BC/NAV TECH DAYS 2022

mibuso.com

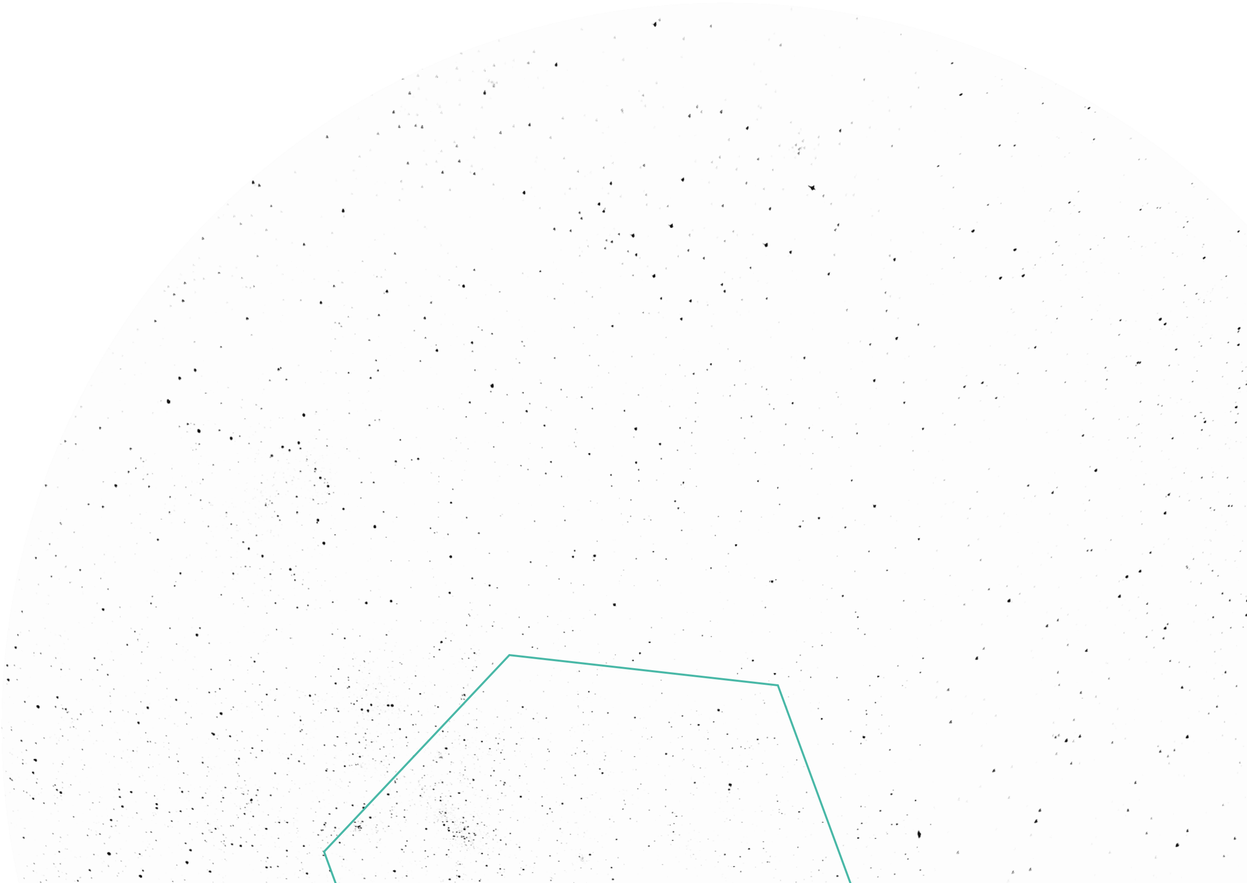
Troubleshooting Business Central SaaS environments

Kalman Beres, Nikolay Dobrev
MICROSOFT

www.bctechdays.com

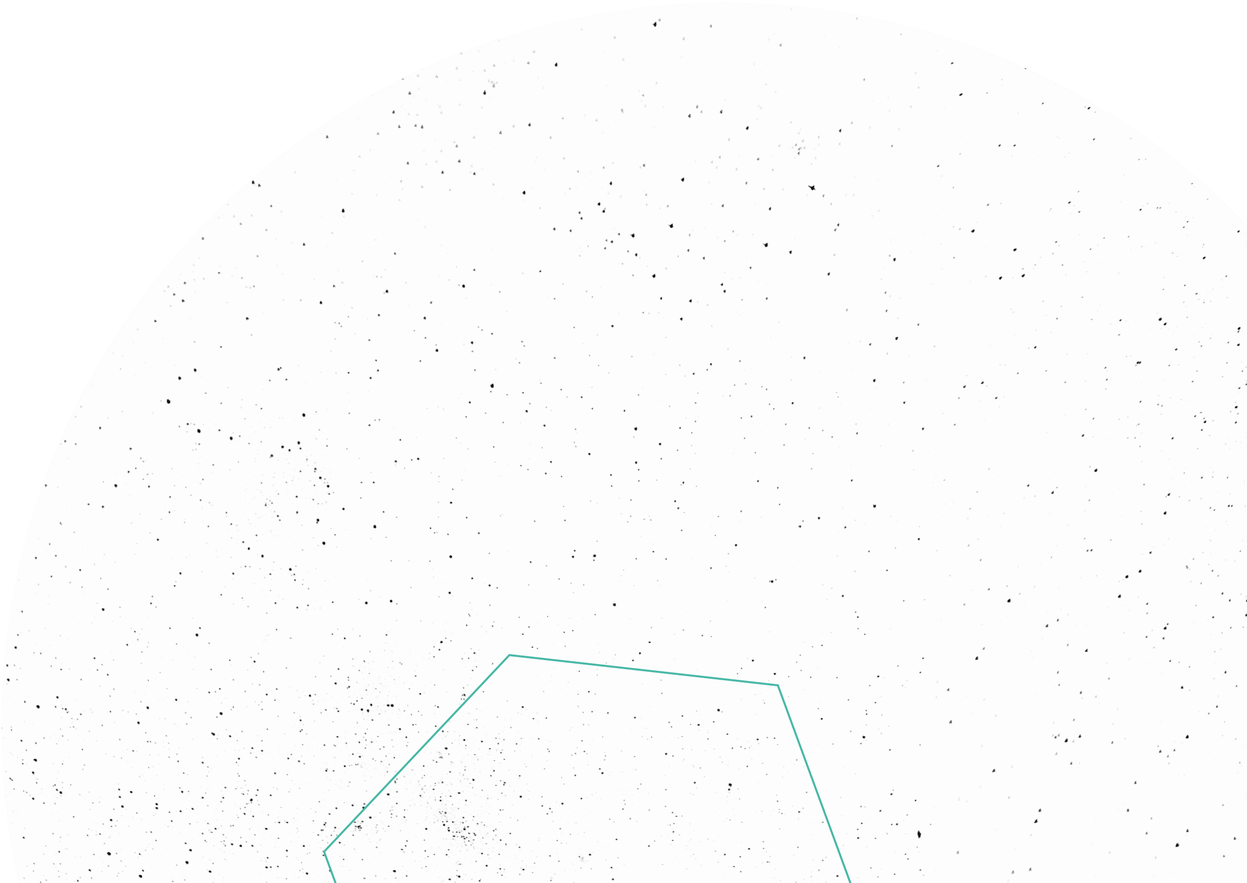
Agenda

- Telemetry
- In-client tools
- Debugging enhancements
- Snapshot debugging
- Profiling



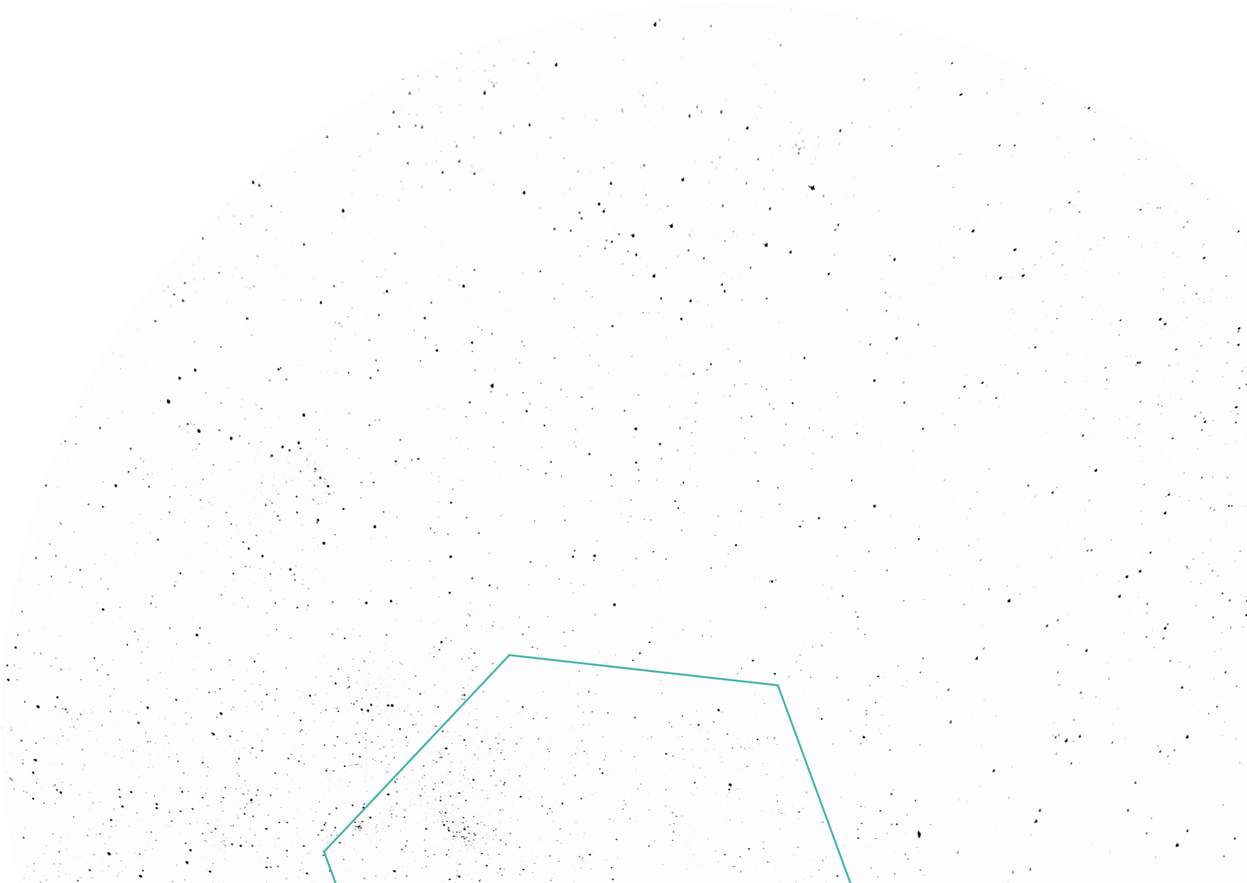
Agenda

- **Telemetry**
- In-client tools
- Debugging enhancements
- Snapshot debugging
- Profiling



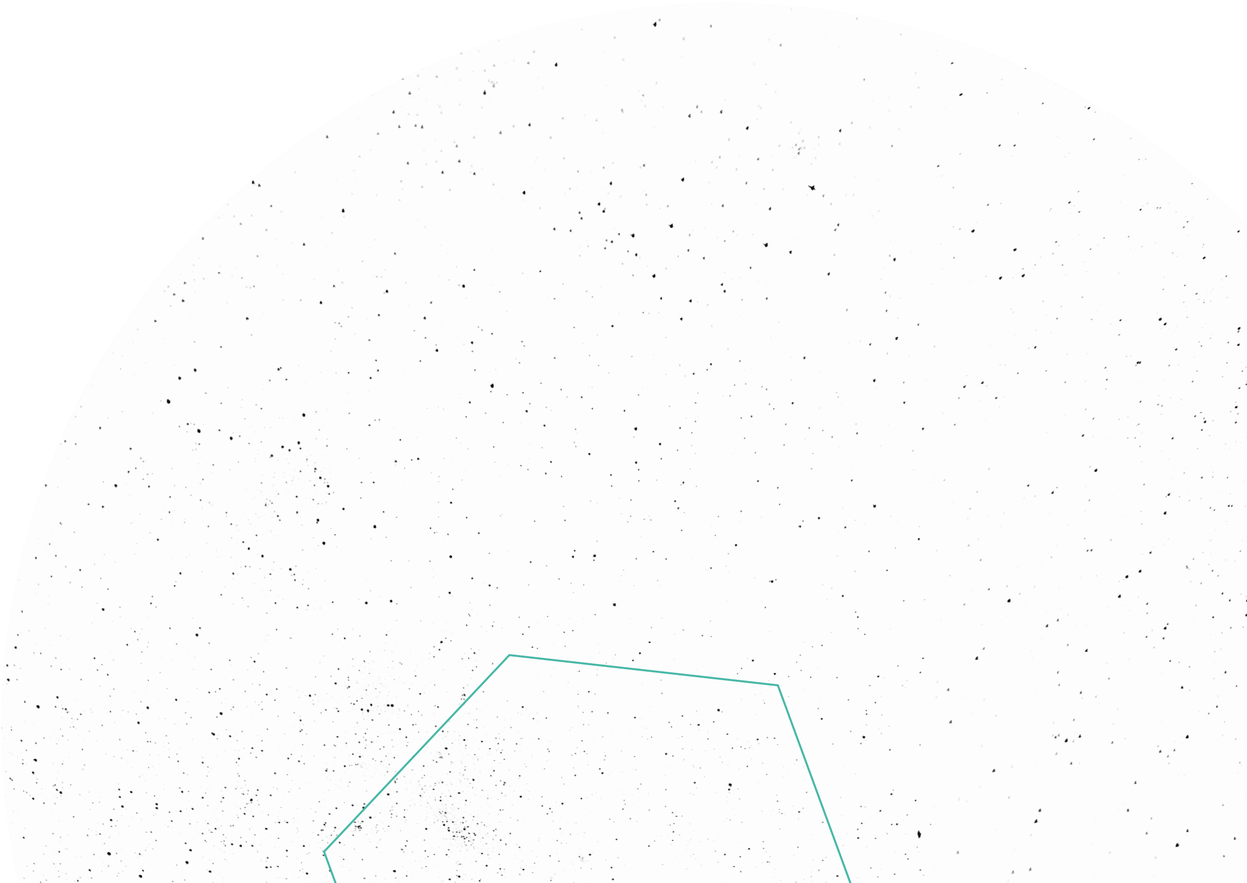
Telemetry

- Lifecycle
- Logins
- Performance
- AL errors
- PowerBI Dashboard
  http://aka.ms/bctelemetryreport



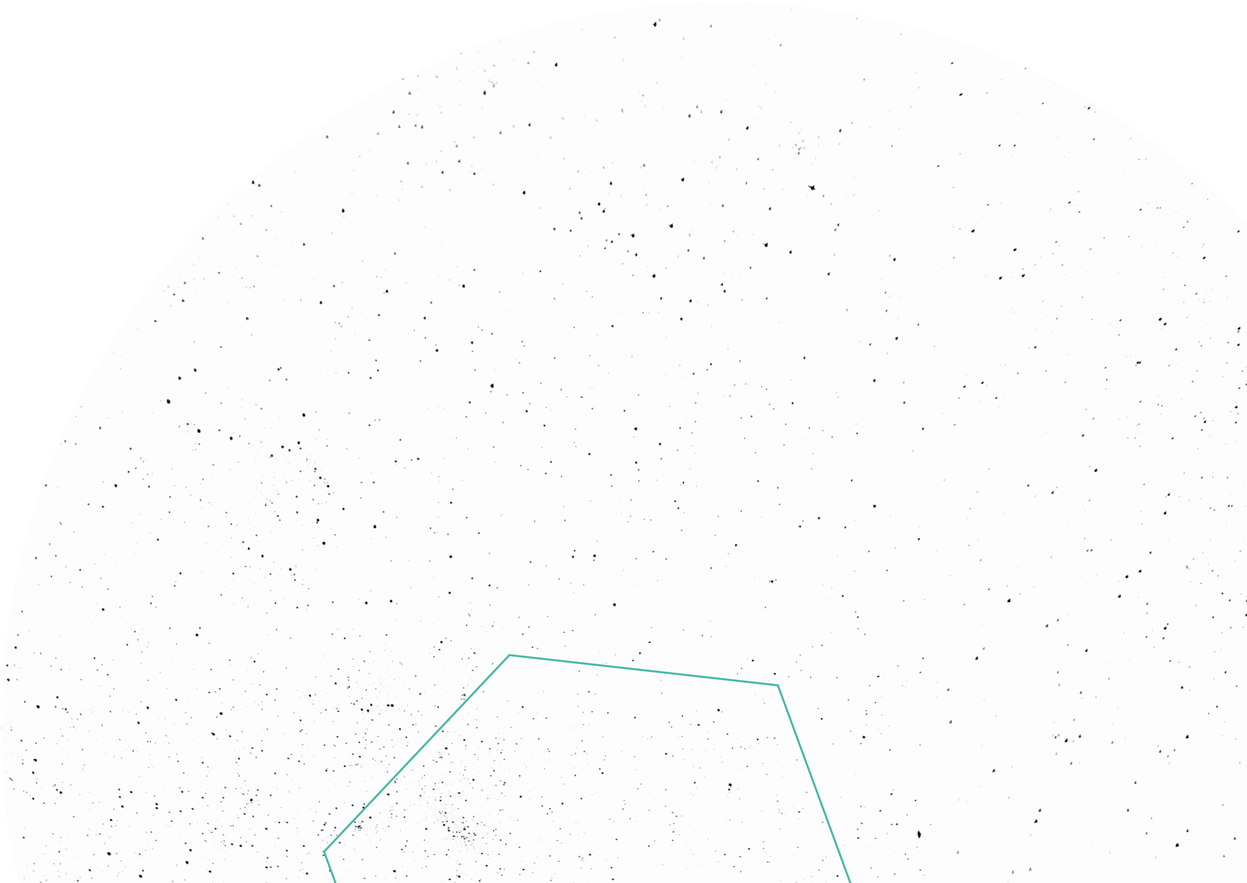
Agenda

- Telemetry
- **In-client tools**
- Debugging enhancements
- Snapshot debugging
- Profiling

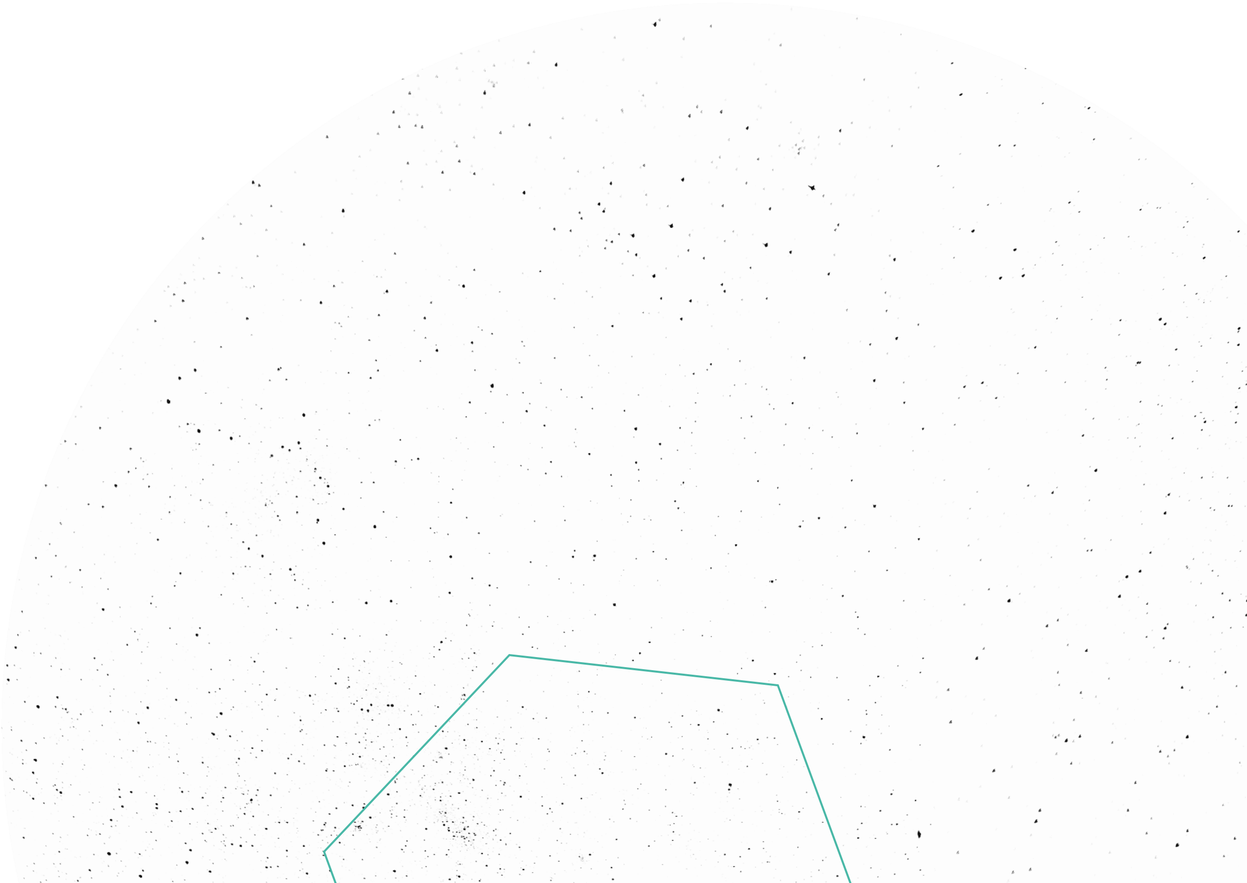


Agenda

- Telemetry
- In-client tools
- **Debugging enhancements**
- Snapshot debugging
- Profiling



Demo



Other debugger enhancements

- New setting to exclude temporary record read writes

```
"configurations": [
    {
        "name": "Publish: Microsoft cloud sandbox",
        "type": "al",
        "request": "launch",
        "environmentType": "Sandbox",
        "environmentName": "sandbox",
        "startupObjectId": 22,
        "breakOnError": "All",
        "breakOnRecordWrite": "",
        "launchBrowser": true,  "All"
        "enableSqlInformationDe  "ExcludeTemporary"
        "enableLongRunningSqlSt  "None"
```

- New setting to exclude breaking on errors on try functions

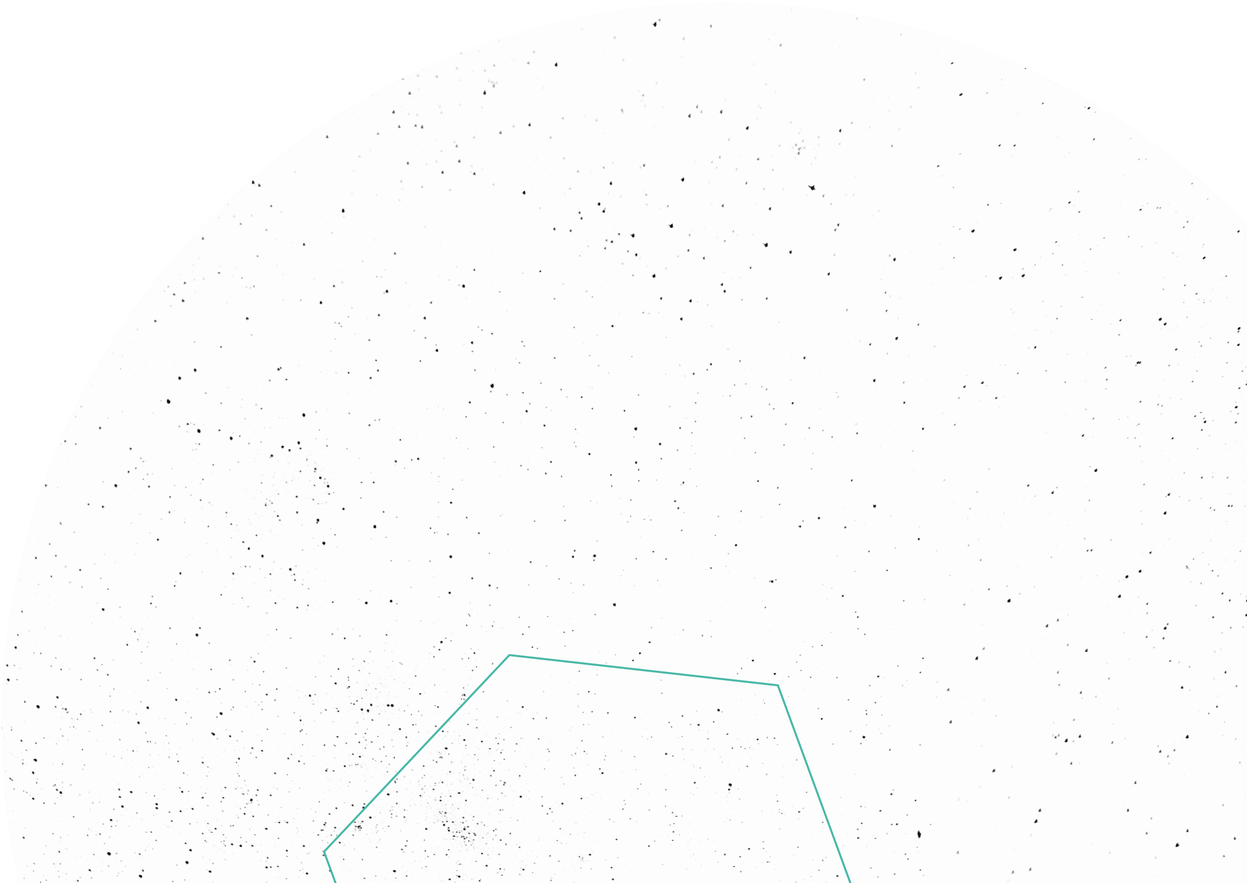
```
"configurations": [
    {
        "name": "Publish: Microsoft cloud sandbox",
        "type": "al",
        "request": "launch",
        "environmentType": "Sandbox",
        "environmentName": "sandbox",
        "startupObjectId": 22,
        "breakOnError": "",
        "breakOnRecordWri  "All"                    Default value
        "launchBrowser":  "ExcludeTry"
        "enableSqlInforma  "None"
```

- Startup company can be specified in the launch.json

```
5            "enableLongRunningSqlStatements": true,
6            "longRunningSqlStatementsThreshold": 500,
7            "numberOfSqlStatements": 10,
8            "startupCompany": "CRONUS USA, Inc."
9        },
```

Agenda

- Telemetry
- In-client tools
- Debugging enhancements
- **Snapshot debugging**
- Profiling



Snapshot debugging

Recording of a session and looking at the state at certain points in time: *snappoints*

Snapshot debugging does not stop the execution and only records state on snappoints.
So, it is enabled in PRODUCTION.

1. Set snappoints and start snapshot debugging for a session
2. Perform the scenario you want to debug
3. Download snapshot result
4. Step through the snapshot and inspect flow & data

Snapshot result will contain call-stack and select local & global variables values recorded for the snappoints.

Allows rapid investigation and collaboration with the customer on exact reproduction steps

Snapshot launch.json configuration

Property	Description
request	Type of launch configuration. Must be “snapshotInitialize”
environmentName	The SAAS environment name
environmentType	Production, Sandbox, OnPrem
tenantId	The AAD tenant id, or the authority domain (<xyz>.onmicrosoft.com)
breakOnNext	WebClient, WebServiceClient, or Background (only UI-less user sessions)
sessionId	Integer id of running session to attach to. Help and Support page/Tenant Admin Center
userId	The ID of the user to attach to. Can be empty (debug next), a user ID security GUID or username
snapshotVerbosity	If set to SnapPoint, call-stack is only captured on snappoints. Default is Full
executionContext	Debugging, Profiling or both
profilingType	Sampling or Instrumentation

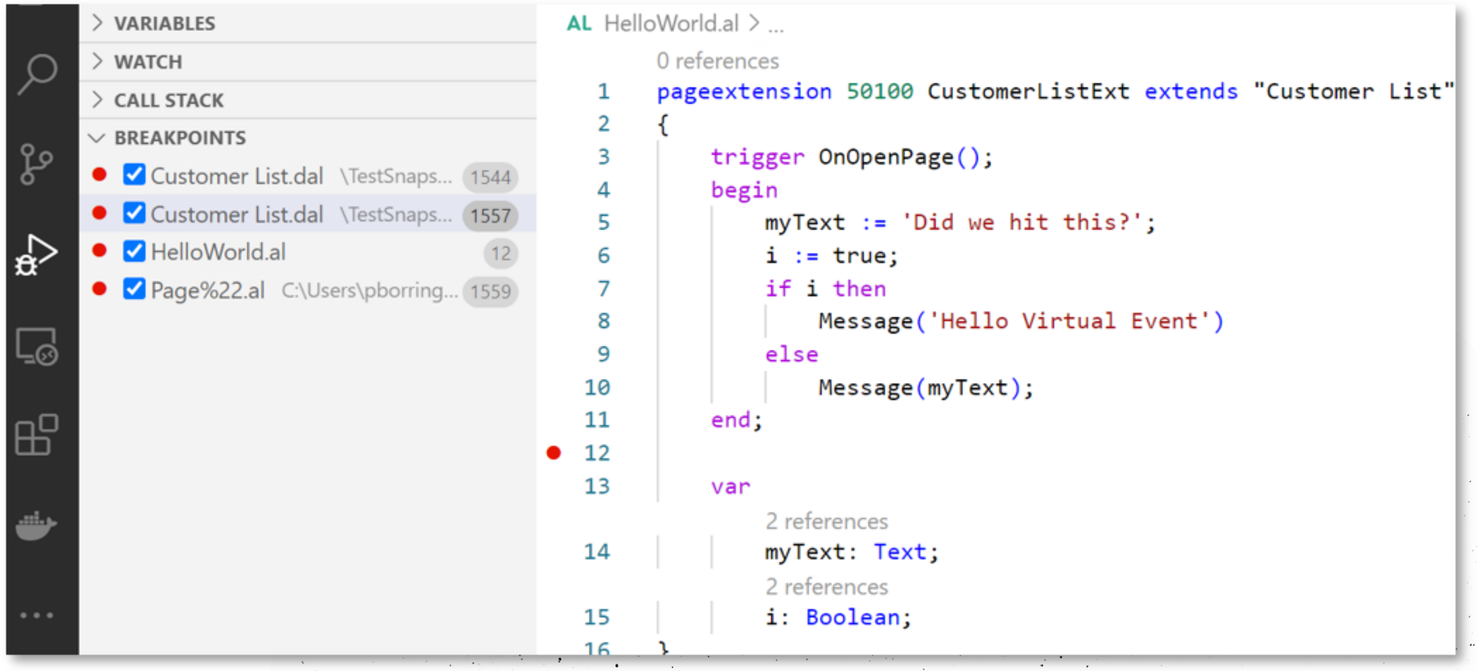
```
{
    "name": "Snapshot sandbox next user session",
    "type": "al",
    "request": "snapshotInitialize",
    "breakOnNext": "WebClient",
    "userId": "JANEDOE",
},
{
    "name": "Snapshot sandbox existing session ID",
    "type": "al",
    "request": "snapshotInitialize",
    "breakOnNext": "WebClient",
    "sessionId": 2233
},
```

[Snapshot Debugging - Business Central | Microsoft Docs]

Adding snappoints

Conceptually same as breakpoints

- Added in gutter or using shortcut
- Shown in breakpoints list
- Also in symbol code (DAL)

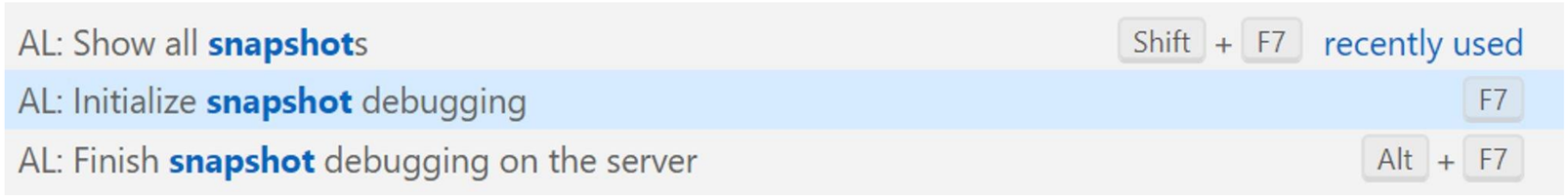


However

- Not a breakpoint, execution is not halted
- Variables only collected at snappoints (and exceptions)
- Must be added <u>before</u> initializing snapshot

Initializing snapshot

F7 or ”AL: Initialize snapshot debugging” command

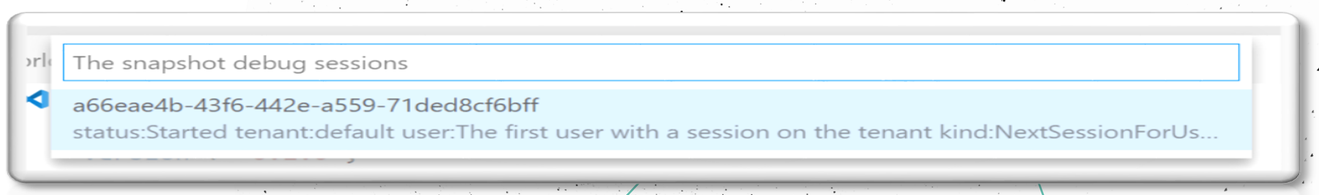


Prepares snapshot capture

- Uploads snappoints and trigger rules to target server
- Starts listening, if it can acquire a session then it sets status to started
- Snapshot debugging session counter on the status-bar will be updated

View all current snapshots

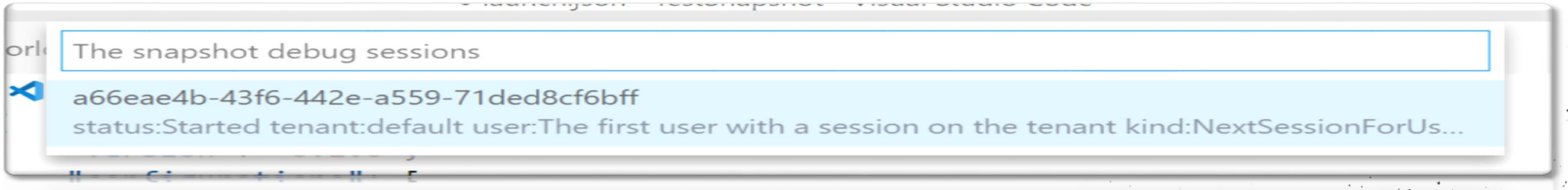
- Click the debug icon in the status bar
- Shift+F7
- “AL: Show all snapshots” command



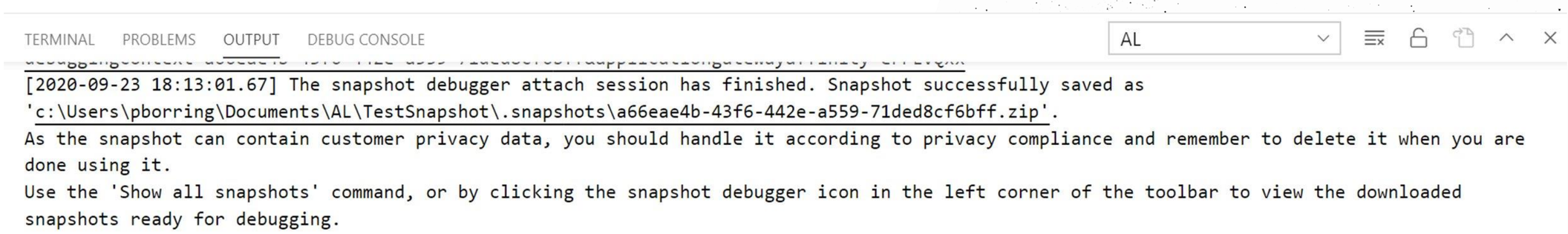
Finish and download snapshot session

You finish snapshot debugging session by pressing Alt + F7

Brings up all snapshot sessions that have been started.



Choosing one will close the session debugging on the server and download the snapshot file.



Snapshot states

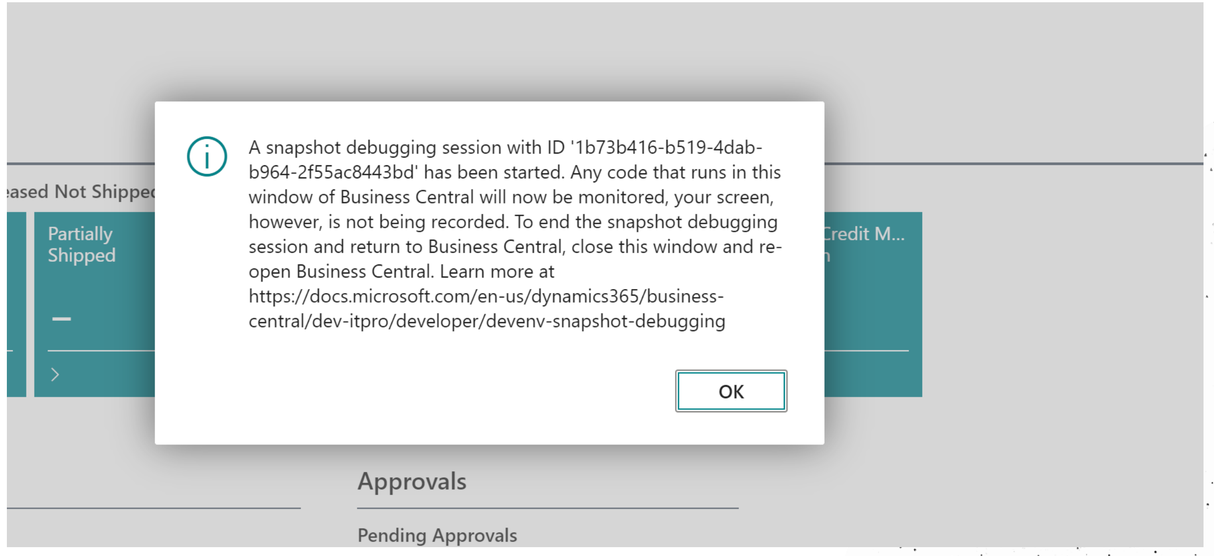
BC/NAV TECH DAYS 2022
mibuso.com

A snapshot debugging session can be in one of these states

State	Description
Initialized	Server is waiting for the next session to be snapshot debugged based on specified session rules
Started	Attached to end-user session and snapshot capture in progress
Finished	Snapshot capture has finished
Downloaded	Snapshot file is downloaded

Warning on capturing user session

If initializing a snapshot session for the next session, the user will be warned



Snapshot access and permissions

BC/NAV TECH DAYS 2022
mibuso.com

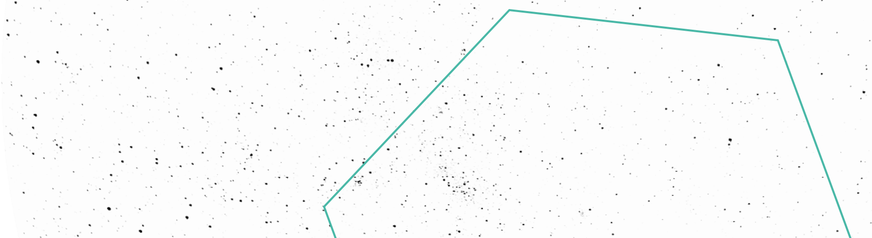
D365 Snapshot Debug permission set

To create and download a snapshot file that exists on the server on behalf of an end-user .

ResourceExposurePolicy (former ShowMyCode) and NonDebuggable, Dynamic IP protection are respected as with normal debugger.

Must be started within 30 minutes after initialization .

Must be finished within 10 minutes.



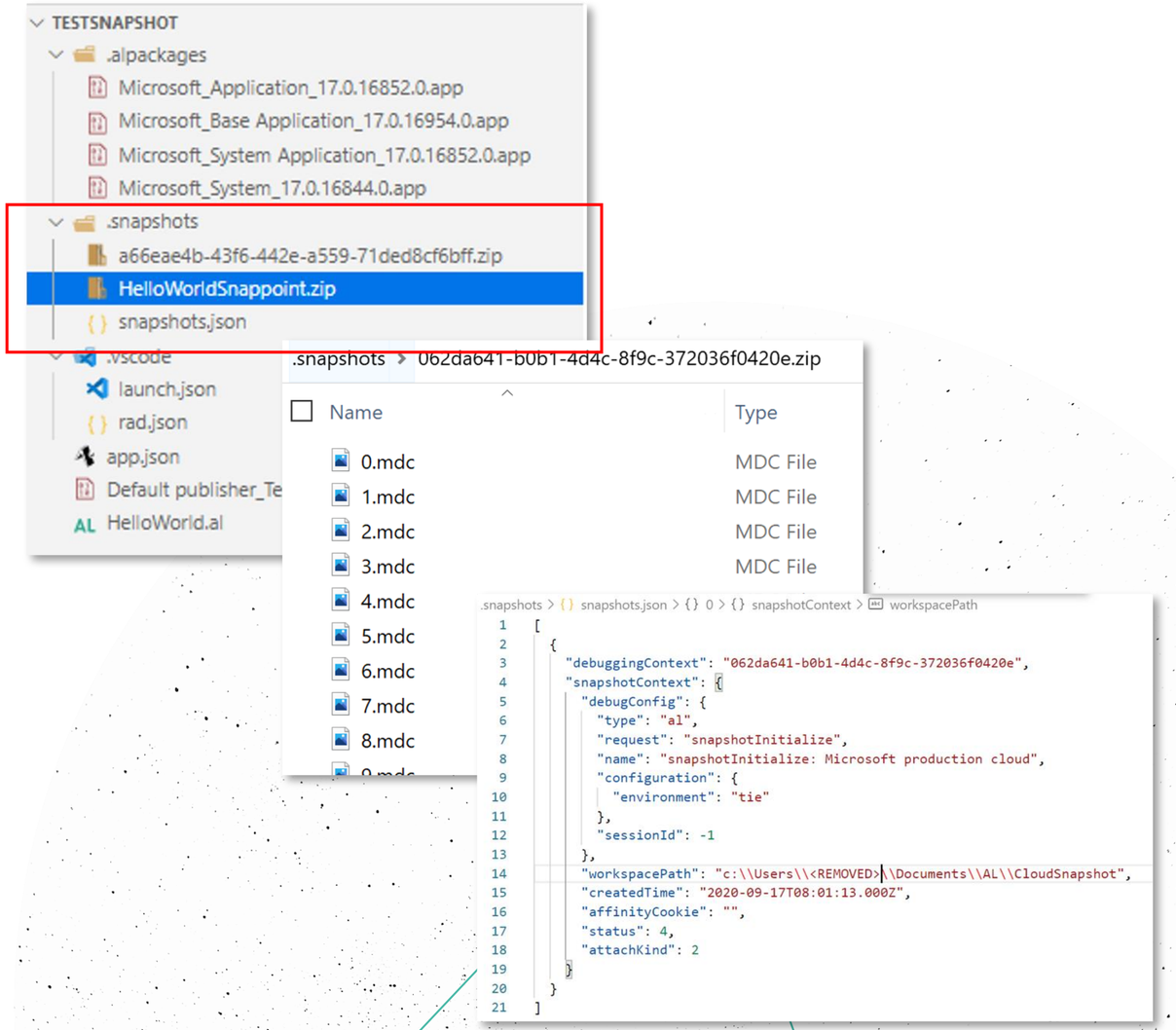
Snapshot files

Archive (.zip) file

- Default downloaded to `./.snapshots` in current workspace.
- Override download location with `al.snapshotOutputPath` setting.
- Contains debug metadata (code, callstack and snapshot data etc.)

Snapshots can contain customer privacy data!

- Must be handled accordingly to privacy compliance.
- Delete when no longer needed.
- Unfinished snapshots pruned on internal server side.



Debugging snapshot file offline

BC/NAV TECH DAYS 2022
mibuso.com

Downloaded snapshot session can be debugged.

Shift+F7 or click debug status icon, select finished snapshot capture in ./.snapshots ("al.snapshotDebuggingPath" override).

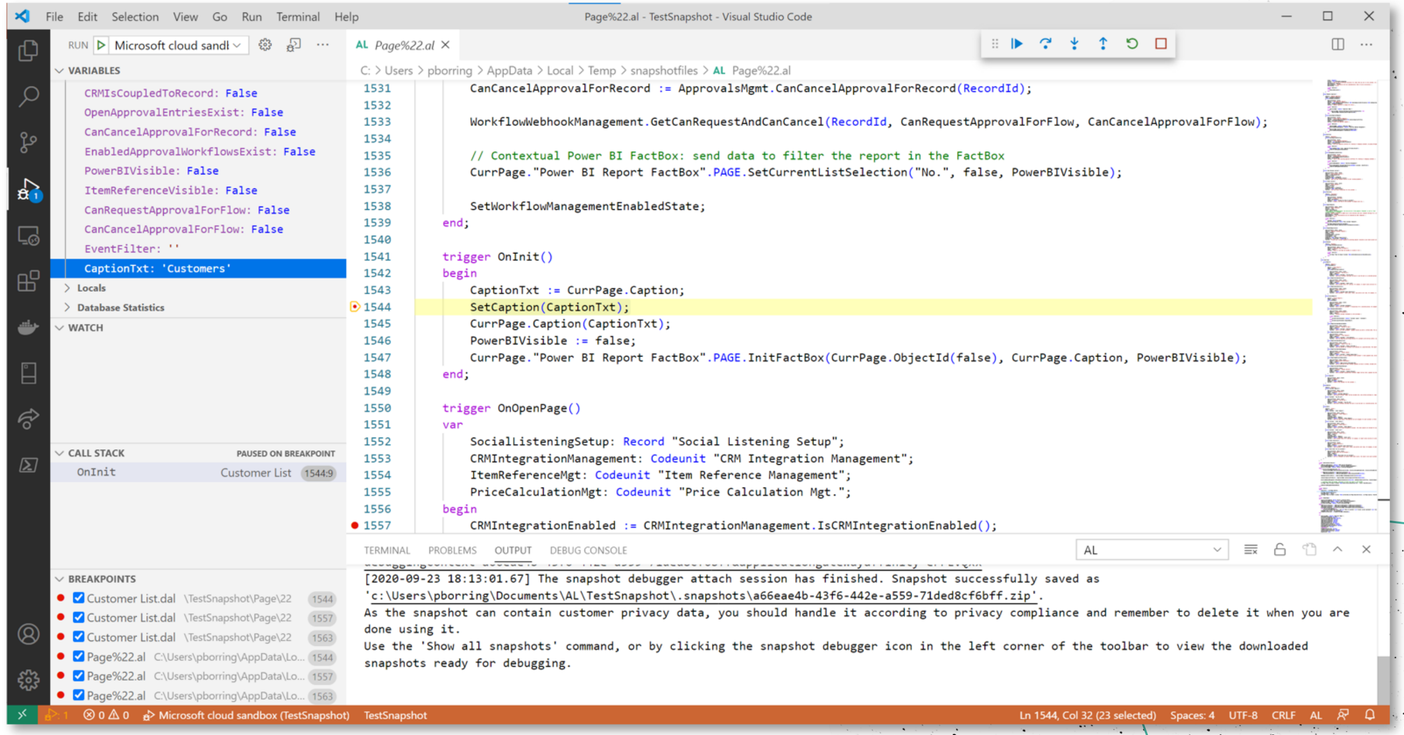
F5 and chose snapshot launch configuration with a "snapshotFileName" set.

Code "execution" will stop at first snappoint hit

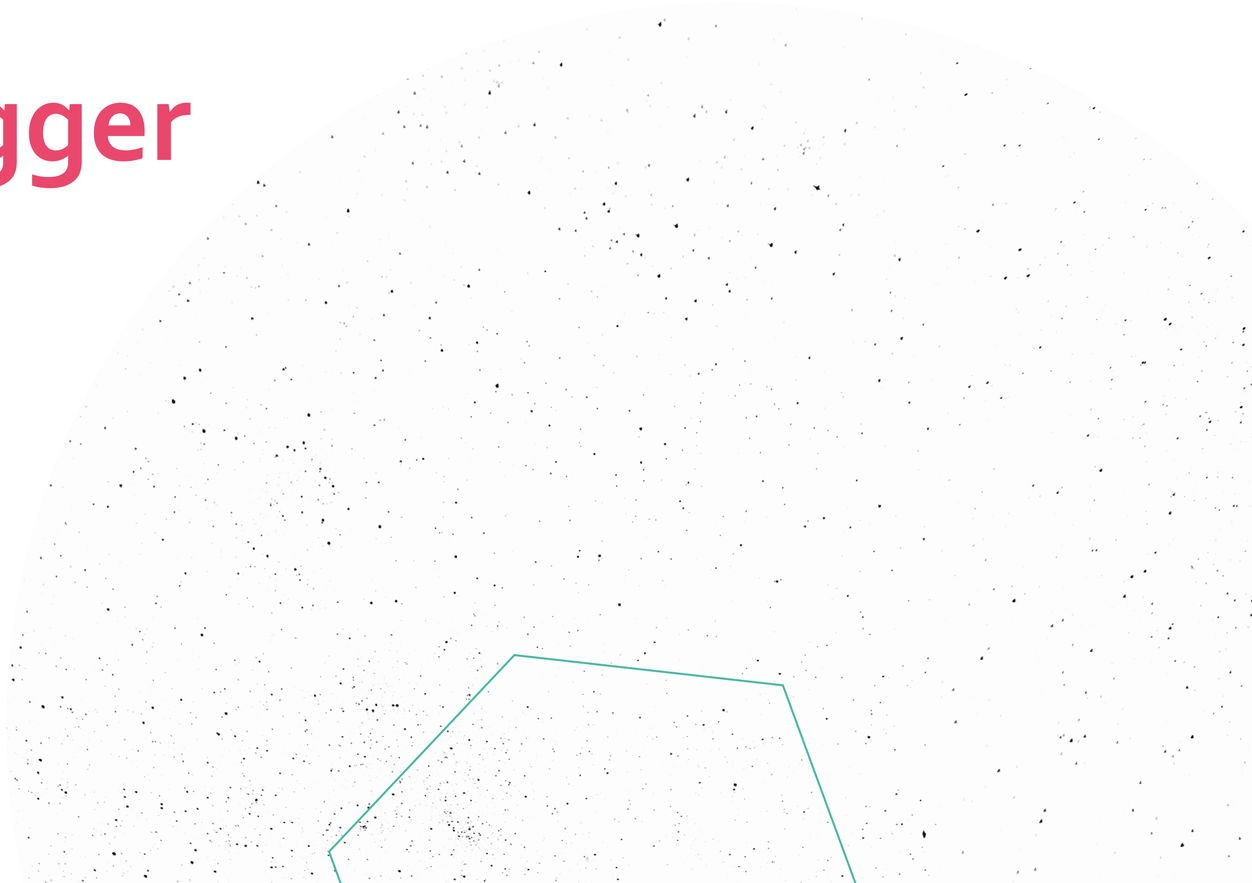
- AL exceptions will be treated as snappoints
- No snappoints? Entry first line in first captured method

F5 will jump between snappoints/breakpoints

- Can insert new breakpoints, but no variable state after capture



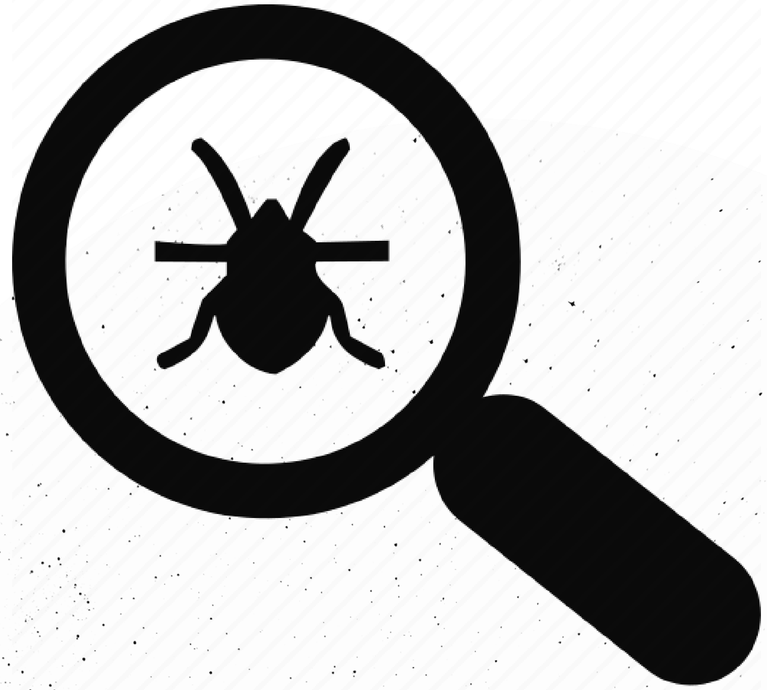
Demo Snapshot debugger



Debugging

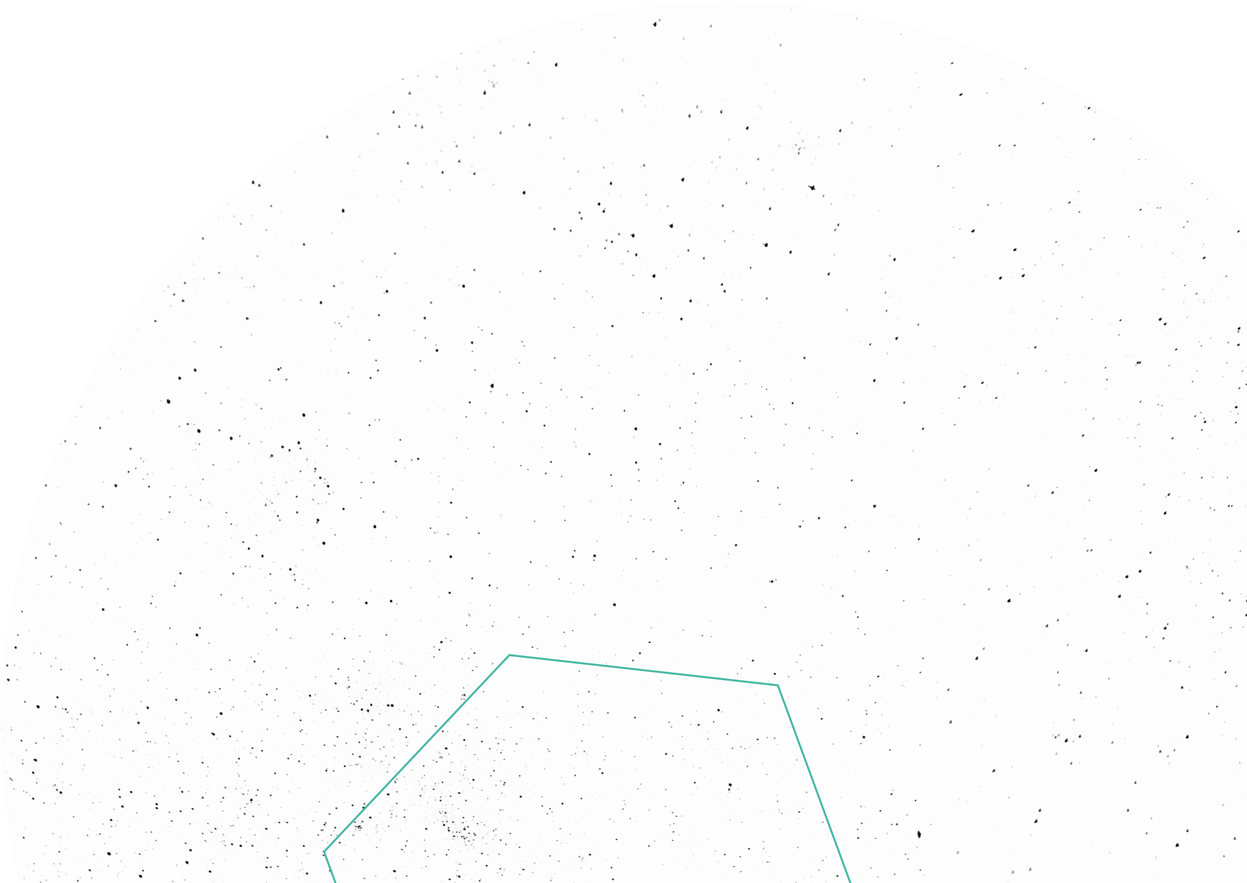
Snapshot debugging recording with state at snappoints in **PRODUCTION** a nd **SANDBOX.**

Normal debugging, halting execution and doing line by line debugging in **SANDBOX** only.



Agenda

- Telemetry
- In-client tools
- Debugging enhancements
- Snapshot debugging
- **Profiling**



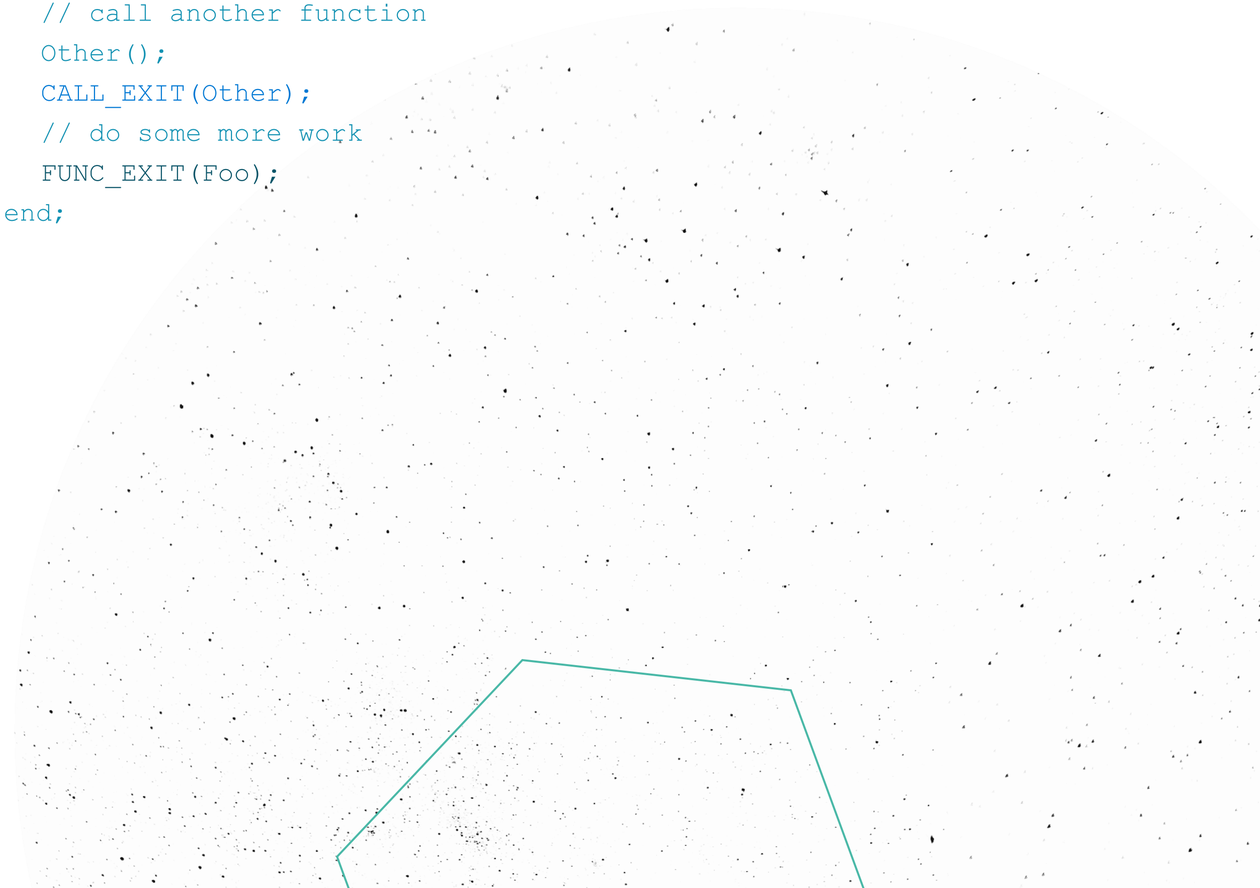
AL Profiler

AL profiler introduced using instrumentation

- Tracks start and stop for each called method and computes time spent
- Tracks # of hits for a called method

The next release we have introduced sampling based profiling.

```
procedure Foo()
begin
   FUNC_ENTER(Foo);
   // do some work
   CALL_ENTER(Other);
   // call another function
   Other();
   CALL_EXIT(Other);
   // do some more work
   FUNC_EXIT(Foo);
end;
```



Sampling vs. Instrumentation

BC/NAV TECH DAYS 2022
mibuso.com

Sampling

- Low overhead
- Not all calls are captured
- UI interactions, external calls are included
- Time is a factor of sampling interval + overhead

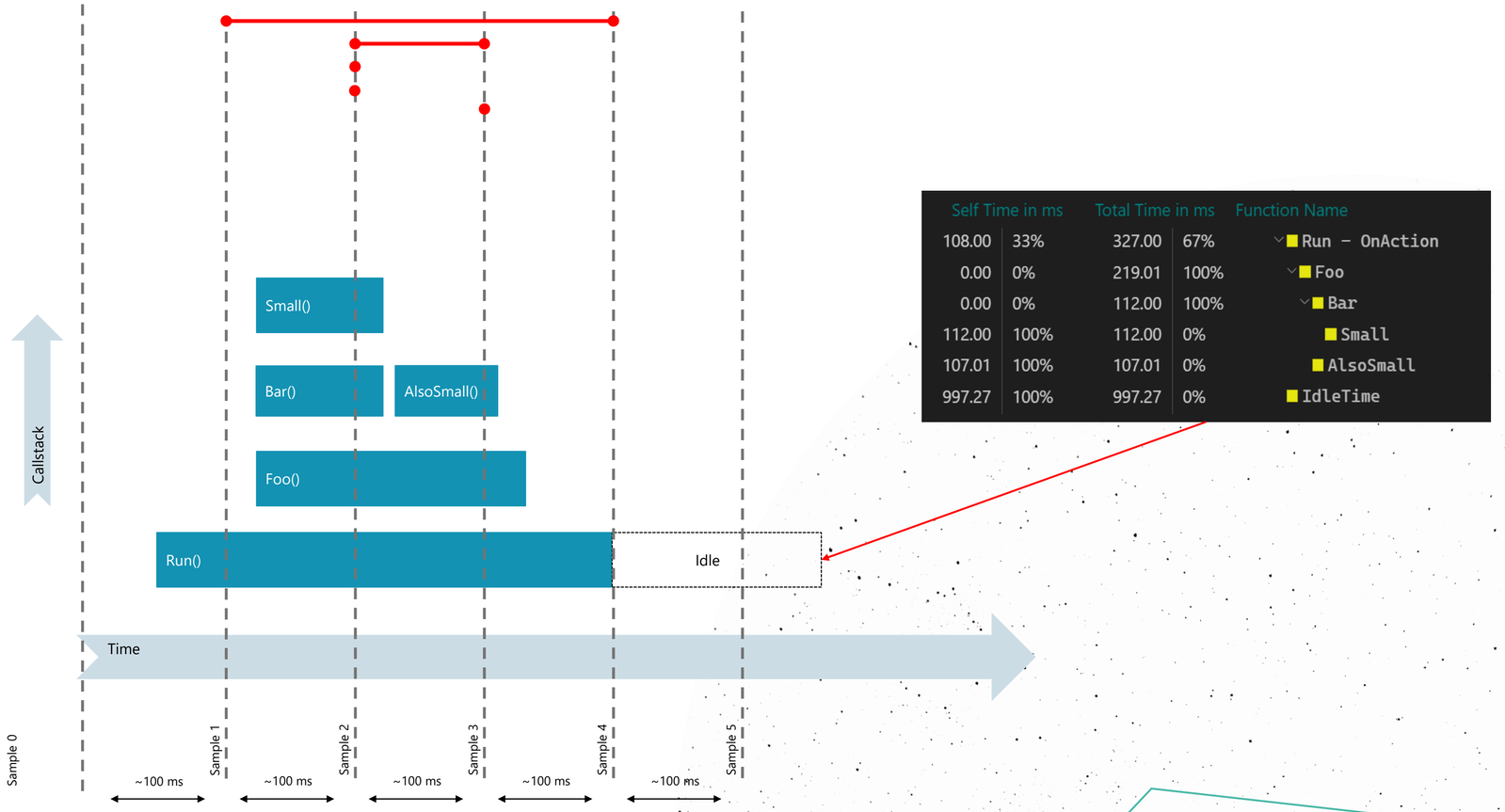
Self Time in ms		Total Time in ms		Function Name
108.00	33%	327.00	67%	Run - OnAction
0.00	0%	219.01	100%	Foo
0.00	0%	112.00	100%	Bar
112.00	100%	112.00	0%	Small
107.01	100%	107.01	0%	AlsoSmall
997.27	100%	997.27	0%	IdleTime

Instrumentation

- Resource intensive
- More details

Self Time in ms		Total Time in ms		Hit count	Function Name
221.89	61%	359.14	39%	1	Run - OnAction
137.25	55%	248.68	45%	2	Foo
32.32	29%	109.28	71%	1	Bar
76.96	100%	76.96	0%	1	Small
79.11	100%	79.11	0%	1	AlsoSmall

Sampling



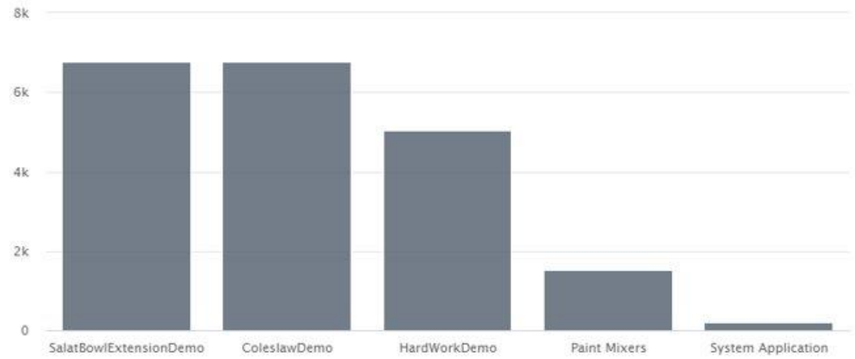
Self Time in ms		Total Time in ms		Function Name
108.00	33%	327.00	67%	Run - OnAction
0.00	0%	219.01	100%	Foo
0.00	0%	112.00	100%	Bar
112.00	100%	112.00	0%	Small
107.01	100%	107.01	0%	AlsoSmall
997.27	100%	997.27	0%	IdleTime

In-client Profiler

Performance Profiler

Start | Stop | Clear | Download | Share | Upload

Active Apps | Actions

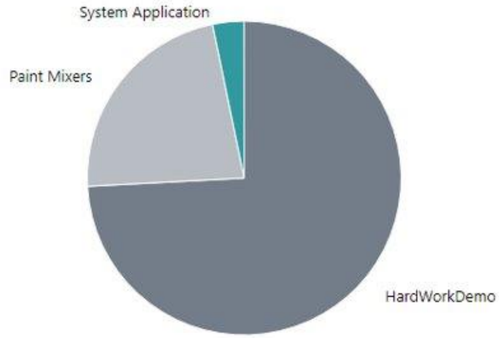


Shows the apps that where active during the recording. They were either running or called other apps. The duration represents the time you might save by removing an app.

Performance Profiler

Start | Stop | Clear | Download | Share | Upload

Time Spent | Actions



Shows which apps were running during the recording. Durations are self-time. They show the length of activity but do not include time spent calling other apps.

Performance Profiler

Start | Stop | Clear | Download | Share | Upload

Time Spent by Application Object

Object Type	Object Name	Time Spent ↓	App Name
Codeunit	HWoWorkerBee	5 seconds 27 mi...	HardWorkDemo
Codeunit	PMxPaintMixer	1 second 532 mi...	Paint Mixers
Codeunit	Azure AD Graph Impl.	219 milliseconds	System Application

Call Tree

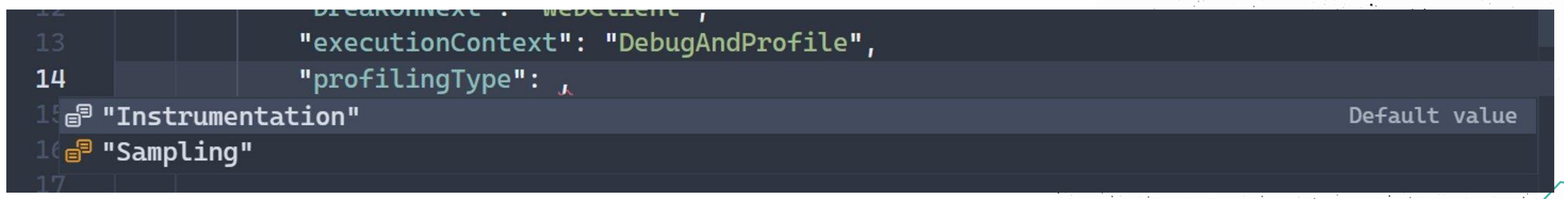
Method Name	Object Type	Object Name	Self
Paint - OnAction	Page	Mixer List	

Sampling in AL profiler

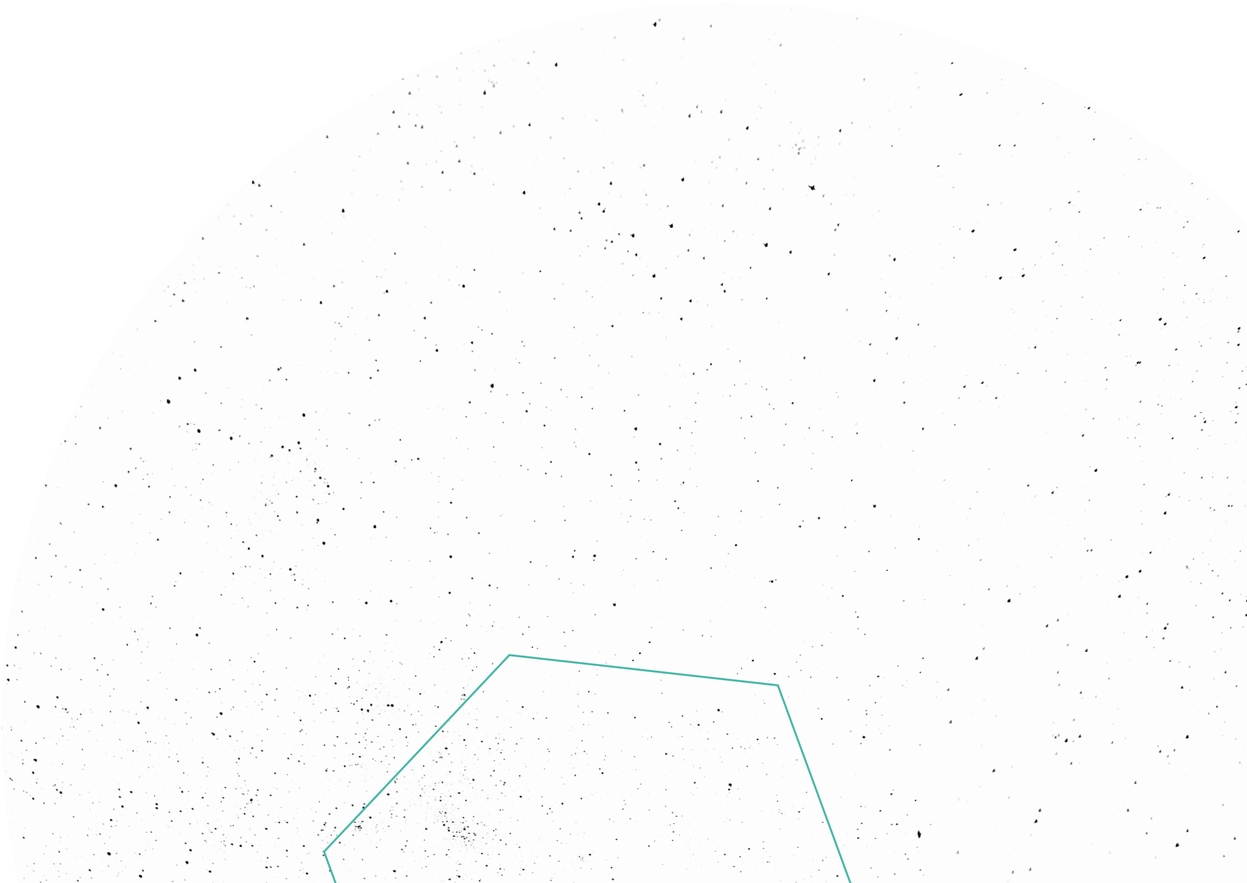
New profiling type: sampling
- At a fixed interval (sample point), track current method/call stack frame
- Sample interval configurable on NST, set to 100ms in cloud
- Total and self-times effectively = # of samples hit * (sample interval + overhead)
- Method calls happening between sampling intervals will not appear

Sampling will include time spent outside AL code
- New Idle node in profile

New profilingType setting in launch.config



Demo- AL profiler



Profiling

BC/NAV TECH DAYS 2022
mibuso.com

In-Client Profiling

- Understanding where time is spent in **PRODUCTION** and **SANDBOX**

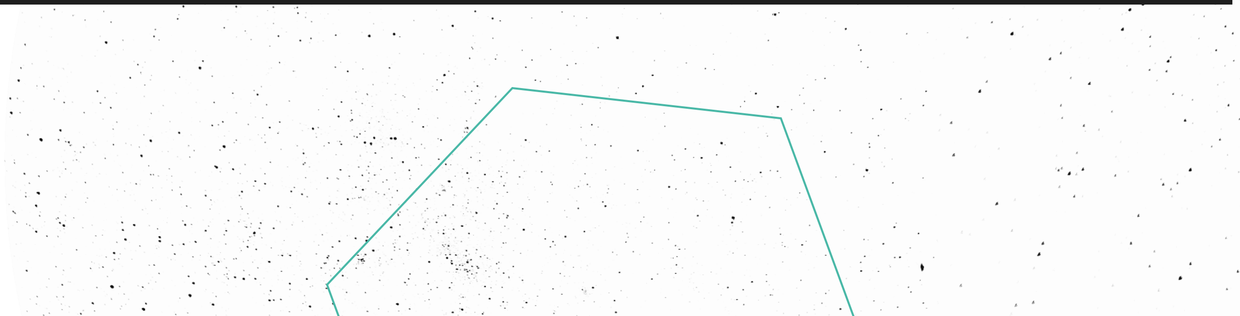
Sampling

- Export profile result from client or profile using the *profiletype* sampling to understand where time is spent in **PRODUCTION** and **SANDBOX**

Instrumentation

- Profile using the instrumentation *profiletype* in VS Code for targeted profiling in a **SANDBOX**

Self Time in ms		Total Time in ms		Function Name
3,273.53	100%	3,273.53	0%	IdleTime
0.00	0%	6,777.02	100%	Paint - OnAction
0.00	0%	6,777.02	100%	PaintContainer
0.00	0%	6,777.02	100%	OnBeforePaintContainer
0.00	0%	6,777.02	100%	OnPaintContainer
0.00	0%	6,777.02	100%	PaintContainerHandler
1,531.62	100%	1,531.62	0%	PreparePaint
5,026.65	100%	5,026.65	0%	DoLongHaul
0.00	0%	218.75	100%	NextSubscribedSKU
0.00	0%	218.75	100%	NextSubscribedSKU



BC/NAV TECH DAYS 2022
mibuso.com

Thank You!