Microsoft Dynamics 365

Unlock the Power of AL extension toolset

A detailed overview of AL code debugging and navigation

Blanca Robledo
Thomas Pedersen

Agenda

- Troubleshooting
- Explorer
- Productivity boosters

Troubleshooting

Microsoft Dynamics 365

Troubleshooting overview

- Overview of debugging capabilities
- Regular Debugging
  - Debugging different type of sessions
  - Breakpoints

Microsoft Dynamics 365

Debugging Overview - Capabilities

	REGULAR DEBUGGING	SNAPSHOT DEBUGGING
Supported environments	Sandbox	Sandbox + Production
Launch Debugging	✓	✗
Attach to next session of type	✓ (WebClient only supported from BC22)	✓
Attach to existing session by ID	✓ (Only supported from BC22)	✓
Attach to next session by user	✓ (Only supported from BC22)	✓
Debug session belonging to another user	✓ (Only supported from BC22)	✓
Real-time Control (pause, step over, etc.)	✓	✗
Profiling information	✗	✓
Replay debugged session	✗	✓

Microsoft Dynamics 365

Regular and Snapshot Debugging use cases

REGULAR DEBUGGING

- Good for validating extension behavior during development.
  - Traditional debug controls
  - Faster publishing
- When you want to debug a scenario that can be reproduced on a Sandbox.

SNAPSHOT DEBUGGING

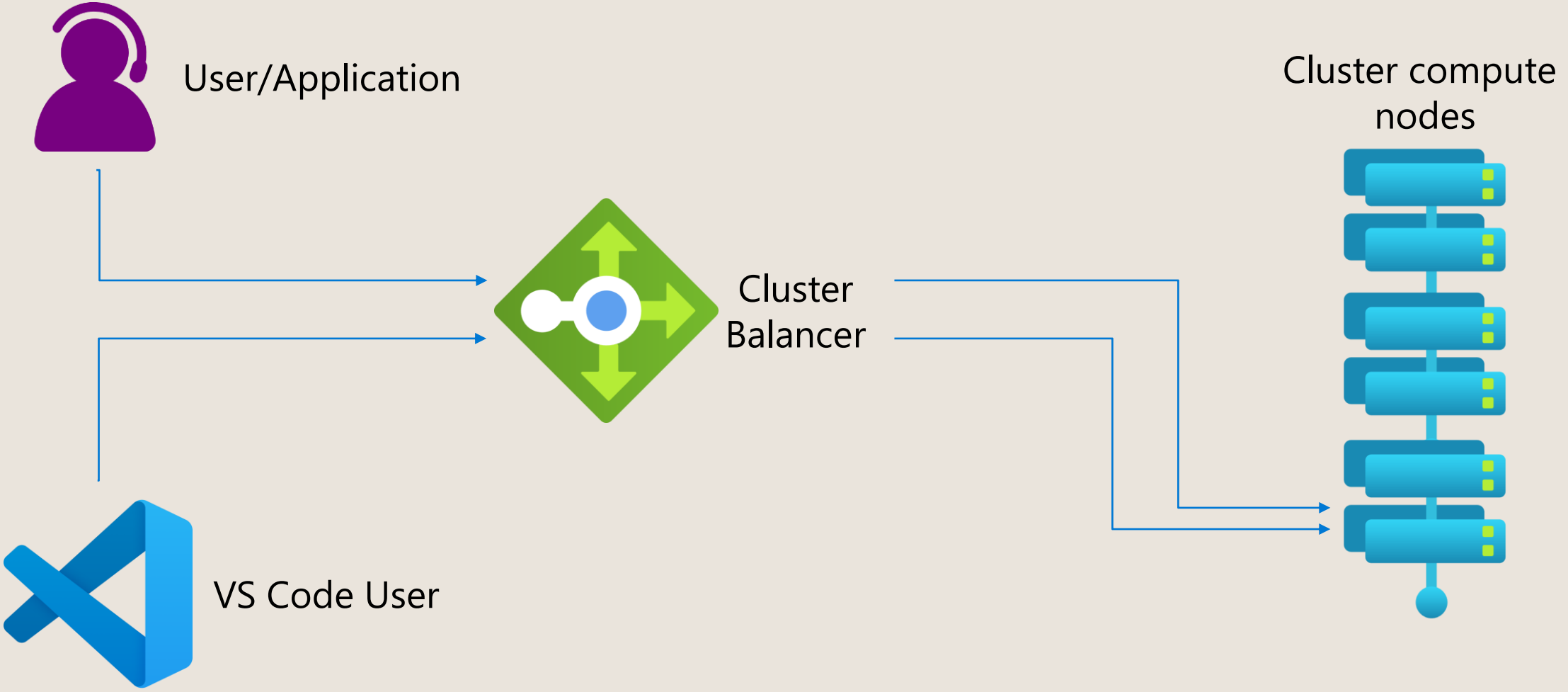
- When you need debug information about a scenario from a Production environment.
- When you need performance profiling information.
- When you want to share the debug results of a session.

Microsoft Dynamics 365

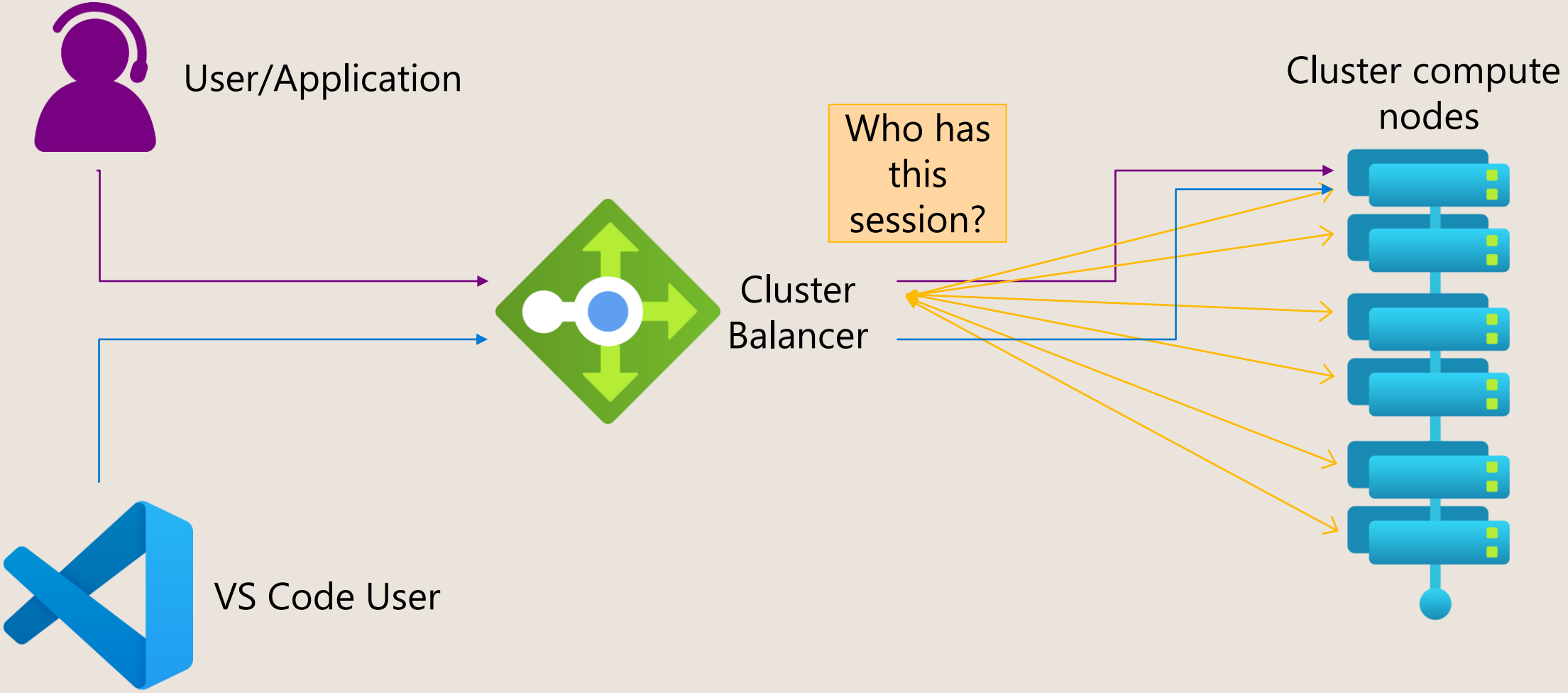
Ways of regular debugging a session

- Launching your own session
- Attaching to the next session
- Attaching to an existing session

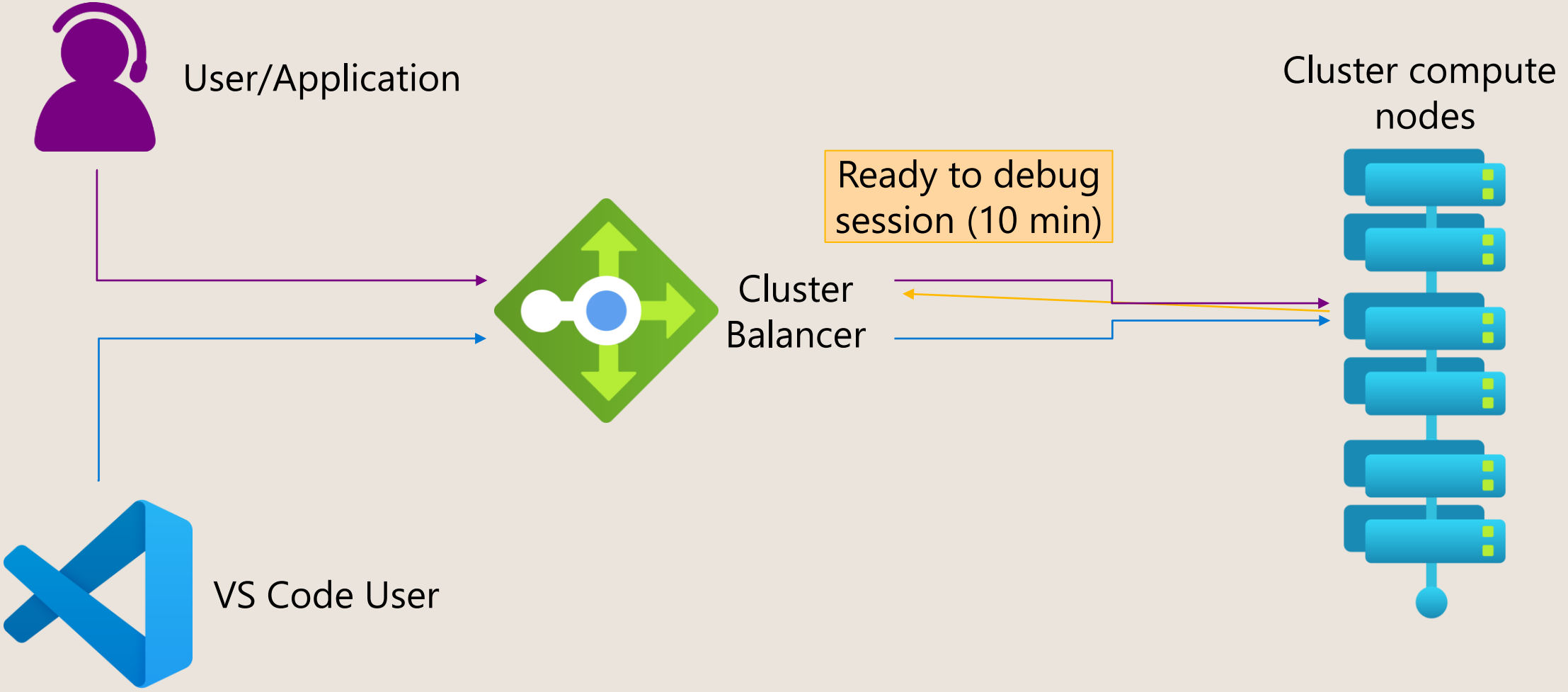
Debugging your own BC online session



Attaching to an existing BC online session



Attaching to the next BC online session



Microsoft Dynamics 365

Regular debugging – Capabilities

	Launch	Attach to Next	Attach to Existing
Target sessions belonging to other users	✗	✓	✓
Target existing session	✗	✗	✓
Publishes extension before debugging	✓	✗	✗
Webclient sessions	✓	✓ (from BC22)	✓
Non-webclient sessions (background, webservice)	✗	✓	✓ (..technically. But getting the session ID of a background/webservice session can be difficult)
Can specify startup options (object type/ID)	✓	✗	✗
Available in	All versions of BC SaaS	All versions of BC SaaS	From BC 22 onwards

Microsoft Dynamics 365

Regular debugging a session – When to use which?

LAUNCH

- Most useful when you're actively developing and need to quickly see changes.
- Allows for rapid iteration with control over how the extension is published (Especially powerful with RAD).

ATTATCH TO NEXT

- Use this when you want to debug a non-webclient session. Majority of the time this is used for debugging service-to-service (S2S) calls.
- This is also the **only** way you can debug install and upgrade codeunits.
- There is a 10-minute timeout on how long the server will wait for the new session. If a session matching the parameters in the launch.json is not found, the debug request will be aborted.

ATTATCH TO EXISTING

- If you have to target an existing session, so if you have a user who is already in the middle of doing something and it's already close to the problem code or is able to reproduce it consistently.

Microsoft Dynamics 365

Regular debugging a session – Configurations

LAUNCH

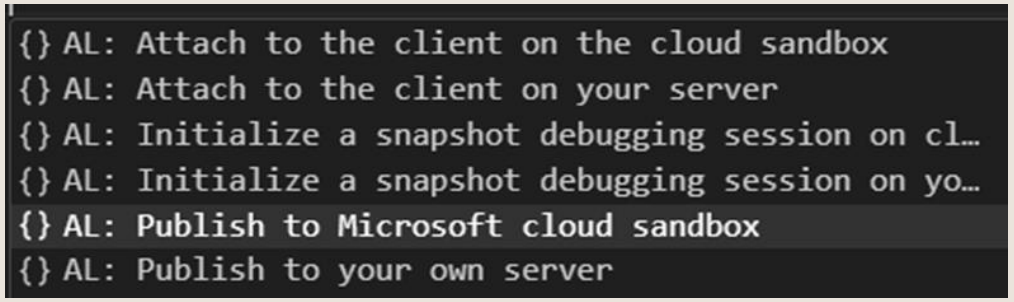
```
{
    "name": "Publish: Microsoft cloud sandbox",
    "type": "al",
    "request": "launch",
    "environmentType": "Sandbox",
    "environmentName": "sandbox",
    "startupObjectType": "page",
    "startupObjectId": 22,
    "tenant": "othertenant.onmicrosoft.com"
}
```

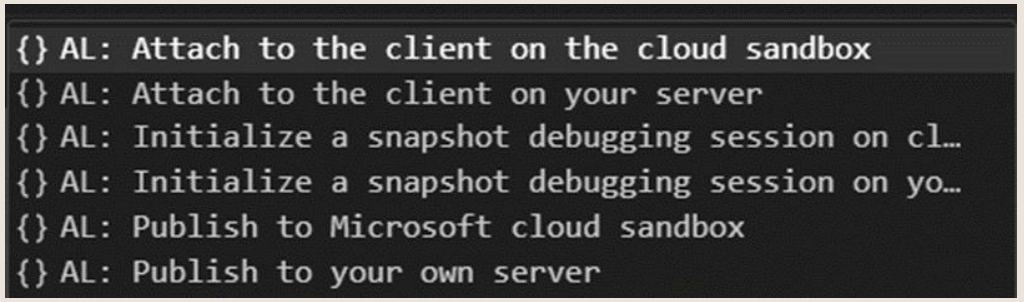
ATTATCH TO NEXT

```
{
    "name": "Attach: Microsoft cloud sandbox",
    "type": "al",
    "request": "attach",
    "environmentType": "Sandbox",
    "environmentName": "sandbox",
    "breakOnNext": "WebServiceClient",
    "user": "myuser@mytenant.onmicrosoft.com"
}
```

ATTATCH TO EXISTING

```
{
    "name": "Attach: Microsoft cloud sandbox",
    "type": "al",
    "request": "attach",
    "environmentType": "Sandbox",
    "environmentName": "sandbox",
    "sessionId": 45355
}
```





Demo

Debugging another user's session

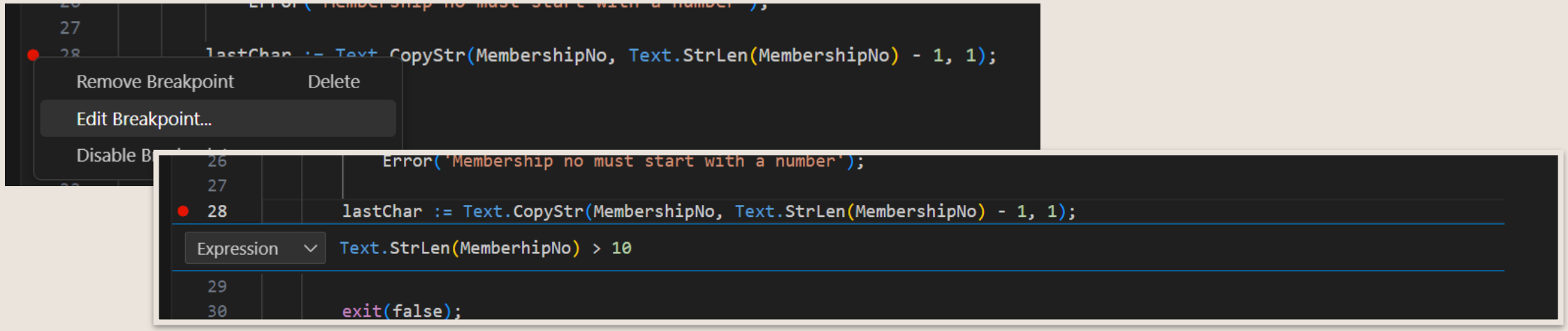
Microsoft Dynamics 365

Breakpoints

- Regular breakpoints
- Conditional breakpoints
- Implicit breakpoints
  - Break on error
  - Break on record write

Microsoft Dynamics 365

Conditional breakpoints



Microsoft Dynamics 365

Implicit breakpoints

```
{
    "name": "Publish: Microsoft
  cloud sandbox",
    "type": "al",
    "request": "launch",
    "environmentType": "Sandbox",
    "environmentName": "sandbox",
    "startupObjectType": "page",
    "startupObjectId": 22,
    "tenant": "othertenant.onmicrosoft.com",
    "breakOnError": "All",
    "breakOnRecordWrite": "ExcludeTemporary",
}
```

"**`breakOnError`**": what to do when it encounters an error.

- `All` – on every error
- `ExcludeTry` – on any error that is **not** within the scope of a try function
- `None` – do not break on any error

"**`breakOnRecordWrite`**": what to do when a record is modified.

- `All` – whenever **any** record is modified
- `ExcludeTemporary` – whenever a non-temporary record is modified
- `None` – do not break when a record is modified

Demo

Implicit breakpoints: break when a record is modified

Navigate and Debug in VS Code from Web Client

From the labs

Explorer

Productivity boosters

- Code Actions
- Go To Implementation
- Type Hierarchy
- Semantic code coloring
- Sticky Scroll
- Global Launch Config

Productivity boosters

- **Code Actions**
- Go To Implementation
- Type Hierarchy
- Semantic code coloring
- Sticky Scroll
- Global Launch Config

Microsoft Dynamics 365

Code Actions

Code Fixers

- Explicit With
- Qualify Implicit With
- Implement Interface
- Spell Check

Code Refactorings

- Promoted Action
- Application Area
- Event Subscriber literals

Code Cop

- AA0008 Use Parenthesis for Function calls
- AA0207 Make Procedure local
- AA0235 Add OnCompany Initialize Subscription
- AA0241 Use Lowercase For Keywords

UI Cop

- 0013 Hidden group with Promoted actions

Productivity boosters

- Code Actions
- **Go To Implementation**
- Type Hierarchy
- Semantic code coloring
- Sticky Scroll
- Global Launch Config

Productivity boosters

- Code Actions
- Go To Implementation
- **Type Hierarchy**
- Semantic code coloring
- Sticky Scroll
- Global Launch Config

Productivity boosters

- Code Actions
- Go To Implementation
- Type Hierarchy
- **Semantic code coloring**
- Sticky Scroll
- Global Launch Config

Productivity boosters

- Code Actions
- Go To Implementation
- Type Hierarchy
- Semantic code coloring
- **Sticky Scroll**
- Global Launch Config

Productivity boosters

- Code Actions
- Go To Implementation
- Type Hierarchy
- Semantic code coloring
- Sticky Scroll
- **Global Launch Config**

General Business Central resources, learn more!

Have a question?
aka.ms/BCYammer

Join the conversation
twitter.com/MSDyn365BC

Looking for resources?
aka.ms/BCAll

Submit your ideas
aka.ms/BCIdeas

Join the office hours
aka.ms/BCOfficeHours

Report an issue
Github.com/Microsoft/AL

For more questions, meet us at the Microsoft booth tomorrow at 8:30, 13:00 or 15:00

Thank You!