It was a quandary. Me, alone, with just a hunch and a hatful of memories to work with. I'd seen it all before of course but that had been then and this was very much now – how could I, Qwik (as in 'w' for witted) Tracy, make progress in the case without a solid lead? I was resigned to another night without a breakpoint when, suddenly, out of the rain stepped a solitary figure and I knew, just knew, I was finally on to something…

xTrace MARKS THE SPOT

It was June and my partner Meisner IT agreed that we should work together to solve the case. He was taller than me, and Danish, so I wasn't going to argue over spilt milk and cookies. And he had a new trick up his sleeve…

xTrace is an alternative diagram tracing tool for CA Gen generated applications on the z/OS (MVS) platform. The combination of an intuitive GUI interface, powerful breakpoint definition, call stack overview, performance analysis and low execution overhead means that xTrace greatly speeds up the testing and debugging of z/OS applications.

X Allows you to trace native batch & CICS applications without needing to generate for TSO or the workstation. Instead of wasting time poring over code or manually editing COBOL to add DISPLAY statements, developers can immediately trace through the code to locate problems. Support for IMS is planned for a future release.

X To help with tracing large or complex applications, you can set breakpoints based on action diagrams, statement numbers or even specific data values. Rather than having to turn off trace for every intermediate action block, you simply set the breakpoints and execute until the breakpoint condition is reached.

X 'Call stack' feature allows you to view and save the complete call stack (code pathways) without needing to trace through each action block individually.

X Brings to the z/OS platform similar functionality to the new Gen r7.6 Diagram Trace Utility that is available for non-mainframe platforms, but with even more flexible and powerful breakpoint technology and multi-user capability.
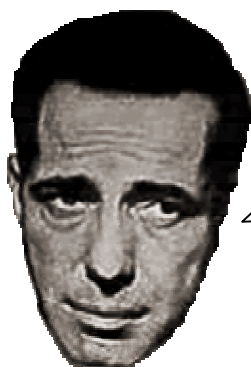
> That trace I placed was sure effective. Customers reported a 4-fold improvement in debugging complex applications when using xTrace.

REACHING BREAKPOINTS

My investigations revealed more information about the xTrace breakpoints – a place or condition within your code where you want to pause execution.

Instead of stepping through most of your application before arriving at the point of interest, simply define a breakpoint and run the application until the breakpoint condition is satisfied.

xTrace supports various types of breakpoints:

Action Block Breakpoint. If you know which action block you want to trace, simply add an action block breakpoint and tracing will start when that action block is first entered.

Value Breakpoint. Use a value breakpoint to start tracing when an attribute is assigned a certain value. For example CODE = -20. A very powerful feature of value breakpoints is the ability to use wild cards, for example *.*.*.WORK.CODE would refer to any view of an attribute code in any action diagram

Entry/Exit Breakpoint. You can stop on every entry and/or exit from an action block.

Statement Breakpoint. When you need to stop every time you reach a certain statement in an action block.

xTrace even allows you to trace code specific to one model – useful if you are using CBD techniques.

POETRY CORNER

An Ode to xTrace

I am a little breakpoint
Nothing do I bear
Except a little pointer
To there, and there and there

I stop when I am asked to
I go when I am pressed
My aim is to make your testing
Easier and less depressed

You can activate me simply
You can let me be involved
For then I know I'll help you
ensure your problems are resolved

So that, I think, is that then
I believe I've made my case:
If you want to do debugging
Look no further than xTrace
(for z/OS anyway)

By EJ Tracy (14 ½) with apologies to
Mother Goose

The Guru
releasing calm into CM...

A customer writes…

Mr Guru Sir

Is there an easy way within GuardIEn of generating the entire contents of a model if I ever decide I need to do that?

You are very kind

Det. Marlowe of Downtown, LA

Guru responds…

Dear Det. Marlowe

It is good to hear from you but save the formalities for greater mortals than myself. As Thoreau once said 'Our life is frittered away by detail…simplify, simplify'. So a simple 'Sir' will suffice.

Anyway. I digress. To answer your question – this IS available using the **Scope Wizard** on the Development Update Content screen via Tools->Scope Wizard.

This allows you to scope the entire contents of the models selected within the DU via the 'All build objects in the target Model(s)' option. I'm not sure how many objects you will have to rebuild but you may have to keep an eye on the limitations here…for details please see the 'System Update Limits' for the HE in the online help and documentation (e.g. 9999 ABs in a single expansion, 5000 in a single migrate).

I trust that will allow you to generate with great joy.

Guru

Guru thanks everyone for their questions, but reserves the right to ignore them completely.

SPOTip

If you wish to enforce the scoping of objects for modify/delete against the correct models you can use the SUBROLES project property. The format of the property is **AAAA!DDDD** where **AAAA** is the list of linkage roles that are allowed to be Modify/Delete and **DDDD** is the list of roles that cannot be Modify/Delete. Roles not listed are downgraded to Access if they are scoped as Modify/Delete but can be changed to Modify/Delete with a warning confirmation.

For example, setting the property to **MC!G** would allow linkages with roles of M or C to have M/D protection, linkages with role G would not be allowed M/D protection and others would be set to A but could be upgraded to M/D.

Warning: One or more objects have linkage roles that are not allowed to be Delete/Modify - they have been reset to Access
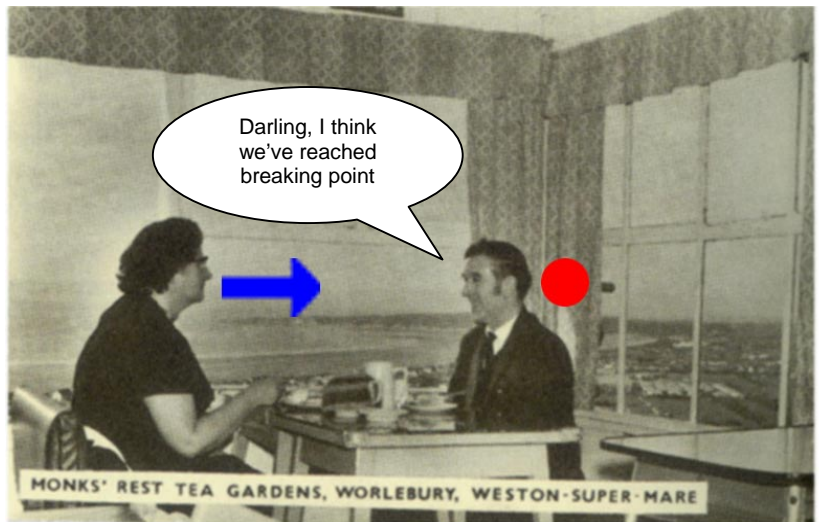
OK

A useful way, particularly if you have multiple development models or use CBD techniques, of ensuring that only the correct models are used for update.

---

xTrace WEBCASTS A SUCCESS

IET recently held 4 well attended and highly successful webcasts to demonstrate how xTrace greatly simplifies and speeds up debugging and tracing code on z/OS. The feedback received was very positive and most of the attendees have now requested an evaluation copy of the product.

If you missed out on these sessions but would still like to see a demonstration or obtain an evaluation copy of the product then please contact us then we would be delighted to hear from you.

Darling, I think we've reached breaking point

MONKS' REST TEA GARDENS, WORLEBURY, WESTON-SUPER-MARE

Ah well. Another edition over - hope you traced a clear and precise path and that maybe you found something useful. Let us know if you have any ideas for future articles. Byeee.

The IET Team

To ensure you don't miss future copies of SPOTLIET why not send us a blank email at spotliet-subscribe@iet.co.uk

---