What you need to know about the IBM z/OS Debugger V14

Hans Emrich
Senior Client IT Professional
PD Tools + Rational on System z
Technical Sales and Solutions
IBM Systems

hans.emrich@de.ibm.com
IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here.
Agenda: What you need to know …

The name, the version, the product number(s):

IBM z/OS Debugger V14
- IBM Developer for z Systems EE V14 5655-AC5
- IBM Developer for z Systems V14 5724-T07
- IBM Debug for z Systems V14 5655-Q50
- IBM ADFz V3.0 5655-AC6

by—the—way: End of support for IBM Debug Tool V12 => 28.02.2018

Packaging and technical strategy
ADFz, IDz and … ?!??

Ways to the plugins

New features and functions
Application Delivery Foundation for z Systems V3.0

- Application Delivery Foundation for z Systems is a bundle of individual products. Individual products can be purchased separately as well.

- The same z/OS debug technology is included in Developer for z Systems, Developer for z Systems Enterprise Edition and Debug for z Systems.

- Debug for z Systems includes a 3270 interface for debugging which is NOT available with Developer for z Systems.

- Developer for z Systems Enterprise Edition includes Debug for z Systems and 1 authorized user license of Developer for z Systems per Value Unit.

- Everything except IBM Developer for z Systems V14.0 are available in Shop z.

- IBM Developer for z Systems V14.0 is continue to be available in Passport Advantage and allows no cost upgrade from RDz V9.5.
IBM z/OS Debugger is IBM’s strategic z/OS debugging solution

All products ship the same zDebug technology

Components of zDebug are packaged differently:
- Engine: the main debug engine
- 3270 interface: enables usage under ISPF for unlimited number of users
- Debug GUI: light weight debug-only GUI for z/OS
- Explorer for z/OS (unlimited number of users permitted)
- Utilities: Load Module Analyzer, Code Coverage Utility and debug profile
And where do i get the Plug-Ins now ?
Tools for Mainframe Development

IBM Explorer for z/OS Aqua offers an integration platform and a single common way to obtain a compatible and integrated set of Eclipse tools to allow system programmers and developers to be productive. Platforms and plug-ins based on IBM Explorer for z/OS Aqua have been built and tested to install side by side taking the guesswork out of choosing which plug-in will install into which platform. Hover over a platform to learn which plug-ins can be installed into which platforms. Click any offering to learn more and see additional download options.

Plug-ins

- Development Assistant
- Performance Analyzer
- Configuration Manager
- Application Performance Analyzer
- Audit Manager
- Troubleshooting Utilities
- IBM Application Delivery
- z/OS Connect
- RACF Explorer
- MOVE Explorer
- z/OS Explorer
- Eclipse

Platforms

Download Eclipse Tools
Download Host Components
Using IBM Installation Manager (IM)

Please click on an option listed below.

Option 1 — Starting from scratch
Option 2 — Installation Manager with Aqua and RDz is installed
Option 3 — Installation Manager with Aqua is installed
Option 4 — Installation Manager is installed
Option 5 — Deploying to machines with no Internet access

Further Installation Manager information

Further help and assistance can be found in this blog Deploying IBM z/OS Explorer Aqua using IBM Installation Manager

Using Eclipse (p2)

Please click on an option listed below.

Option 1 — Starting from scratch
Option 2 — Aqua is installed
Option 3 — Eclipse is installed
Option 4 — Deploying to machines with no Internet access

[Click on the bar of your choice]

[Scroll down]

Where to find
Last but not least

- Be aware: Many searches in internet may result on pages for IBM Debug Tool for z/OS **not** to IBM z/OS Debugger

- Use the IBM Mainframe Development Center
  https://developer.ibm.com/mainframe/
  (Google search: *mainframe devops*)

- Use Knowledge Center for IDz V14 or “z/OS Debugger”
New with the debugger in IBM Developer for z System EE

- Full support for the debug Connection Manager
- Two debug modes
  - compatibility mode
  - standard mode
- Launchers versus DTSP/DTCN
- Visual Debugging feature
- Paragraph Breakpoints functions
  - Stack Pattern Breakpoints
  - Source Level Breakpoints (Deferred Breakpoints)
- Code Coverage Improvements
- Support for z/OS 64-bit applications (PL/I, C/C++, HLASM)
Connection Manager support

Both the Remote System Explorer and Debug Manager are required for DBM or DBMDT TEST parameters

Remote System Explorer (RSEDx) → z/OS

Debug Manager (DBGMBR) → Application (zDebug)
## Standard vs. Debug Tool Compatibility Mode

- **Two debug modes**
  - **compatibility mode**
    - the debug engine work as the former Debug Tool for z/OS to perform all host debugging tasks
    - supports all LE conforming COBOL, PL/, C/C++ and HLAAM versions (except 64bit)
    - plus non-LE conforming versions
  - **standard mode**
    - uses the Program Information and Control Library (PICL) engine technology which performs more internal work in the client machine
    - supports only „Enterprise-Level“ compiler versions and HLAAM inclusive 64bit
    - supports more new features and functions

<table>
<thead>
<tr>
<th>Feature</th>
<th>Standard mode</th>
<th>Debug Tool Compatibility mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Runtime Option -</td>
<td>DBM.userid</td>
<td>DBMDT.userid</td>
</tr>
<tr>
<td>Debug Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Option</td>
<td>DIRECT&amp;ip_name%port</td>
<td>TCPIP&amp;ip_name%port</td>
</tr>
<tr>
<td>Debug View stack</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IBM Debugger for z/OS – TEST invocation parameter

TEST parameter syntax

TEST(test_level, cmd_file, prompt_level, preference_file)

Standard Mode Test preference file parameters:

TEST(,,,DIRECT&1.2.3.4%8001:)
TEST(,,,DBM:*)
TEST(,,,DBM%TSS14:*)

Compatibility Mode Test preference file parameters:

TEST(,,,TCPIP&1.2.3.4%8001:)
TEST(,,,DBMDT:*)
TEST(,,,DBMDT%TSS14:*)

3270 Terminal Mode preference file parameters:

TEST(,,,VTAM%EMRICH:*)

Note: The userid sub-parameter following the DBM or DBMDT is optional.

- It is required if the userid used to sign on to the Remote Systems Explorer by the GUI is different than the userid associated with the program being debugged IBM Developer for z Systems Enterprise Edition (5655-AC5) (IDz EE)
Launchers

- DTSP / DTCN are deprecated (but probably still needed yet)
- The strategic way to manage debugging / debugging profiles with IDz EE

- Eclipse launches can be
  - exported/imported for sharing,
  - marked as favorite for easy to access from toolbar
  - etc.
Remote Application launch

- Remote Profile Tab

- Debug Options Tab
Remote Application launch

- **Common Tab**

- **Code Coverage Tab**
Remote CICS Application launch

- Provides CICS specific debugging settings

Provide CICS specific debugging settings.
Remote MVS Batch specific launcher

- Provide JCL generation option (without saving)
  - Generate JCL to execute a load module (program option)
  - Generate JCL to compile link and execute the resulting load module (source option)
  - Augment existing JCL with debug and code coverage options
DTSP migration assistant

Automatic Pop-up DTSP view is open and PDTCC connection gets established

The Debug Tool DTSP profile views are now replaced by Remote DB2 and IMS Application launches. Please select the profiles you want to migrate:

- Migrate Server Profile to a Debug configuration
- Hide Debug Tool views after migration
- Open Debug Launch Configurations dialog upon completion

Each of the following DTSP profiles will be migrated to a Debug configuration. You can change the launch type if it does not match the one you expect.

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Launch Type</th>
<th>Migrated Debug Configuration Name</th>
</tr>
</thead>
</table>

Finish Cancel
Visual Debugging

- Requires IDz
- Available for PLI and COBOL programs
- Turned on at any point during debug from toolbar locations:

Supports 2 modes:
- Entry mode shows the flow within the program currently open in the debug session
- Program mode shows the flow in/out of the current program
Visual Debugging

- Current stack is highlighted

Stack Pattern Breakpoint

- Complimentary feature to Visual Debugging
- Rather than setting a breakpoint at a specific entry, you can select the path to that entry, reducing the number of times the debugger must stop to get to desired area
- Can only be set from PCF view: select desired path, and use context menu
Stack Pattern Breakpoint editing
Source Level Breakpoint

- Available for PLI and COBOL (System z LPEX Editor, COBOL Editor, PLI Editor),
- Allows the user to set a breakpoint while editing/viewing source
- Use the ruler context menu or outline view to set a breakpoint
- Note: important that you click in an area where there is a paragraph (COBOL) or procedure/entry (PLI), or the menu action may **not** appear
Code Coverage

- To be configured via Launcher tab
Code Coverage

- Code Coverage results view is available with IDz
  - Includes new Import Debug Tool Code Coverage wizard to select debug tool code coverage XML data set and convert to Compiled Code Coverage result format.

<table>
<thead>
<tr>
<th>Code Coverage Results</th>
<th>Status</th>
<th>Coverage</th>
<th>Level</th>
<th>Analyzed Date</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiled Code Coverage Workspace Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JUnit Code Coverage Workspace Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java Code Coverage Workspace Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Code Coverage results view

![Code Coverage Results](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Coverage</th>
<th>Level</th>
<th>Analyzed Date</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiled Code Coverage Workspace Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAM1_2016_07_28_161237_0102</td>
<td></td>
<td>76%</td>
<td>Line</td>
<td>28-Jul-2016 4:12:37 PM</td>
<td></td>
</tr>
<tr>
<td>SAM1_2016_07_28_161159_0386</td>
<td></td>
<td>76%</td>
<td>Line</td>
<td>28-Jul-2016 4:11:59 PM</td>
<td></td>
</tr>
<tr>
<td>JUnit Code Coverage Workspace Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java Code Coverage Workspace Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions?