IBM Software Group

z/TPF Scheduler/Dispatcher Changes

Systems Control Program Subcommittee

Barry Baker
October 2004

AIM Core and Enterprise Solutions
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Topics

- Motivations/Goals/Design Points
- Scheduler/Dispatcher Replacement
  - Externals
  - Migration
- New Dynamic Functionality
  - Externals
  - Migration

Scheduler/Dispatcher aMotivations/Goals/Design Points

- Inability to balance new work in the system across the available I-streams
  - Routing weight based mechanism
  - Weights for each I-stream based on exponentially smoothed utilization calculations
  - Two I-stream weights updated each second
- Problems with this?
  - Assumes a uniform workload (minimal deviation from the average)
  - As the number of I-streams increase, the routing weights become increasingly less accurate at reflecting the impact of dispatching a task on a given I-stream
  - This can result in an unbalanced workload in the system, which will lead to an increase in the response time at high utilization

Scheduler/Dispatcher aMotivations/Goals/Design Points (cont)

- ECBs have an implicit I-stream affinity
  - ECBs stay on the I-stream where they are started unless explicitly moved by application software (SWISC ENTER)
  - Why is this good?
    - Simplifies scheduling of work
    - Some applications must run on a particular I-stream (not designed to take advantage of multiple I-streams)
    - I-stream unique Globals/data
  - Why is this bad?
    - The life of all ECBs is not equal...increasingly diverse workload
    - z/TPF must maintain this **behavior** for existing applications while supporting a more diverse application set

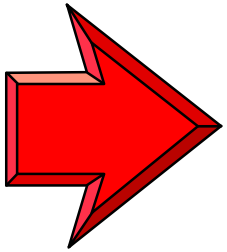
Scheduler/Dispatcher aMotivations/Goals/Design Points (cont)

- More efficient use of the compute resources of a processor
- Minimize response time
- **Replacement** of current routing weight based scheduler with one based on a combination of I-stream shared and unique lists
- **Addition** of dynamic functionality that allows entries to be re-dispatched on any I-stream without the application explicitly having to do this (i.e. SWISC)
- TPF 4.1 Scheduler Design Points
  - Hardware: four I-streams
  - Application: monolithic application set (50K CPU instruction executions, 10 physical DASD I/Os)
- z/TPF Scheduler Design Points
  - Hardware: 32 I-streams (or greater)
  - Application: increasingly diverse application set (compute and I/O bound)

Scheduler/Dispatcher Replacement aExternals

- **Scheduler Replacement**
  - Routine to get the least busy I-stream has been removed
    - ADPC now accepts IS=0

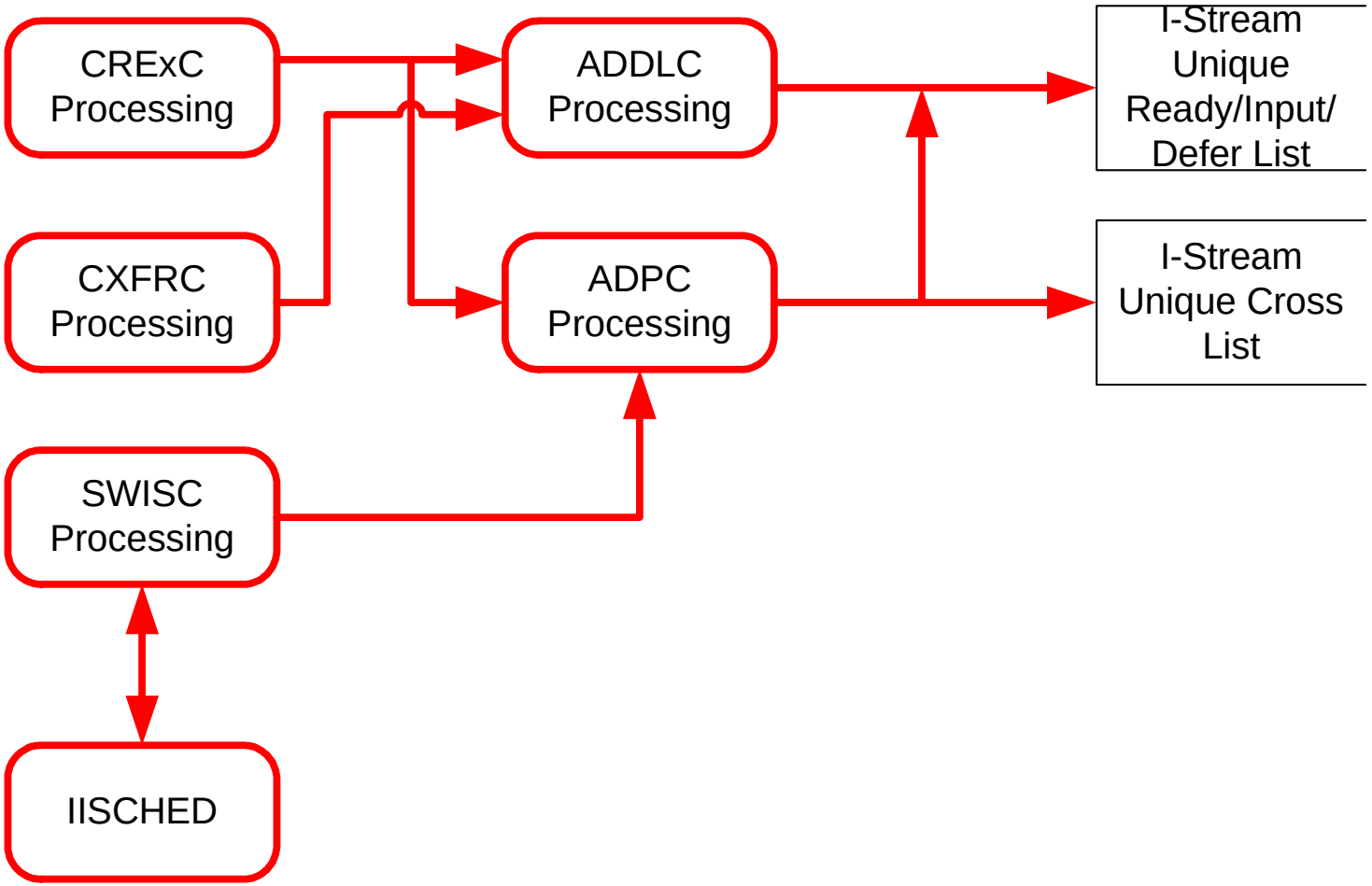
- **TPF 4.1 Dispatcher** (all lists are I-stream unique)
  - Cross List
  - Ready List
  - Input List / OSA Shared List
    - VCT List
    - Suspend List
  - Deferred List



- **z/TPF Dispatcher**
  - I-stream Unique Cross List
  - I-stream Unique Ready List
  - I-stream Shared Ready List
  - I-stream Unique Input List ~~/ OSA Shared List~~
  - I-stream Shared Input List
    - I-stream Unique VCT List
    - I-stream Unique Suspend List
  - I-stream Unique Deferred List
  - I-stream Shared Deferred List

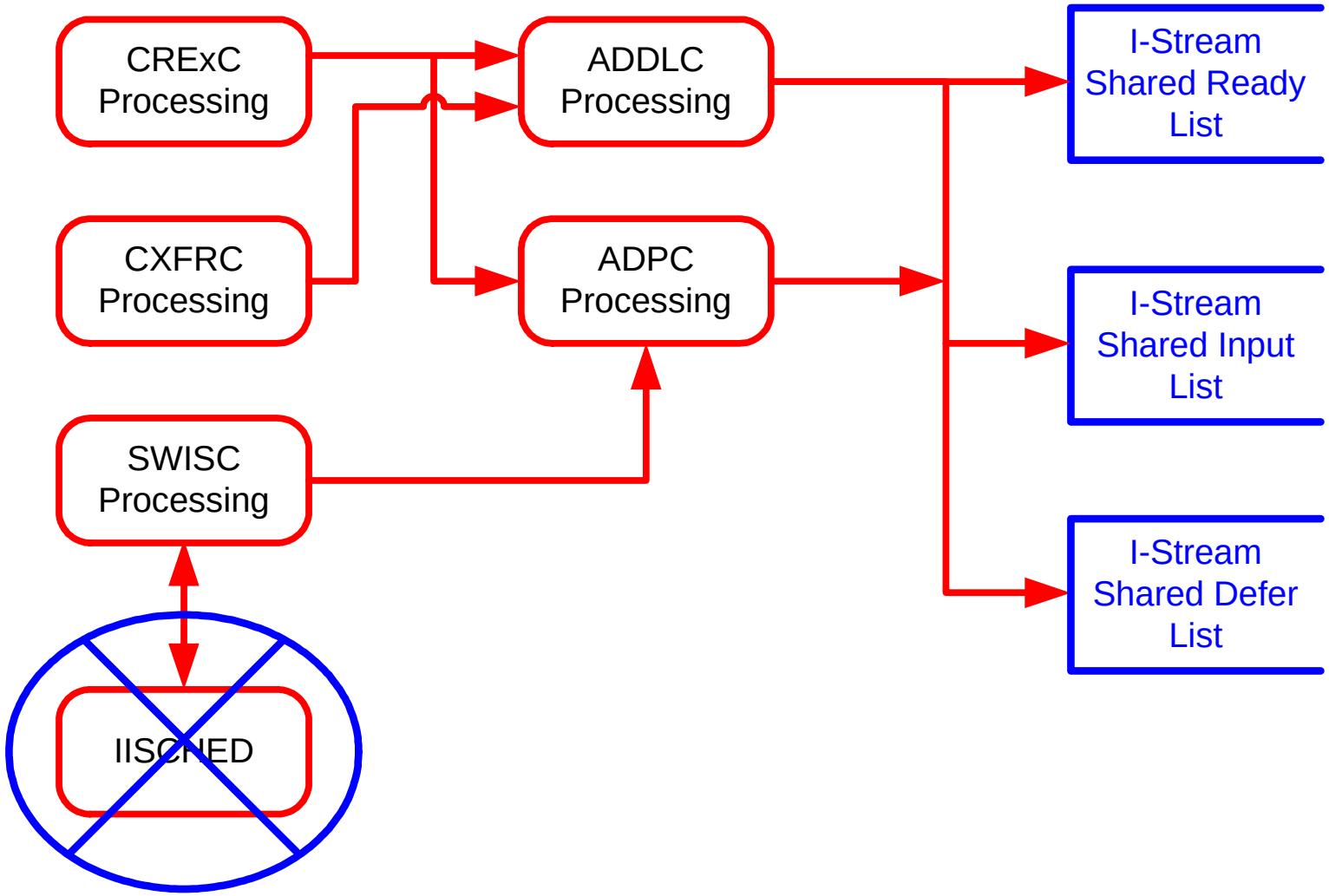
Scheduler/Dispatcher Replacement aExternals(cont)

- Creation of Work in TPF 4.1



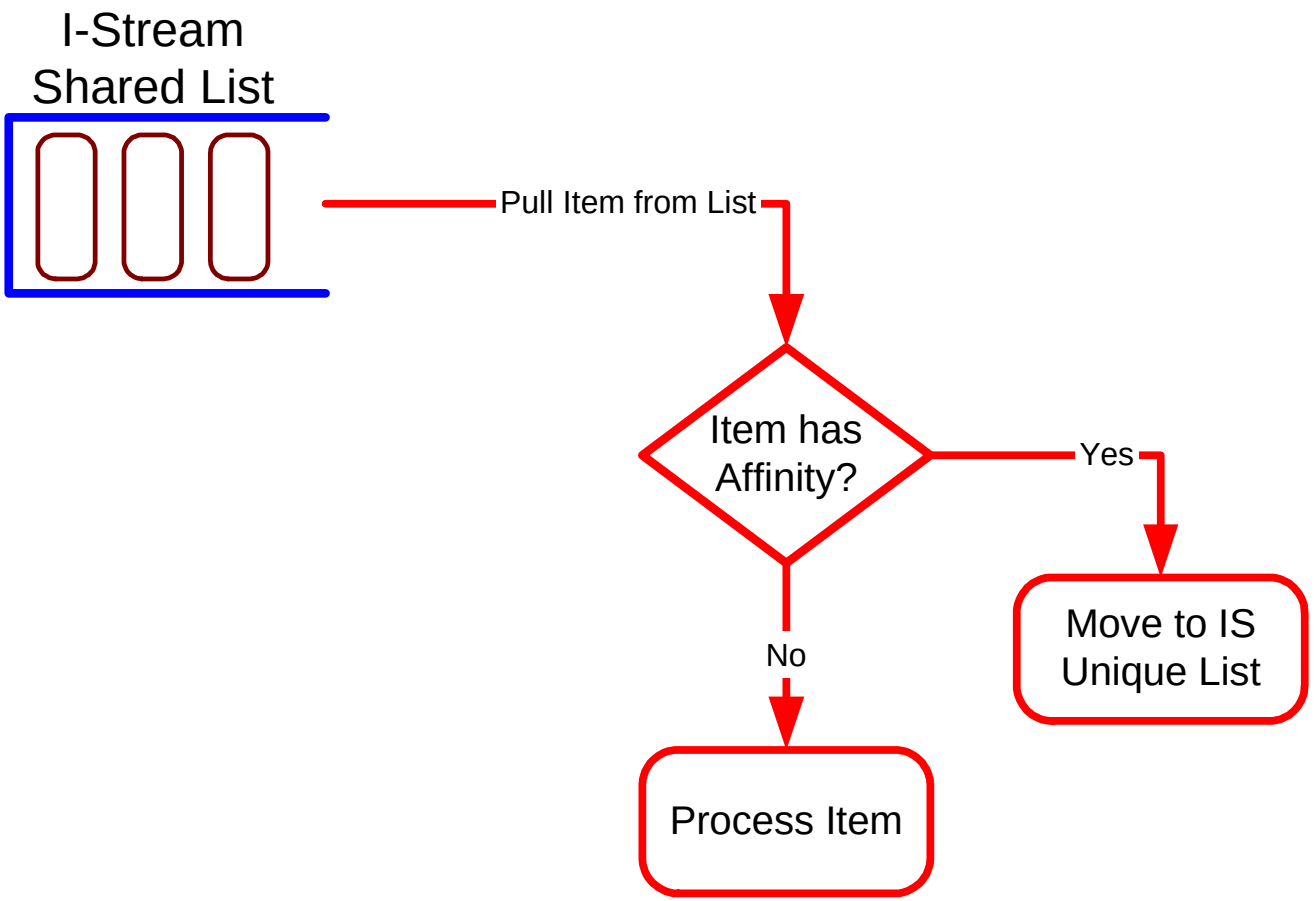
Scheduler/Dispatcher Replacement aExternals(cont)

- Creation of Work in z/TPF



Scheduler/Dispatcher Replacement aExternals(cont)

- Work on the Shared Lists in z/TPF



Scheduler/Dispatcher Replacement aExternals(cont)

```
ZSTAT U
CSMP0097I 18.30.41 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0016I 18.30.41 SYSTEM UTILIZATION DISPLAY
NUM  ADR  UTIL/ ADJ  CROSS READY INPUT   VCT SUSPD DEFER ACT-ECB
IS- 1 00    .1/   .1     0     0     0     0     0     0       1
IS- 2 01    .0/   .0     0     0     0     0     0     0       1 _
IS- 3 02    .0/   .0     0     0     0     0     0     0       0
IS- 4 03    .0/   .0     0     0     0     0     0     0       0
IS- 5 04    .0/   .0     0     0     0     0     0     0       0 _
IS- 6 05    .0/   .0     0     0     0     0     0     0       0
IS- 7 06    .0/   .0     0     0     0     0     0     0       0
IS- 8 07    .0/   .0     0     0     0     0     0     0       0 _
SHARED                         2     0                 0       +
```

Scheduler/Dispatcher Replacement aMigration

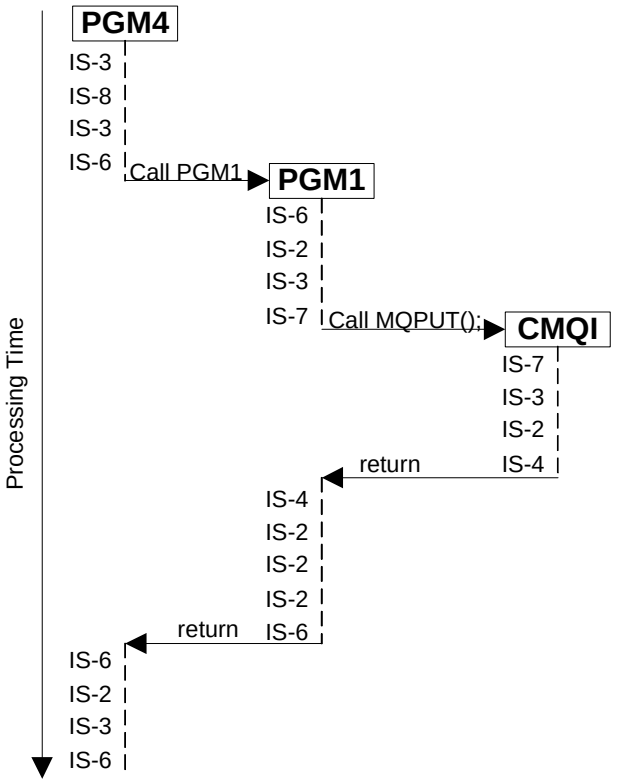
- ADPC now accepts IS=0
  - CP routine IISCHED/CCIISC(CCE7) removed
    - Update calling CP code
      - If the value returned by IISCHED was then passed to ADPC, then remove the IISCHED call and pass a zero to ADPC instead

New Dynamic Functionality aExternals

- ECBs, at execution time, can be defined as having an I-stream affinity of either NONE, PROGRAM,or ECB
  - **NONE**: An ECB processing in a program defined as having an affinity of NONE will dynamically switch I-streams when appropriate
  - **PROGRAM**: An ECB, processing in a program defined as having an affinity of PROGRAM, will only be processed on the I-stream where the ECB was processing when the program was first entered
  - **ECB**: An ECB, processing in a program defined as having an affinity of ECB and all subsequent programs, will only be processed on the I-stream where the ECB was processing when the program was first entered (note: only allowed on EISAC/tpf_eisac())
- Programs can specify in their PAT entry their I-stream affinity characteristic
  - Macro/C API (EISAC/tpf_eisac()) provided to allow an application to override what is defined in their PAT entry

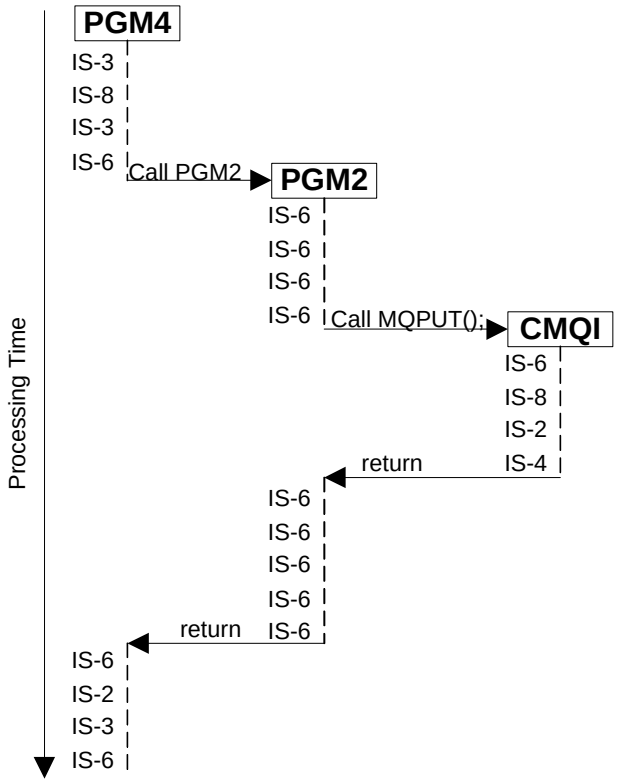
New Dynamic Functionality aExternals (cont)

- Affinity = NONE
  - PGM4, PGM1, and CMQI are defined as having an affinity of NONE
  - As an ECB follows this processing flow, it MAY be processed by any I-stream after giving up control(DASD I/O, Events, DLAYC,...)



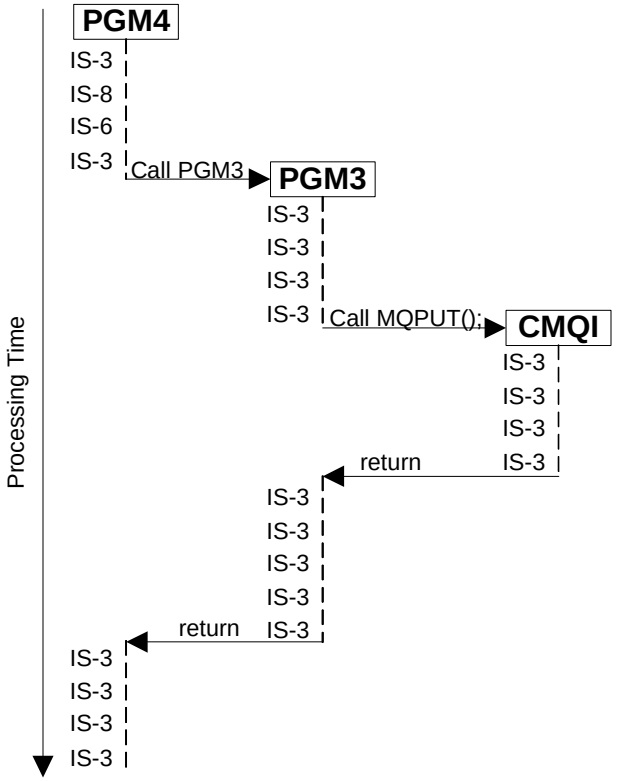
New Dynamic Functionality aExternals (cont)

- Affinity = PROGRAM
  - PGM4 and CMQI are defined as having an affinity of NONE
  - PGM2 is defined as having an affinity of PROGRAM
  - As an ECB follows this processing flow, it MAY be processed by any I-stream after giving up control only while processing in programs PGM4 and CMQI
  - An ECB processing in PGM2 will only be processed on the I-stream where the ECB was processing when PGM2 was first entered



New Dynamic Functionality aExternals (cont)

- Affinity = ECB
  - PGM4 and CMQI are defined as having an affinity of NONE
  - PGM3 is defined as having an affinity of ECB
  - As an ECB follows this processing flow, it MAY be processed by any I-stream after giving up control only while processing in program PGM4
  - An ECB processing in PGM3, and all subsequent programs, will only be processed on the I-stream where the ECB was processing when PGM3 was first entered



New Dynamic Functionality aExternals (cont)

- Programs that are either defined as having an affinity of NONE or portions of code that are designated as having an affinity of NONE with tpf_eisac/EISAC can take advantage of the new functionality.
  - When the following services are used and the ECB gives up control of the I-stream, it MAY resume processing on a different I-stream.
    - DASD I/O, EVNTC, ENQC, CORHC, DLAYC, DEFRC, YIELDC

New Dynamic Functionality aMigration

- No migration needed to maintain TPF 4.1 behavior and avoid new function
- New Function Migration Considerations
  - Small amount of workload using new functionality will yield majority of the benefit
  - Application Considerations
    - I-stream unique data usage
    - High usage of system services that give up control of I-stream
- TPF Shipped Code
  - Working to define TPF code to use new functionality and relieve the customer from having to do anything to receive the majority of the benefit.
  - Considerations: Same as above, plus achieving a broad mix to satisfy customers' TPF usage

Legal

- IBM is registered trademark of International Business Machines, Inc.
- Other company, product, or service names may be trademarks or service marks of others.