z/TPFDF V1.1

TPF Users Group - Spring 2009

z/TPFDF Status Update

Jason Keenaghan
Database Subcommittee

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

Any reference 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.

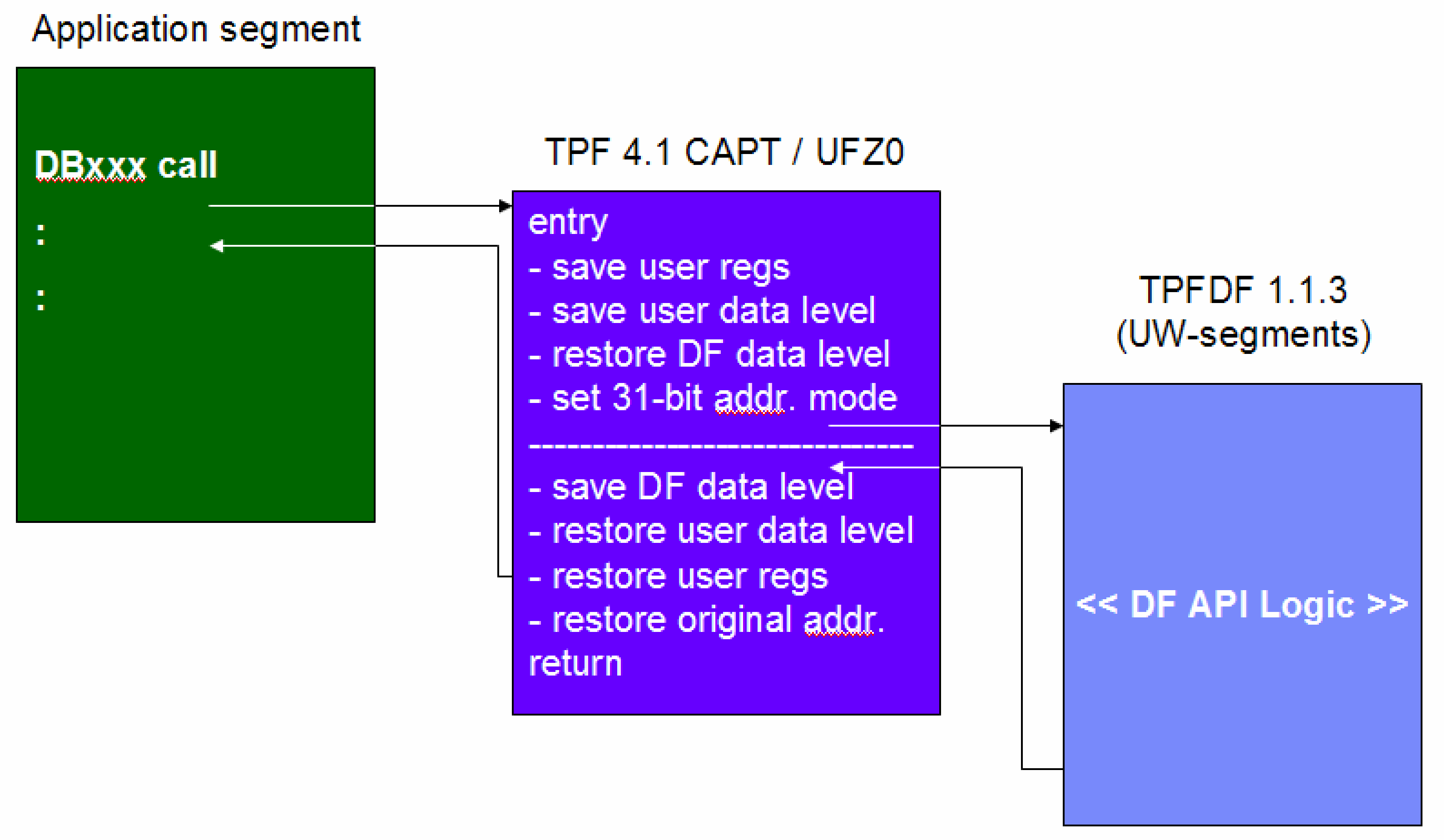
Agenda

- **Improving z/TPFDF performance**
  - TPFDF 1.1.3 linkage vs. z/TPFDF linkage
  - New z/TPFDF linkage
- **z/TPFDF macro breakpoint support for TPF Debugger**
- **TPFUG requirements update**

Linking to the TPFDF central database routines

- **Certain TPFDF macros (DBxxx) and C functions (dfxxx) are processed by the TPFDF central database routines**
- **TPFDF 1.1.3 provided support for a “fast link” mechanism to transfer processing from application code to the central database routines**
- **The fast link mechanism was developed in order to minimize the path length associated with linking to the TPFDF API routines, which can be called very frequently by applications**
- **Some side effects existed due to the use of this unique fast link mechanism:**
  - Special E-type loader considerations
  - No ENTER trace capabilities
  - Limited TPF debugger and performance analyzer capabilities

TPFDF 1.1.3 fast linkage



Application segment

DBxxx call
:
:

TPF 4.1 CAPT / UFZ0

entry
- save user regs
- save user data level
- restore DF data level
- set 31-bit addr. mode

- save DF data level
- restore user data level
- restore user regs
- restore original addr.
return

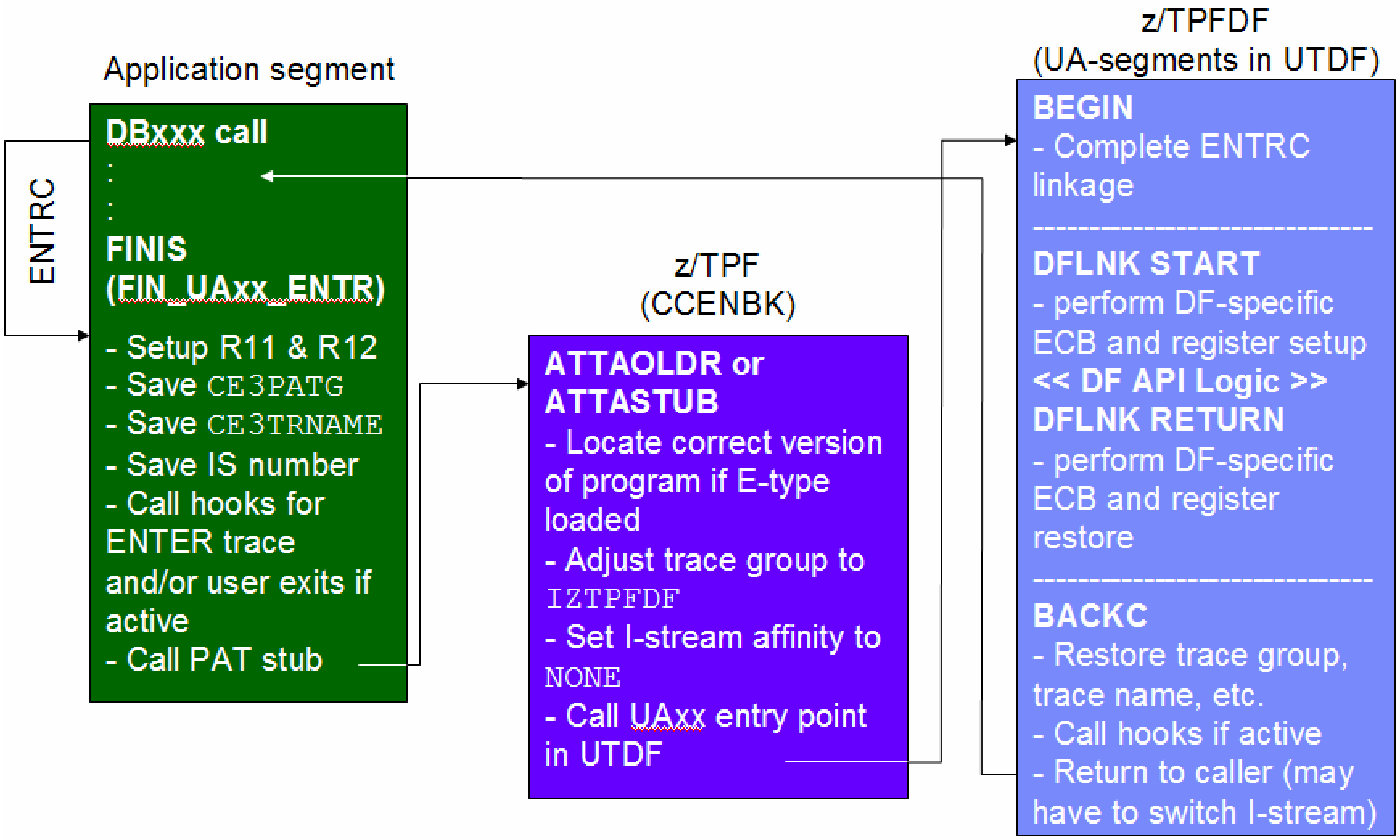
TPFDF 1.1.3
(UW-segments)

<< DF API Logic >>

Changing the linkage model with z/TPFDF

- **z/TPF greatly enhanced the functionality available with the standard Enter/Back linkage and program attributes in the PAT**
  - Diagnostics: Trace groups, enhanced macro trace
  - Runtime: I-stream affinity
- **During z/TPFDF development, the design decision was made to drop the TPFDF-unique fast linkage in favor of using standard Enter/Back linkage**
  - Take advantage of new Enter/Back functionality
  - Allows for seamless integration with the TPF debugger
  - Remove special E-type loader support to help with product maintenance

z/TPFDF standard Enter/Back linkage



Application segment

DBxxx call
:
:
FINIS
(FIN_UAxx_ENTR)

- Setup R11 & R12
- Save `CE3PATG`
- Save `CE3TRNAME`
- Save IS number
- Call hooks for ENTER trace and/or user exits if active
- Call PAT stub

z/TPF
(CCENBK)

ATTAOLDR or ATTASTUB

- Locate correct version of program if E-type loaded
- Adjust trace group to `IZTPFDF`
- Set I-stream affinity to `NONE`
- Call UAxx entry point in UTDF

z/TPFDF
(UA-segments in UTDF)

BEGIN

- Complete ENTRC linkage

DFLNK START

- perform DF-specific ECB and register setup

<< DF API Logic >>

DFLNK RETURN

- perform DF-specific ECB and register restore

BACKC

- Restore trace group, trace name, etc.
- Call hooks if active
- Return to caller (may have to switch I-stream)

Performance implications of new z/TPFDF linkage

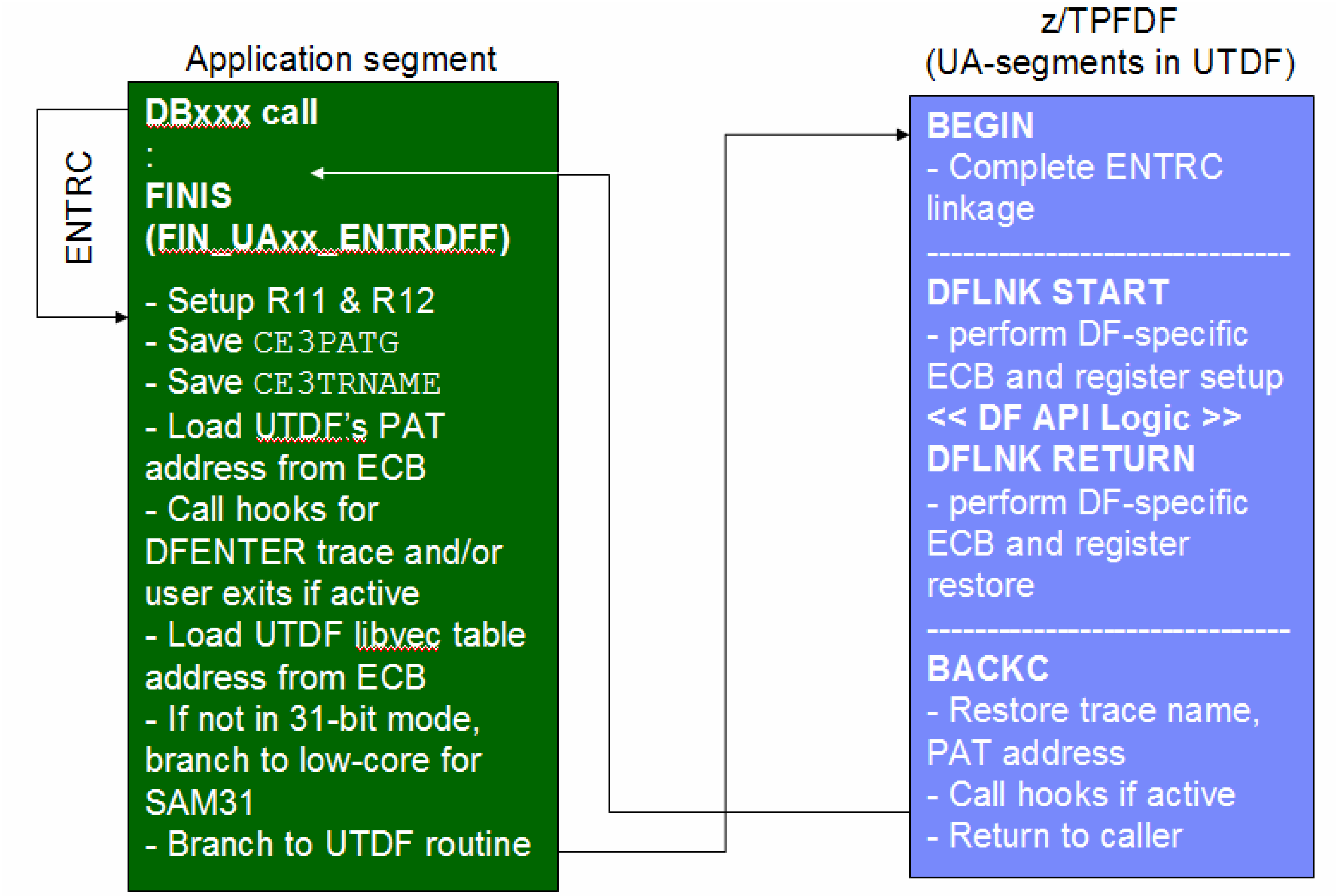
- **Measuring the performance impact of switching from TPFDF fast linkage to z/TPF Enter/Back linkage**
  - Path length associated with a single z/TPFDF API call was in line with the expected results from original z/TPFDF design
  - Increased functional enhancements were believed to greatly outweigh the modest path length increases
- **Since z/TPFDF was made generally available in 2005, IBM has invested in the development of a z/TPFDF-specific test driver that more accurately reflects real-life TPFDF application API profiles**
  - Test results indicated the overall application path length was more greatly affected by the new linkage than original estimations predicted
  - TPF Development Lab has begun an effort to modify the z/TPFDF linkage model in order to find a “middle ground” between performance and functionality

Align z/TPFDF performance with TPF 1.1.3 performance

- **Remove low-benefit functionality**
  - No unique `IZTPFDF` trace group, use trace group of application
  - Change IS-affinity to `PROGRAM` from `NONE`
- **Make some functionality optional**
  - New z/TPFDF Enter/Back trace that can be separately enabled/disabled
  - New z/TPFDF function call, function entry, and function return user exits can be separately enabled/disabled
- **Continue to support other functionality**
  - Standard E-type loader support
  - Standard support for z/TPF debugger

Disclaimer: *This information represents design concepts only for possible additions to the z/TPF and z/TPFDF products. The information is provided for planning purposes only. IBM reserves the right to change these 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.*

New z/TPFDF fast linkage



Application segment

ENTRC

DBxxx call

:

FINIS

(FIN_UAxx_ENTRDFF)

- Setup R11 & R12
- Save CE3PATG
- Save CE3TRNAME
- Load UTDF's PAT address from ECB
- Call hooks for DFENTER trace and/or user exits if active
- Load UTDF libvec table address from ECB
- If not in 31-bit mode, branch to low-core for SAM31
- Branch to UTDF routine

z/TPFDF

(UA-segments in UTDF)

BEGIN

- Complete ENTRC linkage

DFLNK START

- perform DF-specific ECB and register setup

<< DF API Logic >>

DFLNK RETURN

- perform DF-specific ECB and register restore

BACKC

- Restore trace name, PAT address
- Call hooks if active
- Return to caller

Migrating to the new z/TPFDF linkage

- **Testing of the new support is still in progress**
- **Coexistence and migration considerations:**
  - The “old” z/TPFDF linkage will continue to be supported to ease migration
  - In order to take advantage of the new z/TPFDF linkage, you *must* reassemble *.asm applications that call z/TPFDF macros; *no* recompiles are needed for *.c and *.cpp applications
  - You *must* reassemble *all* callers of CROSC
  - New control program user exits called out of z/TPFDF linkage should be investigated and updated as necessary
- **The new linkage will be provided via 2 co-requisite APARs*:**
  - z/TPF: PJ35509
  - z/TPFDF: PK79078

***** *These APARS are currently scheduled to be available on z/TPF PUT 6 and z/TPFDF PUT 6***

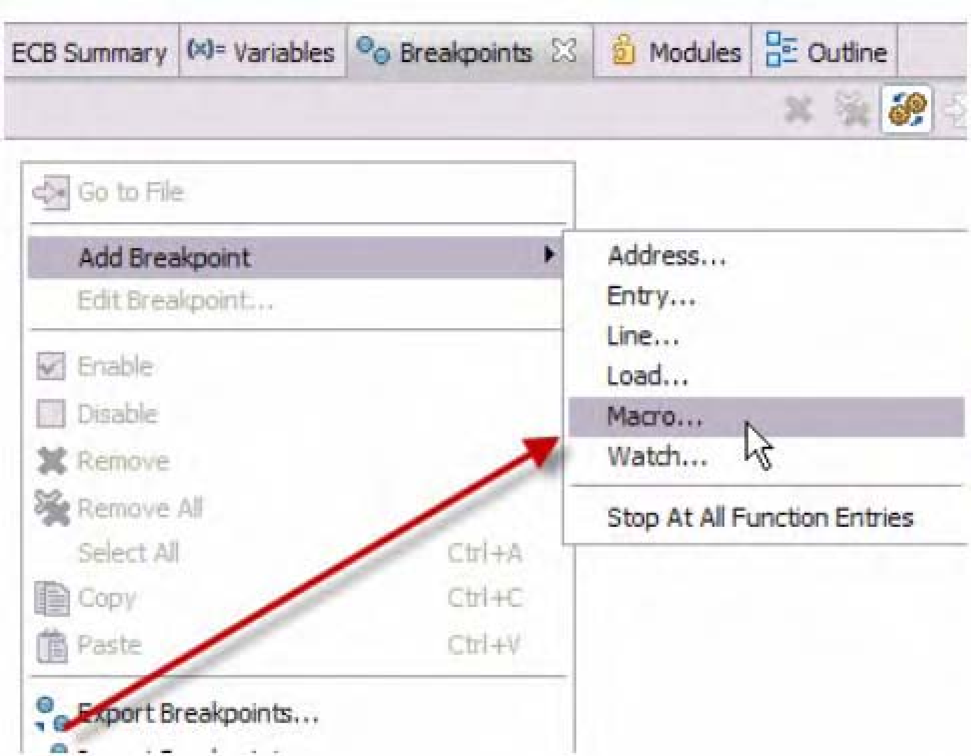
***** *Please check with your customer support representative to determine when these APARs are available***

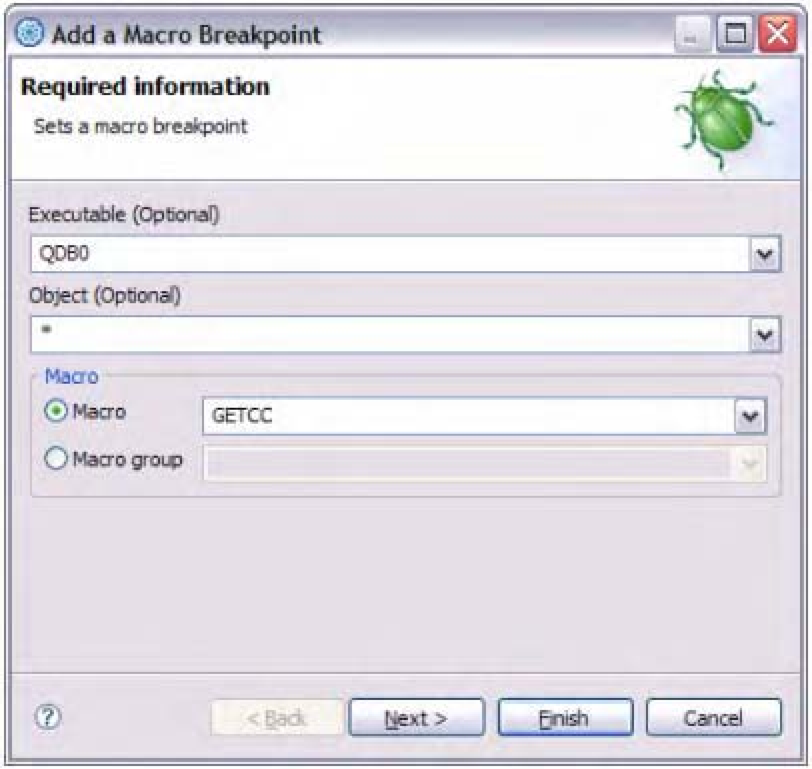
Agenda

- **Improving z/TPFDF performance**
  - TPFDF 1.1.3 linkage vs. z/TPFDF linkage
  - New z/TPFDF linkage
- **z/TPFDF macro breakpoint support for TPF Debugger**
- **TPFUG requirements update**

Macro Breakpoint Add

- **Macro Breakpoints now have their own dialog box which is available by right clicking in the breakpoint view (choosing Entry breakpoint and Defer are no longer required)**




Add a Macro Breakpoint

Required information

Sets a macro breakpoint

Executable (Optional): QDB0

Object (Optional): *

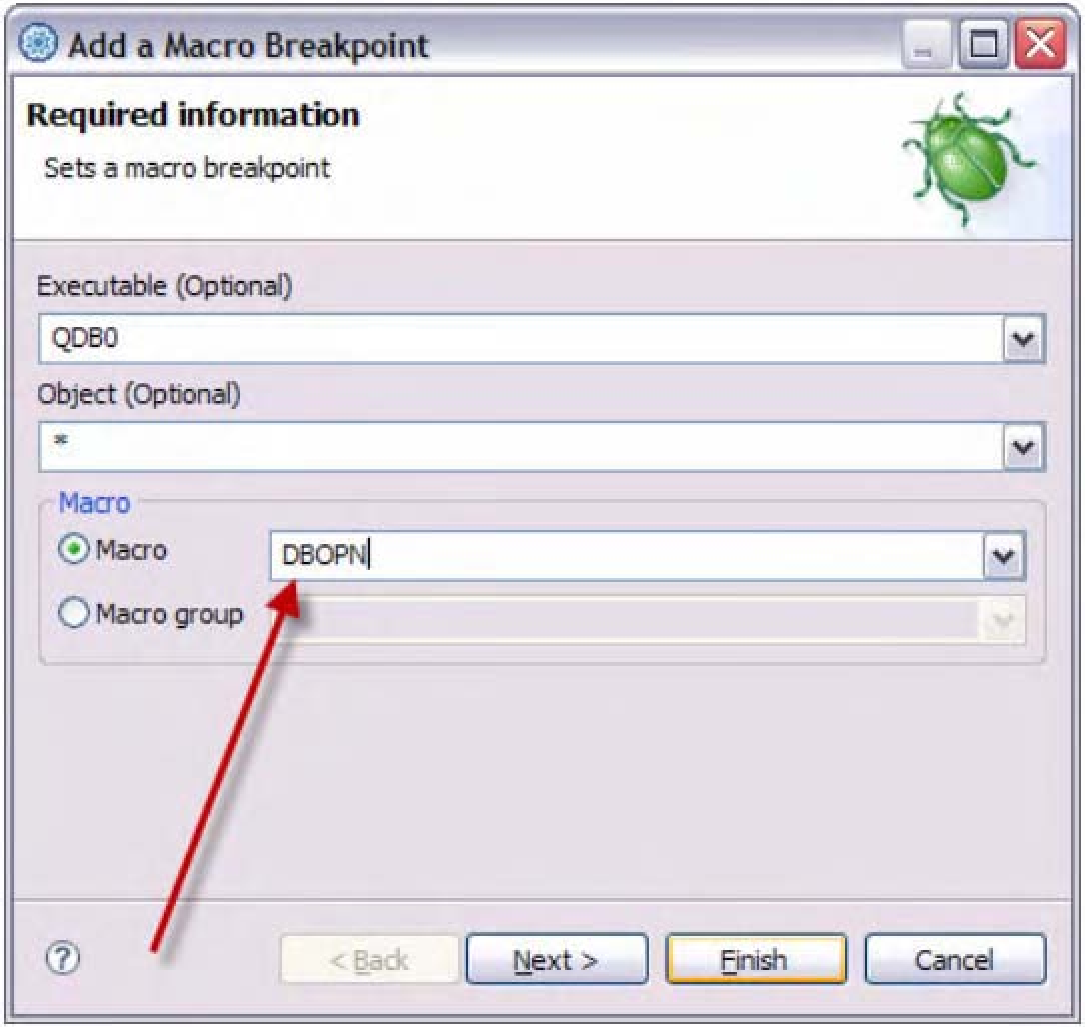
Macro

● Macro: GETCC

○ Macro group

TPFDF Macro Breakpoints

- **z/TPFDF Macro Names can now be entered through the Macro Breakpoint pane (e.g., DBRED, DBOPN, etc)**



Add a Macro Breakpoint

Required information

Sets a macro breakpoint

Executable (Optional): QDB0

Object (Optional): *

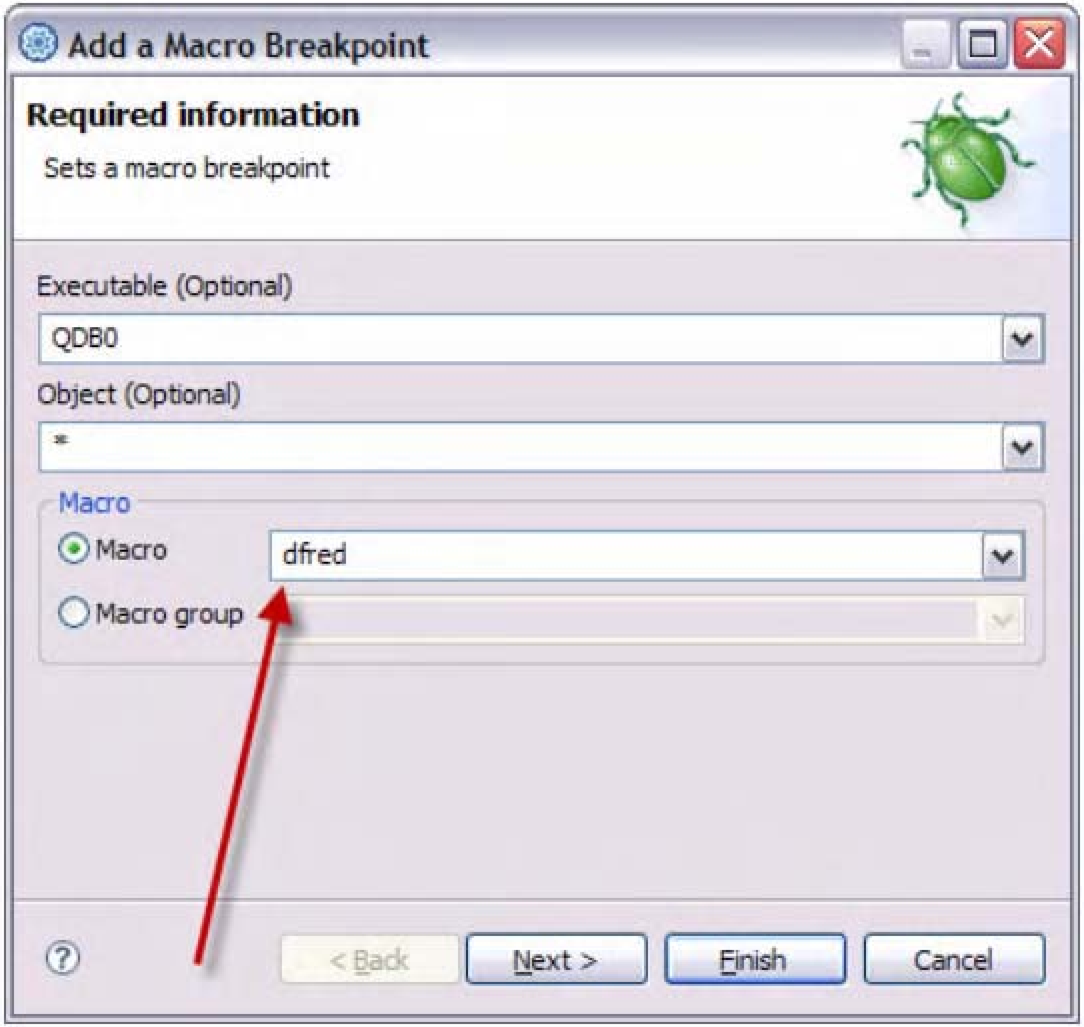
Macro

● Macro: DBOPN

○ Macro group

TPFDF Macro Breakpoints

- **z/TPFDF C/C++ functions can now be entered through the Macro Breakpoint pane (e.g., dfred, dfopn, etc)**



Add a Macro Breakpoint

Required information

Sets a macro breakpoint

Executable (Optional): QDB0

Object (Optional): *

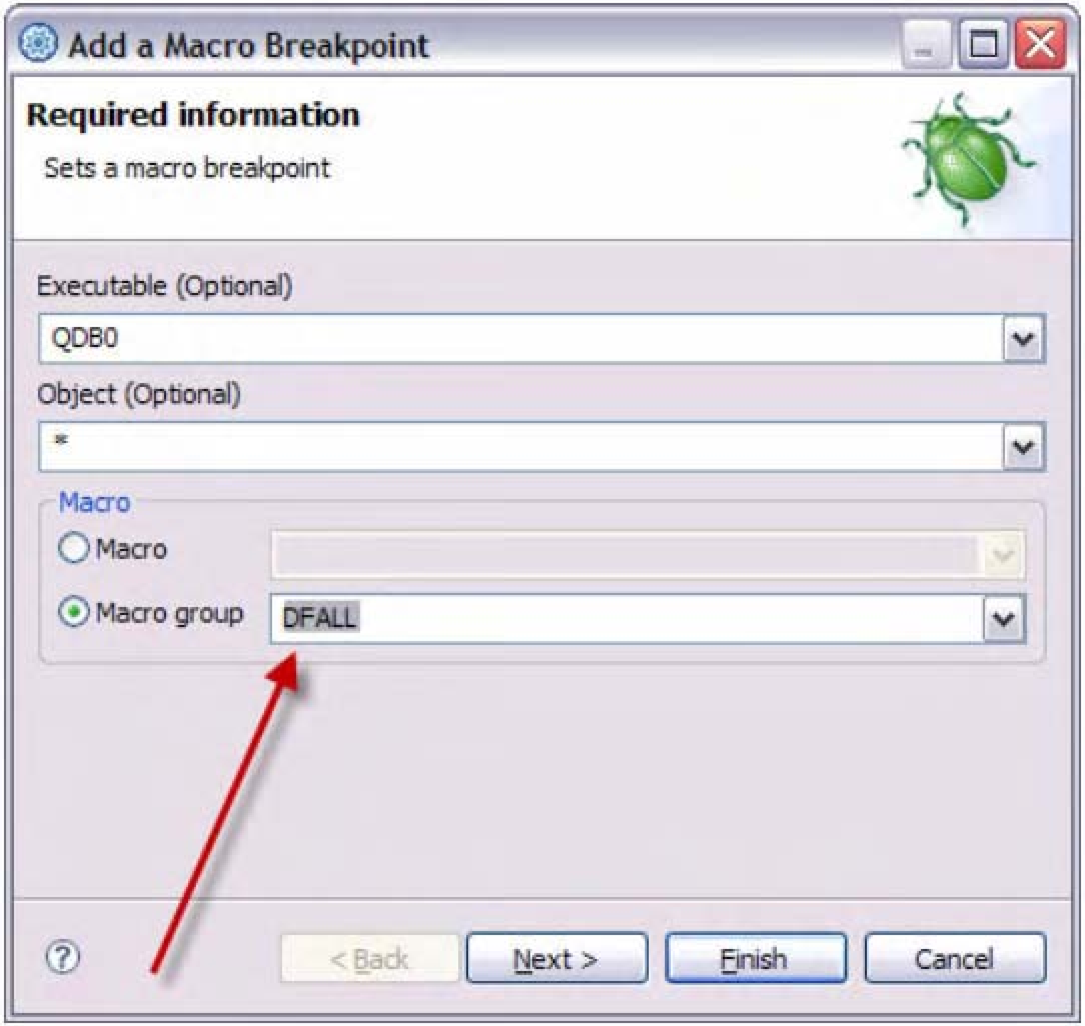
Macro

● Macro: dfred

○ Macro group

DFALL TPFDF Macro Group

- **The DFALL Macro Group will stop the application when any z/TPFDF Macro or C/C++ function is executed by the application**



Add a Macro Breakpoint

Required information

Sets a macro breakpoint

Executable (Optional): QDB0

Object (Optional): *

Macro

○ Macro

● Macro group: DFALL

z/TPF Debugger Deliverable Details

Description	z/TPF APAR	TPF Toolkit Level	TPUG Requirement
Add Macro Breakpoint	PJ35430 PUT6	V3.4.2	V08061S
TPFDF Macro Breakpoints DFALL TPFDF Macro Group	PJ35669 PUT6	None	V08055S

Agenda

- **Improving z/TPFDF performance**
  - TPFDF 1.1.3 linkage vs. z/TPFDF linkage
  - New z/TPFDF linkage
- **z/TPFDF macro breakpoint support for TPF Debugger**
- **TPFUG requirements update**

“Accepted” TPFUG Requirement now “Not Likely”

- **DF00079 – Support memory based z/TPFDF files**
  - DBDEF will specify if a file is memory based
  - DBOPN will allow either:
    - a new instance of the file to be created in memory, or
    - an existing instance of the file to be accessed
  - DBCLS will allow either the instance to be deleted, or simply released allowing other ECBs access

Top 5 TPFDF requirements from Fall 2008 ballot

Rank	Status	Req. Number	Req. Title	Avg. Priority
1	New	DF08191F	Improved TPF/DF Error Checking	13.93
2	Likely	DF08188S	TPFDF CRUISE Provides Additional TARGET Options for Capture	8.57
3	New	DF08190F	TPF/DF to Utilize User Defined Record IDs for GETFCs	8.21
3	Likely	DF00152	TPFDF C API Improvements	8.21
5	Likely	DF08189S	RELFC API for TPFDF	7.14

Trademarks

- IBM is a trademark of International Business Machines 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.