Benefit – Utilizing standards is the key to making **integration of heterogeneous systems** possible. Standards allow each component to use the same mechanism to talk to other components.
  - Benefit – Facilitates Web Services, Service Oriented Architecture, and Grid Computing technologies. Application functions can be packaged as **reusable components** that can be used in business processes.
  - Benefit – By using standards it is possible to **align** a SOA to a **customer's business goals** and directly tie **business processes** to underlying **applications**.
  - Benefit - Opens up a whole new world in the way a company can integrate business practices. Business teams can strategize with system architecture teams to **architect solutions amongst heterogeneous platforms**.
  - Benefit – Customer is able to **deliver new business applications** while **maintaining the investment of existing core systems.**

Feature – z/TPF Extensive and Enhanced Support of Open and Widely Used Protocols

- **Function – FTP Client is planned to be available shortly after GA. This allows z/TPF to initiate FTP functions to other platforms.**
- **Function – Also Secure FTP Client, meaning FTP over SSL, is planned to be available shortly after GA. This allows encrypted data to be sent to and received from other systems.**
- Benefit – Allows for greater **flexibility** and **security** in data access and data offloads.

"Standards are critical to interoperability within a distributed computing platform, especially an advanced platform that provides a service-oriented, loosely coupled, cross-platform programming model.

Standards enable platform services to more simply integrate with middleware and infrastructure. This in turn also helps to reduce the complexity of heterogeneous and cross-enterprise orchestration and integration." **Source IBM Systems Journal Vol 43 No4 2004**

IBM Software Group

Open Development Environment

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Feature – Open Development Environment

- **Function – z/TPF is utilizing the GNU Compiler Collection (GCC) compiler as it's C, C++ compiler and linker.**

  GCC includes:

  – Front ends for C, C++, Objective-C, Fortran, and Java

  – Libraries for these languages (libstdc++, libgcj,…)

  – z/TPF currently supports C, C++

- **Function – The IBM S/390 HLASM product will be ported from z/OS to Linux for z/Series. (statement of direction)**

  – This provides the capability to perform assemblies on the Linux system.

  – Runs on systems with the identifier s390-ibm-linux or s390x-ibm-linux such as SuSe SLES8 or 9, or RedHat RHEL 3 or 4.

Feature – Open Development Environment

- **Function – An additional utility, the goff2elf converter, is run against the HLASM output to create an ELF object file.**

  The output generated by HLASM is a z/OS GOFF (Generalized Object File Format) object file instead of the required ELF (Extended Link Format) object file.

  The resulting ELF object is then compatible with the ELF based objects from the GNU GCC C/C++ compiler and can be linked to become a shared object file (.so) that is loadable onto z/TPF.

Feature – Open Development Environment

- Benefit – Standardizes the development environment which enables the customer to **eliminate multiple tooling solutions** in the enterprise, and **gain synergies** by the use of **common skills**.
- Benefit – Productivity can be greatly increased by taking advantage of the millions of lines of GNU based, open source code available. **Porting existing code** enables the customer to **improve time to market** in order to **gain market share** either directly by the code that is ported or indirectly by freeing up programmer resources to invent in-house that which is uniquely competitive to **gain market share**.

Feature – Open Development Environment

- Benefit – Enables the customer to **hire anyone with Linux skills** to work on z/TPF as the tooling (desktop tools, compilers and libraries) are all common. A z/TPF programmer is not required to know Assembler, JCL, PDS files or anything of traditional mainframe infrastructure.
- Benefit - As the assembler programmer pool shrinks, the customer can **protect current investment** without having to find assembler programmers.
- Benefit - Customer can **hire programmers at more competitive rates** which may result in **reduction** in software development **cost.**

Feature – Open Development Environment

- **Function – The makeTPF toolset is a collection of scripts and configuration files that simplifies the creation of makefiles and other build steps.**
- **Function - It is based on standard makefiles, but it predefines rules and variables that make it easier to define and perform the build steps of applications.**
- **Function – Builds run primarily on Linux, Unix System Services on z/OS used for remaining z/OS based offline programs.**
- **Function – Make TPF Build Tools replace SIP stage 1 & 2**
- **Function - GNU Make and Korn Shell replace JCL**
- **Function – GNU Compiler Collection (GCC) replaces z/OS C/C++**

Feature – Open Development Environment

- maketpf – a single program builder for MakeTPF format makefiles used to assemble, compile, and link an application program.
- bldtpf – a multiple program builder for MakeTPF format makefiles used to assemble, compile, and link a set of application programs. Also used to drive SIP deck assembly, Face table generation, Load deck generation, Stub library source generation, PAT-to-control file conversions.
- loadtpf – a script that packages program segments into a loaddeck and FTPs it to z/TPF
  - Benefit – Highly **customizable** as it can support unique customer application directory structures, rules and build audits.
  - Benefit – A full system build now runs in 2 ½ hours comparable to TPF 4.1 build which requires 6-8 hours, this is an immediate **cost savings**.
  - Benefit – System and development builds use the **same tools**.

Feature – Open Development Environment

- **Feature – The IBM Toolkit for WebSphere Studio simultaneously supports both the TPF 4.1 system and the z/TPF system.**

  This product is the follow-on to VisualAge TPF for Microsoft Windows NT and represents the next generation of application development technology.

  - Benefit - Complete with Editor, Debugger, and Performance Analyzer, this development tool assists Assembler and C/C+ programmers to **develop faster, easier, and produce more efficient software**.
  - Benefit – Single Source tools will **ease migration** as it provides visual queues showing existing code flagged with areas that should be investigated, and where possible, provide the suggested migration to z/TPF.
  - Benefit – Once a MakeTPF environment is created for both TPF 4.1 and z/TPF, the TPF Toolkit can seamlessly work with both environments simultaneously in the Individual Development Environment. This provides **standard tooling** for the developer.

Feature – z/TPFDF is a Co-requisite of the z/TPF system

- **Function - The z/TPFDF product is a database manager for application programs that run in a z/TPF operating environment.**
  - Benefit – **Increased productivity** of application programmers as z/TPFDF provides:
    - A logical method to organize the database.
    - A set of standardized assembler macros or C functions that form the application programming interface (API).
    - Central routines to access and manage the database.
    - Utilities for database maintenance and testing.

Feature – New additions to the z/TPF Base Product

- **Function – z/TPF Application Requester (z/TPFAR) includes the standard TPFAR SQL functions that allow TPF to write or read data to/from a DRDA level-3 compliant database, eg. DB/2 or UDB.**
  - Benefit – Provides the ability to communicate between programs running in different systems that are controlled by different operating systems. This provides **flexibility** in accessing databases.
  - Benefit – **Reduces cost**, as it is included in the base product.
- **Function – The multi-processor interconnect facility (MPIF) is included in the base product.**
  - Benefit – **Reduces cost**, as it is included in the base product.

Feature – New, Additional Mountable File Systems

- **Function – Increased support and functionality of POSIX files. Options now include:**

  1. z/TPF collection support file system (TFS) previously referred to as the TPF file system.

  - Subsystem-unique
  - Processor-shared
  - Uses z/TPF collection support as its storage mechanism
  - Maintained across processor IPLs.

  2. Memory file system (MFS)

  - Subsystem-unique
  - Processor unique file
  - Not maintained across processor IPLs
  - Reinitialized every time it is mounted
  - Uses system heap as its storage mechanism.

Feature – New, Additional Mountable File Systems

3. Fixed file system (FFS)

- Processor-unique
- Built using a fixed file record type for both i-nodes and data records
- Maintained across system IPLs
- To be used where performance is more important than minimizing the use of file space.

4. Pool file system (PFS)

- Processor-unique
- Built using a fixed file record type for i-nodes
- Built using pool records for data
- Provides a file system for files that need to be maintained across system IPLs with a high level of performance, but by using pool records for data, PFS provides a more efficient use of file space.

Feature – New, Additional Mountable File Systems

- Benefit – Assists customers to move applications toward POSIX compliant and **open standards**.
- Benefit – Programmer has the **flexibility** to select the device driver that suits the business need.
- Benefit – Customers using short term pool records to satisfy I/O requests now have the ability to replace these proprietary TPF GFS calls with **standard file system commands** without a loss of performance.
- Benefit – Memory based file system has no I/O therefore is very **fast**.

Available on request – education to assist in understanding, choosing and designing databases using file systems.

IBM Software Group

Coexistence Capabilities

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Feature – Coexistence

- **Function – Customers with Loosely Coupled systems have a TPF 4.1 system and z/TPF system running in the same complex.**
- **Function – Can also maintain a single set of source for applications that need to run on both the TPF 4.1 system and the z/TPF system.**
  - Benefit – **Eases the migration effort** and allows customers to **protect the investment** of the current TPF system while **managing the skilled resource base** as work is performed to migrate to z/TPF.

Note: Must apply specific single source APARs to accomplish this.

IBM Software Group

Easing the Migration Effort

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Easing the Migration Effort

- **Single Source APARs can be installed in TPF 4.1 that allow the customer to use the same application source for both the TPF 4.1 and z/TPF systems. See your CSR for more information.**
  - Benefit – As code is ‘touched’ in the normal development process, it can be modified which **eases** the overall **migration effort** in coding and testing, and enables the customer to **effectively manage skilled resources**.
- **z/TPF incorporated many customer user modifications and user exits. See your CSR for more information.**
  - Benefit – Customer now has code in the base product and no longer needs to retrofit these in each Control Program or E-type APAR update which will **reduce costs**.

Easing the Migration Effort

- **z/TPF Migration Tools to automate migration/conversion tasks. See your CSR for more information.**
  - Benefit – Customer can reduce manually intensive tasks by using these tools which results in **reduced cost** and **effective management of skilled resources**.
- **z/TPF Migration Team - TPF Development Lab, TPF Services, TPF Support, and the TPF Business Team prepared to assist customers.**
  - Benefit – This team developed several documents that are available through your CSR such as; a Generic Planning Document and accompanying Skeleton Project Plan, a Planning Worksheet, an Education and Roadmap. In addition, a web enabled Migration Portal will be available as of April 15th which will house the z/TPF and z/TPFDF Migration Guide and other information and guidance.

z/TPF Pre-Migration Education and Services

- **Feature, Functions & Benefits of z/TPF –** A one day course presenting an overview of the features and benefits of z/TPF with a focus on migration considerations. There is no charge for this course.
- **Web Services and z/TPF –** A two to three day course focusing on application conversion and Web Service opportunities.
- **Single Source Considerations –** A two hour presentation discussing the modifications to TPF4.1 programs that can be applied today to ease migration to z/TPF.
- **Application Analysis –** A one week consulting engagement to review applications and areas of focus with respect to z/TPF migration preparation.

z/TPF Pre-Migration Education and Services

- **z/TPF Pre-Migration Analysis -** A consulting engagement, up to two weeks in length, to analyze the current environment and identify areas of focus for z/TPF migration. This offering will result in the creation of an initial Project Plan as well as define the tasks required for systems and environment preparation.
- **Remote Source Control Manager/Development Infrastructure –** A consulting engagement focusing on the Linux infrastructure in preparation for a migration to z/TPF. This engagement will result in Linux system installation, configuration and set up for use with z/TPF.
- **z/TPF101 –** A two day course providing z/TPF concepts and structures from a functional and business usage perspective. The intended audience for this course is Business Analysts, Systems, Applications, Operations, and Coverage personnel, who have had little if any exposure to TPF.

z/TPF Migration Education and Services

- **z/TPF Migration –** A one week course presenting a detailed overview of z/TPF internals.
- **z/TPF Toolkit –** A one week course focusing on the installation and customization of the z/TPF Toolkit.
- **z/TPF Dump Analysis –** A three and a half day course to assist customers in acquiring and analyzing z/TPF problem information.

z/TPF Migration Education and Services

- **z/TPF Migration Services –** A customized offering designed to review pre-migration analysis and assist in migration efforts. The duration of this offering may vary; these services could include:
  - Assistance in establishing and validating the initial zTPF development environment , as well as providing guidance on Source Control Managers for the z/TPF environment
  - Assistance with migration efforts including project management or periodic project assessments
  - Onsite or on call services during key cutovers
- **z/TPF Linux Build Environment –** A one week Services offering covering the following areas:
  - Linux prerequisites
  - File structure
  - GCC cross compiler source installation and build
  - Introduction to Linux shell scripting
  - Introduction to standard make
  - z/TPF build tools explained
  - z/TPF system code installation
  - Linux system configuration and customization for z/TPF builds
  - MakeTPF details and customization
  - z/TPF Linux offline programs
  - SIP process
  - Build and load z/TPF applications.

z/TPF Post Migration Education and Services

- **Coding with z/TPF –** A one to two day course covering TPF's usage of the new z/TPF BAL instruction set.
- **z/TPF File systems -** A course covering the new file system capabilities of z/TPF. This course will cover TFS (the TPF File System), FFS (the fixed file File System), PFS (the pool File System) and MFS (the memory File System), as well as TPFDF and TPFAR.

z/TPF Post Migration Education and Services

- **z/TPF TOS For System Administrators –** A two day course introducing the concepts, functions, installation and customization of the TPF Operations Server (TOS) to z/TPF systems. Note that this course requires an interactive classroom environment. Please refer to the TPF Education website for details.
- **z/TPF TOS for Operators –** A half day course introducing concepts of the TPF Operations Server (TOS) console automation environment to z/TPF operations personnel with emphasis on the client console interface. Note that this course requires an interactive classroom environment. Please refer to the TPF Education website for details.

Additional z/TPF Education

- **z/TPF Online Pool Maintenance (Recoup) –** A two day customized course which is designed to introduce the online facility to experienced TPF programmers. Instructors can use customer provided record descriptors from their current recoup as a basis for class discussion. The course if normally two days but the duration will vary based on customization requirements
- **z/TPF C/C++ Architecture and Internals –** A four and a half day course designed to provide students, with z/TPF knowledge and basic C/C++ experience, a way to efficiently put C/C++ to work in their respective z/TPF environment as well as be able to analyze z/TPF systems dumps involving C programs.

Additional z/TPF Education

- **z/TPF C/C++ workshop for new programmers –** A nine day interactive workshop which is designed for experienced TPF programmers who are migrating to a C/C++ environment. It will introduce students, with a basic knowledge of BAL and C/C++, to the TPF Toolkit development environment and provide a working knowledge of the TPF Toolkit. Note: This course requires an interactive classroom environment.
- **z/TPF C/C++ workshop for experienced programmers –** A four and a half day interactive workshop which is designed for programmers *with a minimum of three years C/C++ experience* who are new to either TPF Toolkit for WebSphere Studio and/or TPF applications development. It is designed to provide a working knowledge of the TPF Toolkit development environment and introduces the TPF C/C++ API to students. Note: This course requires an interactive classroom environment.

Additional z/TPF Education

- **z/TPF TPFCS –** TPF Services offers two courses designed to provide an understanding of z/TPF Collections Support to students with basic TPF knowledge.
- **z/TPF Collections Support for Applications Programmers -** a three day course which focuses solely on applications.
- **z/TPF Collections Support for Systems and Applications Programmers -** a four and a half day course covering both systems and applications topics.
- **z/TPF Communications Overview –** A two and a half day course providing new z/TPF systems or applications programmers with z/TPF communications training. It introduces a high level overview of the z/TPF communications environment and the protocols supported including MATIP, MQ and TCP/IP.

Additional z/TPF Education

- **z/TPF TCP/IP Introduction & APIs –** A three day course designed for students who will support z/TPF in a TCP/IP networking environment. It introduces z/TPF's implementation of TCP/IP networking concepts and the Application Program Interfaces (APIs) it supports.
- **z/TPF TCP/IP Internals –** A three day course providing an introduction to TCP/IP networking concepts and the z/TPF APIs that support TCP/IP.
- **MQ Series for z/TPF –** A two day course providing an introduction and overview of z/TPF's implementation of Websphere® MQ Series® support for TPF and MQ programmers. The course is designed to enable students to incorporate z/TPF in an MQ environment and to analyze/diagnose MQ data flows to, from and within z/TPF.

Enhancements for Applications

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Feature – Program Packaging Flexibility

- **Function – Assembler segments can be optionally re-packaged into callable libraries similar to the C language based dynamic link library (DLL), referred to as ‘shared objects’.**

  Example: A group of 100 Assembler programs could either be loaded as 100 BSOs (BAL Shared Objects) or repackaged into a smaller number of BSOs, grouped by logical function.

  - Benefit – **Reduction of system overhead** because of faster program linkage. Linkage has been changed from a system service to a function call similar to C language.
  - Benefit – Reduction in system overhead for an application because two segments can be linked together as one unit, creating a **shorter path length**.
  - Benefit – Program expansion beyond 4KB. To exploit this capability, convert assembler programs to use baseless instructions. Greater than 4K programs are supported which **eases programmer constraints** which may result in **faster programming**.
  - Benefit – **Promotes reusability** of the package.

Feature – Application Programming Enhancements

- **Function – Shift from assembler to C/C++ as the primary development language and programming model**
  - Benefit – Ability to **obtain readily available experienced programming resources**
  - Benefit – Ability to easily **port code** which **reduces cost** and **improves time to market**
  - Benefit – Facilitates **hiring at competitive rates**
  - Benefit – Facilitates **integrating valued content knowledge with the next generation of programming talent**
  - Benefit – **Gain synergies by the use of common skills**
- **Function – Support for 4-byte Wide character support in UCS-4 (unicode format)**
  - Benefit – Eases support of double byte languages such as Japanese and Chinese which results in **improved time to market** for support of new business opportunities and **gain market share**.

Feature – Application Productivity Enhancements

- **Function – ALL programs (accessible from both C and assembler) have their own private stack area called the *application stack* which is configurable by the customer and provides immediate storage for use in the program. Previously in TPF 4.1 only C code had access to a stack.**
  - Benefit – **Improves productivity** as this reduces the likelihood of breaking existing code. For example, when you modify an assembler program in 4.1, use an EBWxxx field to save data, and find out that 3 programs back, the field was already being used.
  - Benefit – Provides an additional save area for registers and other values, which **eases programming constraints.**

Feature – Application Productivity Enhancements

- **Function – Four new additional registers are now available for applications use; R8,R11,R12,R13**
  - Benefit – Eases register constraint, and facilitates faster pipelining of code which may **reduce** programming effort, and **cost**, resulting in **improved time to market**.
- **Function – Assembler programs can now call a C function directly by using the new CALLC macro.**
  - Benefit – simplifies the interface between C and Assembler programs, and **eases the programming effort** which results in **cost reduction** and may **improve time to market**.

Feature – Application Productivity Enhancements

- **Function – The amount of main storage that is available for use by the application program is greatly increased. Before the ECB Private Area was limited to 1 meg of private storage attached to the ECB, now it is user-defined (CTKA) and can be up to 16meg.**
  - Benefit – applications now have more work area which **facilitates porting memory intensive C/C++ applications.**
  - Benefit – provides flexibility for the applications programmer which results in **reduced development complexity and cost**.
- **Function – Can assign a unique token to an area of system heap storage, and use this token later to access the area of storage by name instead of by address.**
  - Benefit – **Eases** the **programming effort** and provides **programming flexibility**.

Feature – z/TPF Debugger Enhancements

- **Function – The z/TPF Debugger runs with code compiled with the optimization option.**
  - Benefit – Allows the debugger to be used against the same object that is used in the run-time production environment which facilitates **easier, faster, more accurate debugging.**
- **Function – The z/TPF Debugger can be used with the z/TPFDF product.**
  - Benefit – Programmers debugging z/TPFDF programs now have all the advantages the debugger provides for z/TPF programs, which facilitates **easier, faster, more accurate debugging**.
  - Benefit – Provides **standard tooling** for z/TPF and z/TPFDF.

Feature – z/TPF Debugger Enhancements

- **Function – ASCII, EBCDIC, and Unicode translations displayed in dumps. Dump format has hex on the left side, and translation on the right.**
  - Benefit – Facilitates **faster error resolution** which **reduces** the **programmer effort** in resolving errors.
  - Benefit – **Simplifies** debugging efforts.

Feature - Format 2 Global Support

- **Function – Format-2 global area contains global records only.**
- **Function – Format-2 globals can reside below the 2-GB bar for 31-bit globals, or above the 2-GB bar for 64-bit globals.**
- **Function – Separate set of assembler macros and C language functions to access format-2 global records.**
  - Format-1 globals use application programming interfaces that are based on the physical position of the particular global record or global field in a global directory.
  - Format-2 globals use APIs that are independent of the physical position of the global record in a global directory. The result is that the layout of the format-2 global directory is transparent to the user application.

Feature - Format 2 Global Support

- **Function – A new set of commands (ZGLBL) allow the customer to define and manage format-2 global records dynamically.**
- **Function – There are no size restrictions for format-2 globals.**
- **Function – There is no maximum number or maximum size for records that can be keypointed or synchronized.**

Feature - Format 2 Global Support

- **Function – Format-2 globals reside in dynamically allocated areas of system storage (system heap).**
- **Function – Ability to define format-2 global records to be entirely system-controlled or entirely user-controlled through the use of user exits.**
- **Function – Format 1 globals (TPF 4.1) are not changed, however 24 bit globals are no longer supported.**

Feature - Format 2 Global Support

- **Function – Migration path available to go from Format 1 to Format 2 Globals**
  - Benefit – Globals are significantly **easier to define** and modify which **simplifies the use of globals** for the programmer.
  - Benefit – **Eases global maintenance** for Operations. Updates do not require the use of the system test compiler (STC), pilot tapes (GOA), or the online data loader.
  - Benefit – Programmer can have massively large globals above the 2 GB line facilitating **ease of programming** and **faster response time** for the end user.
  - Benefit – A great deal of power in the environment in terms of **application choices and flexibility**.

Feature - Update two different Global keys simultaneously

- **Function – New architecture allows the program to update to two different keys simultaneously.**

The process to update protected globals in C was inefficient.

z/TPF implemented a new architecture with the new z/series machines which allows you to update two different keys simultaneously. This facility is called Storage Protection Override.

Programmer can stay in Key 1 for the application and also update a protected global in Key 9 storage.

Instead of having globals in Key C, all protected globals will be in Key 9 in order to allow the simultaneous update.

Assembler programs will essentially be able to do the same thing.

We do require the program to turn off storage protection override prior to giving up control.

  - Benefit – Programmer is able to stay in Key 1 for the application and also update a Key 9 global and stack area. , which provides **flexibility**, **reduces programming effort** as well as **path length.**

Feature – Global Synchronization

- **Function – Ability to specify a WAIT=YES parameter on the SYNCC macro so that return is not given to the caller program until all I-streams have acknowledged that synchronization is completed.**
  - Benefit – Ensures that global updates have taken place on all I-Streams, thus providing for **database integrity**.

Feature – Keypoint Support Enhancements

- **Function – Keypoints can now exceed 4KB. New keypoints consist of a full track of 4KB records, that is, a total size of 48KB.**
- **Function - The 4KB records are logically concatenated into a single keypoint record, with the keypoint size in the logical record header. CTKA, CTKD, CTK2, CTK3.**
- **Function – New assembler macros and new C functions have been added to access these keypoints.**
  - Benefit – Improves performance, greater flexibility, and easier to manage.

IBM Software Group

Availability Enhancements

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Feature – RIAT Access

- **Function – The RIAT (Record ID Attribute Table) is now directly indexable by Record ID, it no longer uses a hashing mechanism.**
  - Benefit – Huge, 64K entry RIAT table **provides** for record **growth**.
  - Benefit – Accessing Records defined in the RIAT is now **faster** and **more efficient.**
  - Benefit – Ability to add RIAT entries without disrupting production workloads.
  - Benefit – **Fast access** method - Take the ID in hex and multiply it by the length of each item, and add that to the base of the RIAT table.

Feature – Enhancement of Timeout Processing (CTL-10)

- **Function - Every program can have its own timeout value, specified in the Program Attribute Table (PAT)**

  Example: The timeout value in the PAT table is the number of 10 millisecond intervals before a CTL-10 is taken. Max value is x'FFFF'. 65,535 x 10 milliseconds = 655 seconds = 10 min. 55 seconds. Which means an ECB controlled program could run for nearly 11 minutes without taking a CTL-10.

  - Benefit – Provides granular control of long running programs which may result in **system stability**.
  - Benefit – Provides **operational flexibility** and control as the value can be changed anytime with the ZAPAT command.

Feature – Improved Core Corruption Detection

- **Function – Within the ECB Private Area when a 4K is mapped, the next 4K is not mapped. Every other block is mapped.**
  - Benefit – If a program writes beyond it's block, it would receive an error. No processing overhead is associated with this enhancement which **detects core corruption** before the culprit overwrites core.
  - Benefit – Greatly **enhances system stability**.
  - Benefit – Real **storage is not wasted** as these are virtual addresses which are not being used.

❖ Note: We still have multiple 1055 blocks in a 4K block, unless you are in 'block check' mode. In block check mode if the program runs beyond the 1055 block, that is caught as well.

Feature - Improved Resource Abuse Detection

- **Function – Physical Block Owner Support allows certain resource functions such as GETBC and GETCC to have an OWNER to be identified.**
  - Benefit – The OWNER parameter allows the customer to identify the culprit who might be abusing system resources. This results in **improved resource tracking** and **problem determination** which may **improve system stability**.

Feature – Improved Resource Abuse Detection

- **Function - The ECB resource monitor provides a centralized facility that monitors the use of selected system resources for each ECB. The ECB resource monitor detects and, optionally, stops an ECB that requests excessive amounts of monitored resources.**
  - Benefit – Enables the customer to monitor the use of system resources before the system has a problem, thus avoiding unexpected downtime, and **ensuring system stability**.
  - Benefit – Provides **operational control and flexibility** with the ZECBM command to turn the ECB resource monitor on or off.

Improved Dump Processing

- **Function – Through the use of a dump buffer area (DBA) in memory, and then the paused I-Stream is released. The dump data is written to tape from the buffer area.**
  - Benefit – Managed via functional entry ZDBAI which provides for **operational flexibility and ease of use.**
  - Benefit - The time that the system stops while taking a dump is much shorter which **improves system performance**.

Improved Dump Processing

- **Function – Dumps can be written to the File System as well as to tape. From a file system perspective these files are processor unique, however they can be viewed on another system – any processor.**

  Programmer can use FTP to move the dump from TPF, bring up a debugger session, and the debugger will recreate the memory essentially just as it was at dump time. At this point, it is as if you stopped at dump time with the ability to look at all the registers, variables, look at the listing, etc.

  - Benefit - This is a powerful enhancement for dump analysis which can greatly **improve system stability and availability**.
  - Benefit – Makes **dump analysis quicker** as you don't need to issue commands to change tapes, get the tape, move the tape to z/OS, process the dump, etc.

Feature – Flexible Module File System Organization

- **Function – Customer has the option to generate the system either in the traditional module format or in the prime/duple module pairing format aka even/odd module pairing.**

Before Prime modules followed by Duplicates	With Even/Odd Module Pairing
	Prime Module is Paired with its Dupe Module
011 PR0001	011 PR0001
012 PR0002	012 PR0005 Dupe of PR0001
013 PR0003	013 PR0002
014 PR0004	014 PR0006 Dupe of PR0002
015 PR0005 dupe of PR0001	015 PR0003
016 PR0006 dupe of PR0002	016 PR0007 Dupe of PR0003
017 PR0007 dupe of PR0003	017 PR0004
018 PR0008 dupe of PR0004	018 PR0008 Dupe of PR0004
Difficult to expand	**Easy and safer to expand**

Feature – Flexible Module File System Organization

- **Function – User exits have been added at strategic points to facilitate customer specific DBR mechanisms.**
  - Benefit – **Eases the maintenance effort** for customers with unique DBR packages
  - Benefit – **Reduces** operational **complexity**.
  - Benefit – Customers with their own even/odd pairing will no longer need to retrofit their software modifications which results in **cost reduction**.

Feature – TCP/IP Messaging Priority Enhancements

- **Function – Ability to define a TCP/IP application, socket, or both as high priority so that it’s input messages go on the ready list instead of the input list.**

- **Function – Ability to define the discard priority for a TCP/IP application, socket, or both, which defines the order in which TCP/IP input messages are discarded. This allows the customer to throw away less important messages, in order to buffer the more important messages in memory**

Feature – TCP/IP Messaging Priority Enhancements

- Originally the TPF system was terminal centric and when overloaded, shutting down the input list could keep the system functioning.
- Today with transactions interacting with various platforms, going into input list shutdown prevents data from arriving from these platforms.

– Benefit - By placing high priority messages on the ready list these transactions can be completed even if the system is not accepting input messages. This facilitates **high availability and system stability**

– Benefit – Allows **prioritization** of network traffic **based on critical business need.**

– Benefit – Allows you to determine which messages are discarded first when the system runs low on TCP/IP resources which may prevent outages and **improve system stability**.

Feature – Availability Enhancements – 1052 State

- **Function – Via OSA connection, TCP/IP is available in 1052 state.**
  - Benefit – Allows tracing and problem resolution using the Debugger in 1052 state which in turn facilitates faster problem resolution and improved system stability.
- **Function – Ability to define a socket to be 1052-state capable so that the socket can be created in 1052 state. (SetTCP1052 function). Before sockets could not be started below CRAS state.**

Feature – Availability Enhancements – 1052 State

- **Function – Customer has the option of obtaining Get File Storage in 1052 state.**
  - Benefit – Operational command ZPOOL 1052 provides **security** and **flexibility** in turning this feature on and off.
  - Benefit – Allows the customer the **flexibility** to run any application that use GFS-type APIs in 1052 state. For example database/file maintenance could be performed whilst the database is not available to the end user**.**
- **Function – FTP is available in 1052 state**
  - Benefit – Provides greater **flexibility** as you could load a database on the File System, before it can be used by the Network.

Feature – Availability Enhancements – 1052 State

- **Function – Can define applications to the Internet daemon as 1052-state capable.**
  - Benefit – Provides greater flexibility when communicating with various platforms.

IBM Software Group

New Concepts for Programs and Loaders

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Feature – New Concepts for Program Allocation

- **Function – Dynamic Linkage resolves external references online which means there is no need to maintain compatible offline Allocator Table.**
  - Benefit – **Reduces complexity** and **reduces** the **dependency on z/OS**
  - Benefit – SALTBL and TABLE40 are obsolete and there are no more timestamp mismatches. Less errors results in more **efficient use of programming talent** and **reduction in cost**.
- **Function – System dynamically maintains order of programs in the PAT (Program Attribute Table)**
  - Benefit – No need to be concerned with order of programs in offline files (control files in z/TPF, formerly IBMPAL and USRTPF in TPF4.1) to prevent allocator shifts.
  - Benefit – No need to maintain program order which reduces complexity and process can be performed with greater efficiency and accuracy, which results in **reduction in cost**.

Feature – New Concepts for Program Allocation

- **Function – Ability to allocate new programs dynamically.**

  Before it was necessary to create a new program allocation table offline, load the table online using the ZOLDR LOAD command with the PATU parameter specified.

  - Benefit – Provides operational flexibility through the use of the ZAPAT ADD command.
  - Benefit – Saves time and effort if update is needed immediately.

- **Function – Ability to display program listings online using the ZDPGM command with the LISTING parameter specified.**
  - Benefit – **Enhances problem diagnosis** which may **improve system stability.**

Feature – New Concepts for Program Allocation

- **Function – Program Attribute Table (formerly known as the program allocator table) has new options, is defined in the control file and built from the control file using a makefile.**
  - Benefit - Improved granularity of program attributes enables stricter controls which may result in improving **system stability**.
- **Function – New online command, ZDECK, allows the customer to extract data from the online PAT table to a hierarchical file system (HFS) file that can be used as input to create the offline control file.**
  - Benefit – By capturing the online PAT table, changes that were made online via ZAPAT are captured. A utility called pat2ctl which is a part of bldtpf takes the PAT file written by the online ZDECK command and creates a new control file. This combination of processes eases the maintenance effort, which may **reduce** manpower (**cost**) as well as make the process less prone to error, resulting in **system stability**.

Feature – New Load Deck Changes and Options

- **Feature – Removal of fixed column restrictions**
  - Benefit – Provides **greater flexibility** and is easier to create.

- **Feature – More options for search paths such as multiple paths for a given component type and override path with program lists.**
  - Benefit – **Granularity and flexibility promote accuracy and efficiency.**

Feature – New Load Deck Changes and Options

- **Feature – New Patch format allows validation of data and is consistent with ZAPGM format**
  - Benefit – Easier to validate and cross check to ensure patch accuracy and **system stability**.
- **Feature – Option to verify programs against IPAT**
  - Benefit – Reduces manually intensive comparison task, promotes accuracy which in turn reduces errors and ensures **system stability**.

Feature – Program Load Facility Enhancements

- **Function – Ability to use the File Transfer Protocol (FTP) to load E-type programs/data into TPF as well as current media (GDS, Tape, VRDR)**
  - Benefit – Reduces the need for offline media for software loads
  - Benefit - Speeds up the load process

IBM Software Group

External Storage Enhancements

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Feature – Write Large Block Sizes to Tape

- **Function – z/TPF supports block sizes from 32,760 bytes to 128K.**

  Note: RTA tape block size must be 32,760

  RTL tape block size must be 128 KB

  - Benefit – Provides operational ease in setting the block size for a tape via the ZTLBL command.
  - Benefit – Provides operational flexibility to override the block size via the ZTMNT command.
  - Benefit – Reduces programming complexity when writing large amounts of data to tape.
  - Benefit – Writing large amounts of data to tape is faster.

Restriction: Can no longer force a dump from the RTL tape to the RTA tape if the RTL tape becomes unavailable. Therefore you must have an RTL standby or ALT tape mounted if the RTL tape becomes unavailable.

Feature - Enhanced tape write and read capabilities for non-contiguous storage

- **Function – Ability to read and write multiple non-contiguous storage areas to tape using C functions tpf_tgetc and tpf_tputc as well as Assembler macros TGETC and TPUTC.**
  - Benefit – **Reduces programmer effort and provides flexibility** when writing non-contiguous areas to tape.

IBM Software Group

Problem Determination

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

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Feature – Combined macro trace and C function trace facilities

- **Function – Ability to have 32 different levels of ECB trace; 16 for macros and 16 for functions in a combined macro trace and C function trace in a single trace facility called *ECB trace.***
  - Benefit – Provides **operational flexibility** to start and stop via functional command ZSTRC.
  - Benefit - The C portion of ECB trace includes parameters, return value, types, and names for all functions to facilitate **faster problem resolution**.
  - Benefit – The application trace remains unchanged no matter what other activity is taking place. For example middleware, like MQ, will no longer interfere with application macro trace, and programmer will always have the last 50-60 functions/macros that the application performed which facilitates **faster problem resolution**.
  - Benefit – Available for functions other than dumps, such as ZDECB displays, which facilitates **faster problem resolution**.
  - Benefit – Facilitates easier and faster system error resolution which can result in **reduced programmer effort** and **improved system stability**.

Feature – Trace Support Enhancements

- **Function – ECB heap trace provides additional trace information about the latest ECB heap storage requests and releases for an ECB.**
  - Benefit – Facilitates easier and faster system error resolution which can result in **reduced programmer effort** and **improved system stability**.
- **Function – Ability to turn the register trace facility on or off without IPLing the system.**
  - Benefit – Provides greater operational flexibility and **system stability**.

Feature – Trace Support Enhancements

- **Function – Socket trace provides the capability to trace socket APIs and debug TCP/IP socket applications with two types of traces.**
  - **Socket level trace** traces all socket APIs that were issued on a given socket. When the socket API trace is enabled, can display the socket trace online using the ZSOCK command.
  - **ECB level trace** traces all socket APIs that were issued by a given ECB. When the socket API trace is enabled, the ECB level trace is included and formatted in the dump output.

– Benefit – Greatly improves diagnostic capability which may improve **system stability**.

Debugging and Tracing Consolidated Business Benefits

- **The next few slides highlight many trace and debugging functions.**

  Each function has the following benefits:

  - Facilitates easier, faster problem determination and resolution
  - Facilitates system Stability, Availability, and Reliability

Feature – Trace Support Enhancements

- **Feature – Trace groups** define how application trace data is separated from system trace data to maintain as much information as possible. Trace group levels are used to map trace data to different buffers for a given ECB.
- **Feature – Trace Log Facility** provides the capability to trace application function and macro calls to a file or real-time tape so that application debugging is not disruptive.
- **Feature – I/O Trace** was enhanced to include the ZIOTR DISPLAY and ZIOTR SET COUNT commands. The ZIOTR DISPLAY command allows the customer to display I/O trace information online. The ZIOTR SET COUNT command allows the customer to change the number of trace entries that are used to record trace information for one or more I/O devices**.**

Feature – Debugging Enhancements

- **Function – Heap Check mode,** flags certain coding errors related to the use of ECB heap to assist in determining when an application is changing or accessing storage beyond its allocated heap buffer or accessing an ECB heap after it was released.
- **Function – Block Owner** display which tags core blocks with a system component that obtained the block. For example, TAPE, DASD, COMMS. If the system is running out of core blocks, use the ZSTAT command with the OWNER parameter specified to display the block usage and determine which component is having a problem.
- **Function – With the z/TPF debugger,** ability to view a captured ECB dump on the z/TPF console.
- **Function – With the z/TPF debugger** can take a snapshot of a long running ECB and view it with the debugger.

Feature – Stand-alone Dump and Postprocessor Enhancements

- **Function – The stand-alone dump (SADUMP) utility** is an offline utility that is used to dump various storage areas when a normal online dump cannot be processed because of a system hang condition. SADUMP is now updated to handle z/Architecture machines and has been enhanced to include new areas in the dump.
- **Function – The stand-alone dump postprocessor (PPSADUMP) utility** is a C language program that runs on Linux to create a printed storage dump from the data that was captured on the SADUMP tape. PPSADUMP replaces the SADPRT utility on the TPF 4.1 system that ran under IBM MVS.

Feature – Operations Enhancements

- **Function – ZACNF and ZDCNF** commands were added to allow you to change and display data by referencing the CINFC label.
- **Function – ZDCOR** was enhanced to support 8-byte addresses, and if the address is not on a fullword boundary, no fullword boundary alignment is done…the display begins at the address specified.
- **Function – ZDMAP** displays the link map data for all programs, assembler and C language.
- **Function – ZDECB** has a new parameter, TR, to display an ECB trace for an ECB that is in use. (replaces MTR parameter)

Feature – Operations Enhancements

- **Function – New ZDECK** command to extract data from an online z/TPF table to a hierarchical file system (HFS) file that can be used as input to create an offline control file.
- **Function – ZDSVC** command updated with a new parameter, CODE, which gives the ability to display a macro name by specifying the SVC number.
- **Function – ZAPFS and ZDPFS** commands to manage positive feedback support. Positive Feedback support provides a way to keep offline information for the z/TPF system synchronized with online information.
- **Function – ZDSYS** command has a new parameter, ALL, which provides the ability to display the system state for all subsystems.

Feature – Dump Format Enhancements

- **Function - Dumps provide the executable name and the object file name.**
- **Function – The format of the dump has changed to have hexadecimal information on the left side and EBCDIC or ASCII translation on the right side for most items.**
- **Function – The entire program is no longer included in the dump, instead, the linkmap for the failing program and the 4KB area surrounding the error or SERRC will be included in the dump.**
- **Function – Only Heap storage that was in use at the time of the error will be dumped unless specified otherwise on the ZIDOT command or SERRC macro. The heap information will indicate who received the storage and what size was requested.**

System Error Processing – Customizing Dumps

- **Function – New parameters on the ZASER command allow the customization of the percentage of frames and system work blocks (SWBs) to be included in the dump.**
- **Function – Ability to customize the dump by using the OWNER parameter which allows the customer to dump storaged based on the owner of the block. Modifiable realtime via the ZIDOT command.**
- **Function – Enhancements to the ZIDOT and ZASER commands make dumps easier to solve.**
- **Function – Ability to include the system heap in a dump by using a unique token name rather than an adress on the LISTC macro.**
- **Function – A formatted version of the current z/TPFDF SW00SR block is now included in dumps.**

System Error Processing – Customization Enhancements

- **Function – Ability to create different named manual dump definitions that can be used with the ZDUMP command to dump predefined areas of storage via ZIDOT command.**
- **Function – Dump formatting extensions allow a BAL Shared Object (BSO) or C Shared Object (CSO) to be called during dump processing to determine additional areas to be dumped and formatted.**
- **Function – Ability to group programs in dump groups to make it easier to define overrides for programs that require the common areas to be dumped. This allows the customer to associate dump overrides on an application wide basis instead of having to define an override for each program.**

Feature – Dump Enhancements

- **Function – OSA polling during dumps reduces likelihood of lost messages.**
- **Function – Last 10 branch trace entries are formatted.**
- **Function – LDEV trace size can be set dynamically and LDEV trace table can be displayed online**

Feature – Display and Alter Capabilities for C Shared Objects (load modules in 4.1)

- **Function – Ability to display and alter all application programs in memory or on file, including C load modules**
- **Function – ZDMAP has a new ability to find a function, object or address in ANY in-memory (fetched) program.**

In Summary

- **z/TPF Enables Customers to**
  - ✓ Reduce Costs
  - ✓ Effectively Manage a Skilled Resource Base
  - ✓ Improve Time to Market
  - ✓ Develop Competitive Products and Solutions
  - ✓ Gain Market Share
  - ✓ Protect the Current Investment
  - ✓ Position for Business Growth
  - ✓ Ease the Migration Effort

See your Customer Service Representative for more information.

Legal

- IBM, z/Architecture, s/390, z/OS, z/VM, and Websphere are trademarks or registered trademarks of the International Business Machines Corporation in the United States, other countries, or both.
- IBM may have patents or pending patent applications covering subject matter in this presentation.
- 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. All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- LINUX is a registered trademark of Linus Torvalds.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.
- 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.