

What's new in Server and Database



Agenda

- a faster runtime
- a faster data stack
- more stable web services
- new reporting capabilities

What's new in Server and Database

A faster runtime



Agenda

- **a faster runtime**
- a faster data stack
- more stable web services
- new reporting capabilities

ErrorInfo and Notifications

In ErrorInfo datatype

- AddAction now supports [tooltips](#)
- AddAction now checks if you try to call an on-premises function

Action will not be added if not allowed

Source code example: ErrorInfo in Approvals Mgmt.

```
ErrInfo.AddAction( RejectApprovalRequestLbl  
, Codeunit::"Approvals Mgmt."  
, 'RejectApprovalRequest'  
, RejectApprovalRequestToolTipLbl  
);
```

```
ErrInfo.AddAction( ShowCommentsLbl  
, Codeunit::"Approvals Mgmt."  
, 'ShowApprovalCommentLinesForJournal'  
, ShowCommentsToolTipLbl  
);
```

This may also affect you

SecretText type

See other *What's new in AL* session

Effective Permissions page shows permissions from Security Groups

Navigate to Query in client

Open query URL:
`/?query=<query object id>`

New: *Analysis* toggle button – opens new UI

New: Support for [RunObject](#)

Runobject on a query

```
pageextension 50110 ItemListWithQuery extends "Item List" {  
    actions {  
        addbefore("Inventory - List") {  
            action("Analyze items") {  
                ApplicationArea = All;  
                Caption = 'Analyze Item ledger';  
                RunObject = query "Item hero query";  
                Tooltip = Analyze further';  
                Image = Item;  
            }  
        }  
    }  
}
```

Long running AL telemetry now handles suspended time

Partners feedback: event RT0018 does not provide value since it included time spent in modal events.

Therefore, we now also calculate the time excluding the time in modal events.

New dimension - exclusiveTime

```
7 | where customDimensions.eventId == 'RT0018'  
8 | where customDimensions.alObjectId > 0 // filter out internal server calls  
9 | project aadTenantId = customDimensions.aadTenantId  
10 , environmentName = customDimensions.environmentName  
11 , environmentType = customDimensions.environmentType  
12 , companyName = customDimensions.companyName  
13 , alMethod = customDimensions.alMethod  
14 , alObjectId = customDimensions.alObjectId  
15 , alObjectName = customDimensions.alObjectName  
16 , alObjectType = customDimensions.alObjectType  
17 , alStackTrace = customDimensions.alStackTrace // use the KQL snippet parseStackTrace to get top/bottom of the stack trace details  
18 , clientType = customDimensions.clientType  
19 , exclusiveTime = customDimensions.exclusiveTime // This dimension was introduced in Business Central 2023 release wave 1, version 22.1. Backported to version 21.6  
20 , exclusiveTimeInMS = toreal(totimespan(customDimensions.exclusiveTime))/10000 //the datatype for exclusiveTime is timespan  
21 , executionTime = customDimensions.executionTime  
22 , executionTimeInMS = toreal(totimespan(customDimensions.executionTime))/10000 //the datatype for executionTime is timespan  
23 , extensionId = customDimensions.extensionId  
24 , extensionInfo = customDimensions.extensionInfo // parse this json structure to find out if other extensions is involved  
25 , extensionName = customDimensions.extensionName  
26 , extensionPublisher = customDimensions.extensionPublisher  
27 , extensionVersion = customDimensions.extensionVersion  
28 , longRunningThreshold = customDimensions.longRunningThreshold  
29 , longRunningThresholdInMS = toreal(totimespan(customDimensions.longRunningThreshold))/10000 //the datatype for executionTime is timespan  
30 , sqlExecutes = customDimensions.sqlExecutes // This dimension was introduced in Business Central 2023 release wave 1, version 22.0  
31 , sqlRowsRead = customDimensions.sqlRowsRead // This dimension was introduced in Business Central 2023 release wave 1, version 22.0  
32 , usertelemetryId = case(  
33 // user telemetry id was introduced in the platform in version 20.0
```

Long running AL methods – total and exclusive time

Now shows both total time and exclusive time

The screenshot shows the Dynamics 365 Business Central Usage report interface. On the left, a navigation menu lists various performance-related sections. Two red arrows point to the 'Performance' section and the 'Long Running AL meth...' item under 'Incoming webservice calls'. The main content area displays 'AL method performance details' with a table titled 'AL Method Statistics' and several charts: 'Long running AL methods by Client Type', 'Long running AL methods by Extension type', 'Long running AL methods by Extension Publisher', and 'Long running AL methods over the Day - (UTC-06:00) Central Time (US & Canada)'.

AL Method Statistics

Object / Method	Count	Total sum time (sec)	Total max time (sec)	Exclusive sum time (sec)	Exclusive max time (sec)
BCPT Role Wrapper	25078	2156783	1864	1169960	1824
BCPT CommandLine Card	988	1519251	1864	845025	1824
BCPT Start Tests	986	1518891	1864	844667	1824
Test Runner - Mgt	24086	637427	279	324858	279
BCPT Create SO with N Lines	20251	427872	191	235720	181
BCPT Post Sales with N Lines	3472	154281	197	105021	153
Sales-Post	3912	128240	166	70924	166
Total	378429	10724351	38186	4127911	12957

Object Details (in which object does the endpoint come from)

Object Name / Id / Type	Count
Accessible Companies	6
Aged Acc. Receivable Chart	1
App Key Vault Secret Pr. Impl.	1
Apply Retention Policy	12
Apply Retention Policy Impl.	18
Assisted Company Setup	164
AttachmentManagement	1
Azure AD Graph Impl.	18
Azure AD Plan Impl.	1
Azure AD User Management	1
Total	378429

App Details (in which app does the endpoint come from)

Publisher / App (id) / Version	Count
BCTech	270
Continia Software	16
Default publisher	25762
DirectionsAsia 2023	20
EOS Solutions	8
Insight Works	6
Microsoft	68366
MSFT	26
NaviPartner	1
Total	378429

Long running AL methods by Client Type

Client type	Percentage
Background	77.46%
ChildSession	22.10%
ClientService	

Long running AL methods by Extension type

Code Ownership	Percentage
ENVIRONMENT	81.85%
ISV_ONLINE	
ISV_ONPREM	
MICROSOFT	18.13%

Long running AL methods by Extension Publisher

Publisher	Percentage
(Blank)	
BCTech	75.03%
Continia Software	18.07%

Long running AL methods over the Day - (UTC-06:00) Central Time (US & Canada)

A timeline chart showing the distribution of long running AL methods over a 24-hour period, centered on UTC-06:00 Central Time (US & Canada). The x-axis represents time from 00:00 to 23:00. The y-axis represents the number of methods. The chart shows a significant peak around 06:00 UTC-06:00, with a smaller peak around 18:00 UTC-06:00.

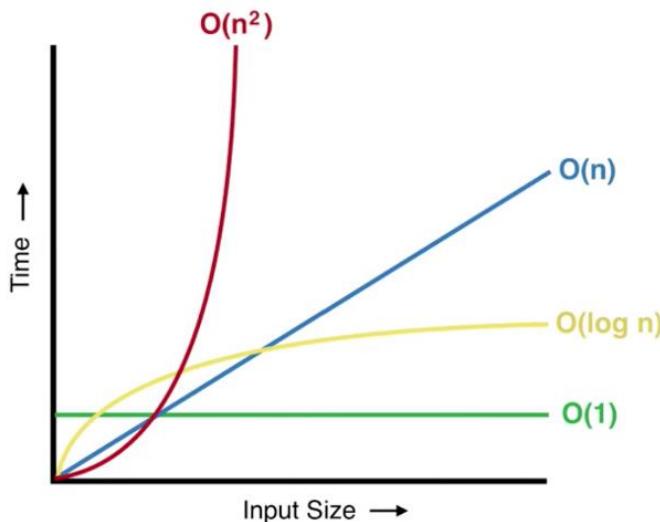
Session creation now 2 ms faster

What: Creating a new session is now 2-4 ms faster

How: Tenant level cache the companies that a user has access to

Fun fact: 5 hours per day for busiest customer

Event initialization optimized



When starting the NST, we need to create event subscribers.

Event subscription creation is now 1000x faster (for 1000 events), going from quadratic to linear growth.

Also, lower memory consumption.

Go dos

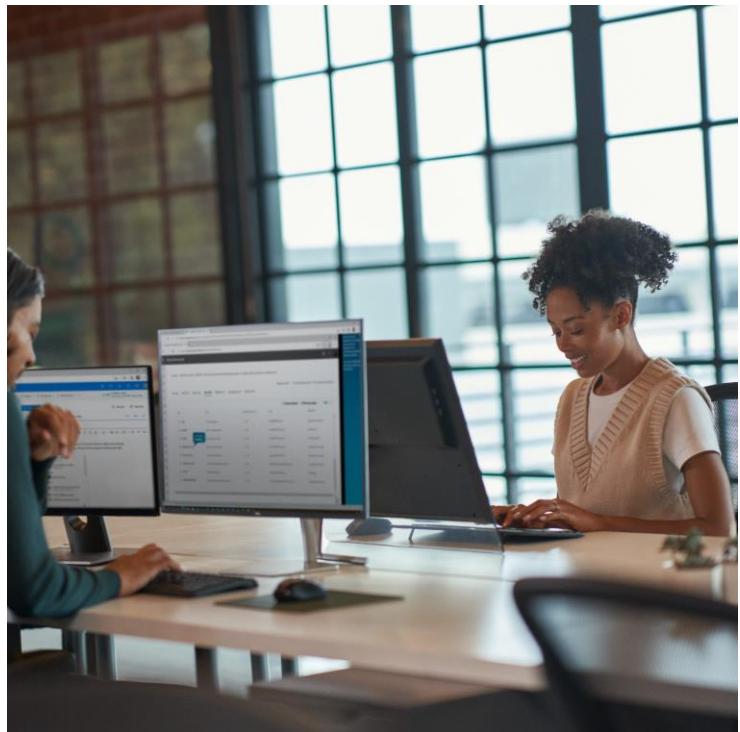
Stop using old Error method

- Use the new one that uses ErrorInfo

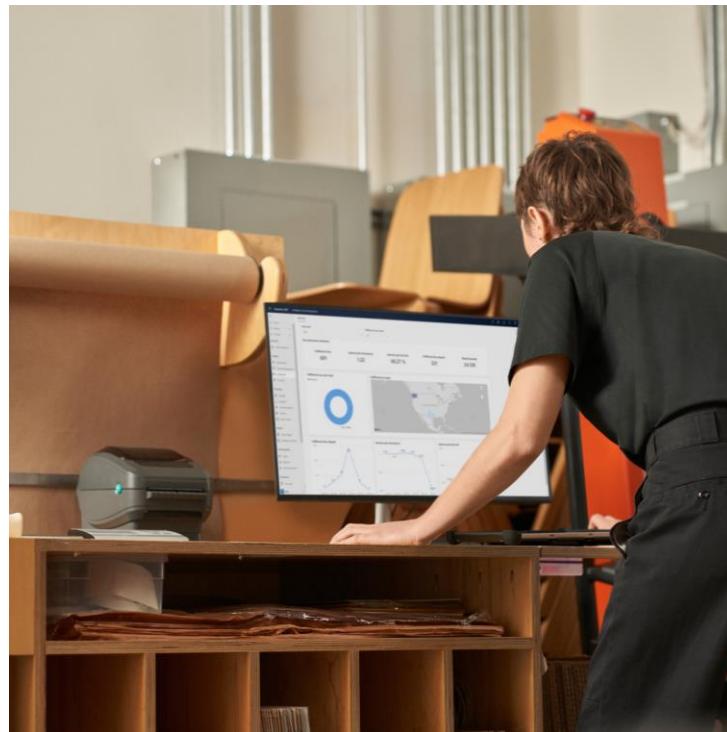
Harden your AL code

- Start using SecretText where appropriate

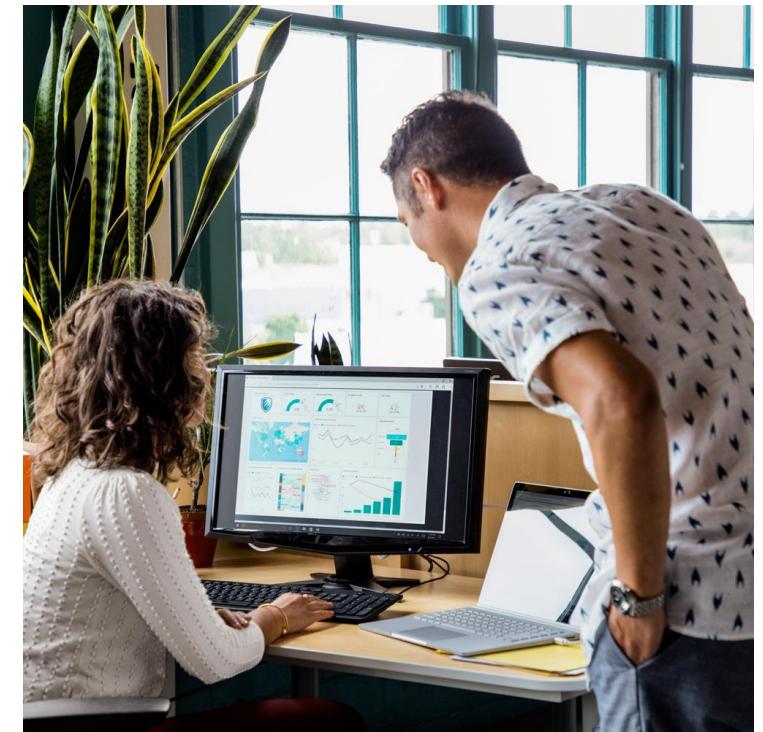
Watch other relevant launch event sessions



What's new: Developer tools

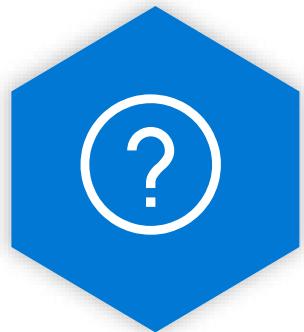


What's new: Dataverse & Dynamics 365 app integration - Part 2



What's new: Telemetry

General Business Central resources, learn more!

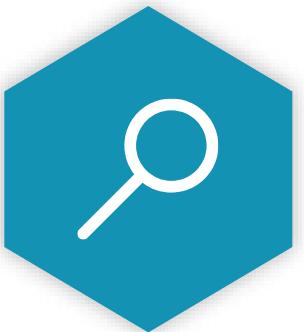


**Have a
question?**

aka.ms/BCYammer



**Join the
conversation**
[twitter.com/
MSDyn365BC](http://twitter.com/MSDyn365BC)

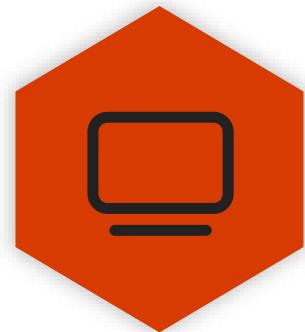


**Looking for
resources?**

aka.ms/BCAll



**Submit
your ideas**
aka.ms/BCIdeas



**Join the office
hours**

aka.ms/BCOfficeHours

Thank you

What's new in Server and Database

A faster data stack



Agenda

- a faster runtime
- **a faster data stack**
- more stable web services
- new reporting capabilities

How companion tables work in version 22 (and earlier)

Each table extension has data *in its own table*.

Runtime enforces a 1:1 relationship.

Primary key of base table is also PK for all companion tables.

Fetching data requires joining with *all extension tables*

- Execution time grows linearly with number of extensions
- SQL Server often 'gives up optimizing' when you have 8+ joins
- You can opt out of joining using *partial records*

Behaviour of previous data model

N-way join of 1-1 tables is slow.

No particular app/extension is to blame.

Cannot ask customers to uninstall functionality.

Only platform can solve the problem (so we did!).

New data model for table extensions

- Minimizes use of joins when working with table extensions.
- Delete performance (2x-5x faster)
- Insert performance (2x faster)
- Modify performance (1,5x-2,5x faster)
- Read performance (faster, but NST caches data)

Extension schema in version 22 and earlier

dbo.MyCompany\$TableName\$<baseAppGuid>

PK1	Field1	Field2	\$systemId	\$<audit fields>	rowVersion
V22	2023	Wave1	<guid1>	...	Xxxxxxxxx
V23	2023	Wave2	<guid2>	...	Yyyyyyyy

dbo.MyCompany\$TableName\$<Guid1>

PK1	YellowField	rowVersion
V22	Fast	Zzzzzzzz1
V23	Faster	Zzzzzzzz2

dbo.MyCompany\$TableName\$<Guid2>

PK1	RedField	rowVersion
V22	ReadIsolation	Wwwwww1
V23	New tooltips	Wwwwww2

Extension schema in version 23

dbo.MyCompany\$TableName\$<baseAppGuid>

PK1	Field1	Field2	\$systemId	\$<audit fields>	rowVersion
V22	2023	Wave1	<guid1>	...	Xxxxxxxxx
V23	2023	Wave2	<guid2>	...	Yyyyyyyy

dbo.MyCompany\$TableName\$<baseAppGuid>\$ext

PK1	YellowField\$Guid1	RedField\$Guid2	rowVersion
V22	Fast	ReadIsolation	Vvvvvvvv1
V23	Faster	New tooltips	Vvvvvvvv2

Expected performance impact

Main improvement: read performance

At most 1 join

No triggers to run

After 1st join, duration only depends on data size

Modify/Insert/Delete

Harder to do in bulk

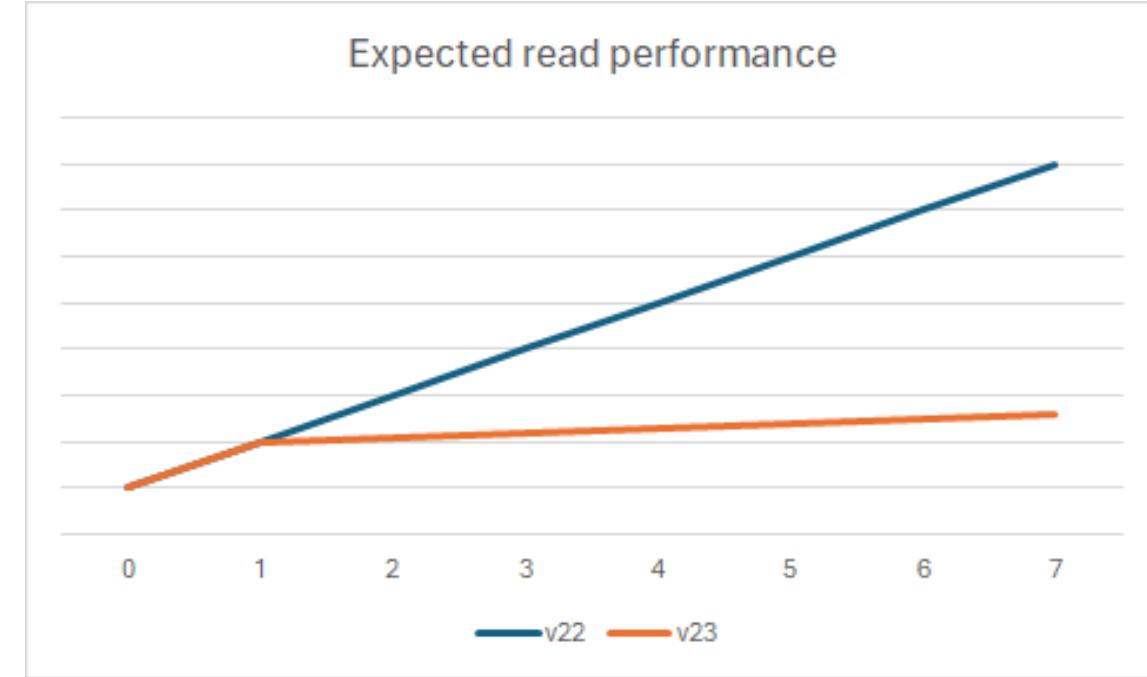
Often determined by triggers / events

Mileage will vary

Field counts and sizes

Trigger code

Indexes



Cloud migration and new table schema

Cloud migration relies on SQL schema + system tables.

Version 23 exposes the mapping from previous versions.

Updates to the cloud migration tool (ADF flows) have been shipped.

Cloud migration is fully supported – also from previous versions.

But we have data warehouse/business intelligence on-premises that read directly on the database

Mimic the old schema with database views.

Find code samples on Github.

Is SetLoadFields still need? YES!

Removing JOIN statements

Can still remove the one remaining JOIN statement

Limits fields loaded

Less data to transfer and cache

Ability to hit covering index

So: continue using SetLoadFields

Find performance critical code

Create covering indexes

New method

SetBase LoadFields

`SetBaseLoadFields` adds all fields from base table to loaded fields

Developers are lazy (this is a good thing).

Lower maintenance cost if making changes later.

SetBaseLoadFields

```
VATEntry.Reset();
VATEntry.SetBaseLoadFields();
VATEntry.SetRange("Transaction No.", GLEntry."Transaction No.");
VATEntry.SetRange("VAT Bus. Posting Group", GLEntry."VAT Bus. Pos-
ting Group");
VATEntry.SetRange("VAT Prod. Posting Group", GLEntry."VAT Prod. P-
osting Group");

if VATEntry.FindSet() then begin
// code continues here
```

Key takeaways on new extension schema

No action required on partners

Performance improvements with 2+ extensions

Mainly on read performance

Results will vary based on extension count, field sizes and indexes

Cloud migration fully supported

Only on-premises: Work to do for solutions accessing SQL tables directly.

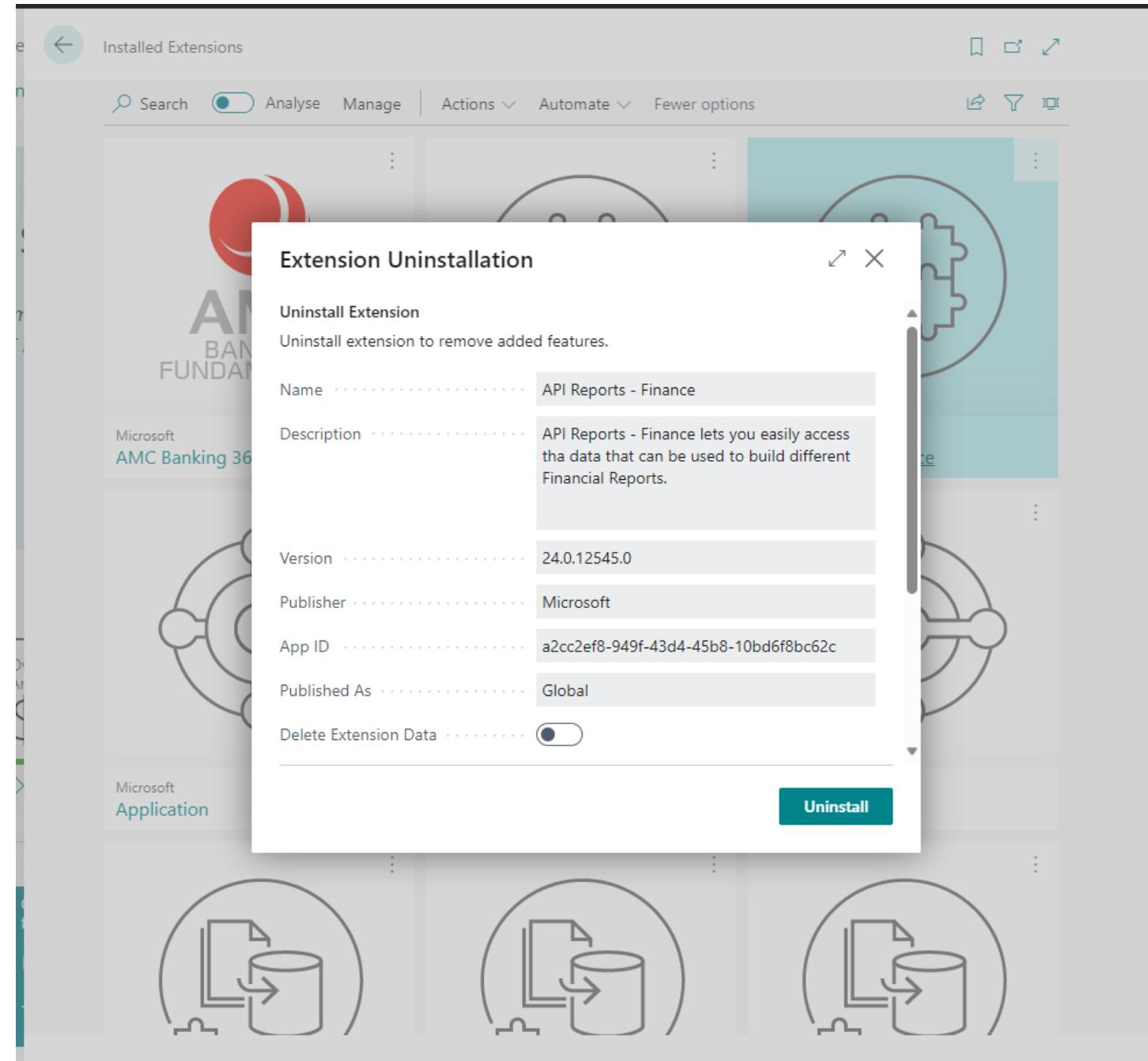
Deleting data from orphaned extensions

Admins can delete data from un-installed extensions.

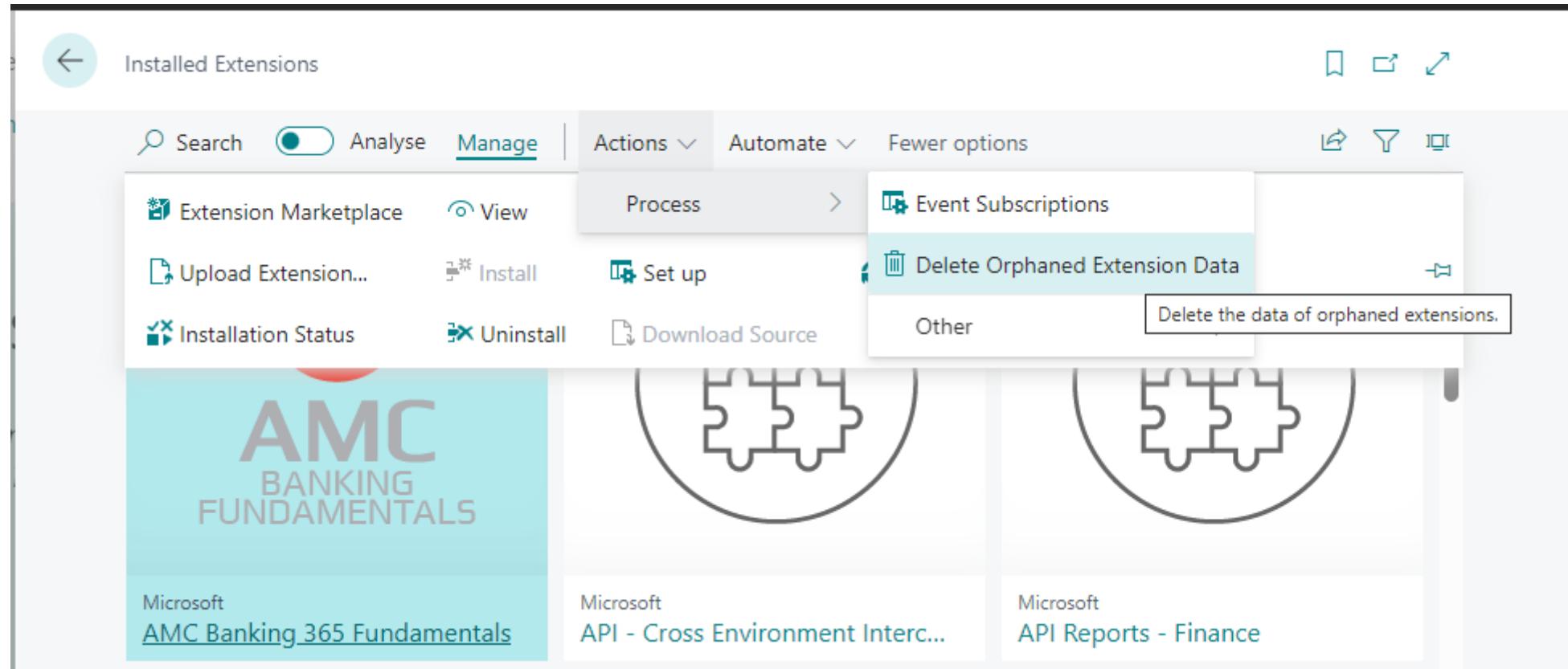
Frees up database capacity.

Reduces row size in the extension table.

If you forget to mark “Delete Extension Data”



Deleting data from orphaned extensions

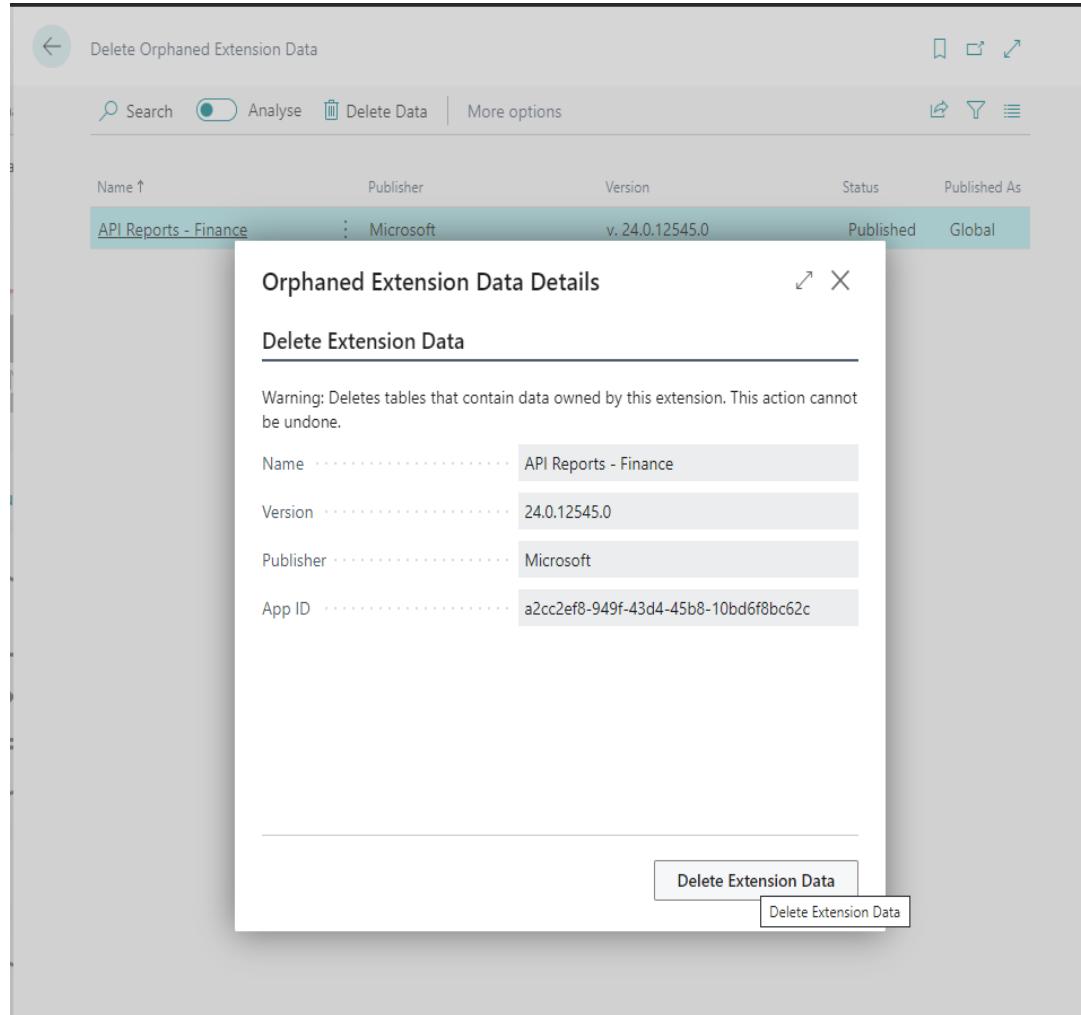


Find the orphaned extension here

The screenshot shows a web-based administrative interface for managing extensions. At the top left is a back arrow icon and the text "Delete Orphaned Extension Data". On the right are three small blue icons: a clipboard, a square with a checkmark, and a circular arrow. Below this is a horizontal toolbar with a search icon, an "Analyse" toggle switch (which is turned on), a "Delete Data" button, and a "More options" dropdown. To the right of the toolbar are three more icons: a refresh, a filter, and a list. The main area is a table with the following columns: Name, Publisher, Version, Status, and Published As. A single row is highlighted in light blue, representing the extension "API Reports - Finance" published by Microsoft with version v. 24.0.12545.0, status Published, and published as Global.

Name ↑	Publisher	Version	Status	Published As
API Reports - Finance	Microsoft	v. 24.0.12545.0	Published	Global

Deleting data from orphaned extensions – and delete the data



New locking model

Minimizes use of database locks in AL.

Benefits all usage scenarios: web service integrations, background jobs, and interactive sessions.

Improves throughput and responsiveness of the service and reduces likelihood of deadlocks and lock timeouts.

Example - before

```
trigger OnAction()
var
    curr1: Record Currency;
    curr2: Record Currency;
begin
    curr1.FindFirst(); // READUNCOMMITTED

    curr1.Code := 'BTC';
    curr1."ISO Code" := 'XBT';
    curr1.Symbol := '฿';
    curr1.Insert();

    curr2.FindLast(); // UPDLOCK
end;
```

Example – with new locking enabled

```
trigger OnAction()
var
    curr1: Record Currency;
    curr2: Record Currency;
begin
    curr1.FindFirst(); // READUNCOMMITTED

    curr1.Code := 'BTC';
    curr1."ISO Code" := 'XBT';
    curr1.Symbol := '฿';
    curr1.Insert();

    curr2.FindLast(); // READCOMMITTED
end;
```

New locking model – for a given record type

No writes or LockTable done to record type in current transaction

All reads have the READUNCOMMITTED hint applied

Data has been written to the record type in current transaction

Further reads will have the READCOMMITTED hint applied

If LockTable has been called on the record type in the current transaction

Further reads will have the UPDLOCK hint applied

Data written to the record type in current transaction

Before

- Default locking behavior was UpdLock.
- Session would acquire update lock on data from the table until it committed or rolled back its changes.
- Could cause blocking and contention issues when multiple sessions tried to access or modify the same table.

New locking

- Default locking behavior is ReadCommitted
- Session will only acquire a shared lock when reading data.
- Allows other sessions to read and write to the same table concurrently, if they do not conflict with each other's changes.

New locking behavior can be turned on/off

Go to the Feature Management page

Tip! Enable in sandboxes and run (BCPT?) tests

The screenshot shows the Dynamics 365 Business Central Feature Management page. The top navigation bar includes icons for Home, Search, Notifications, Settings, and Help, along with a user profile icon.

The main area displays a table with the following columns:

Feature	Automatically enabled from	Enabled for	Get started	Current Company Status	Update Start Date/Time
→ Feature: Enable Tri-State locking in AI	Update 25.0 (Q4 2024)	All Users	-	Enabled	

A dropdown menu is open over the "Enabled for" field, showing two options: "None" and "All Users".

At the bottom of the table, there are buttons for "Learn more" and "Edit List". The "Edit List" button is highlighted in blue.

New locking: Rollout

- Version 23 (2023 release wave 2)
 - Upgraded tenants: Opt-in via Feature Key.
 - New tenants: Already enabled.
 - Both can enable and disable at will.
- Version 24 (2024 release wave 1)
 - All tenants: Enabled by default.
 - Both can enable and disable at will.
- Version 25 (2024 release wave 2)
 - New locking is the only way.

Data stack performance improvements

Optimization: Faster inserts on tables with Auto-Increment field

Optimization: Faster SetRange(from, to) when to == from

Database changes coming in 2024 release wave 1

Server settings for database hints FORCEORDER and LOOPJOIN (removed)

Removed or Replaced?	Why?
Removed	For Business Central installations on-premises, it is possible to control many server settings. Two of these control the way the Business Central server add hints to SQL statements: DisableQueryHintForceOrder and DisableQueryHintLoopJoin. Starting in version 24, these server settings will no longer be available.

Remove write access to app database tables from AL (removal)

Removed or Replaced?	Why?
Removed	For Business Central on-premises, the main difference between running in single tenancy mode or multi tenancy mode is that in the former, it is possible to write to the app database tables from AL. Starting in version 24, this is no longer possible.

Go dos

Test new locking

- Run BCPT tests with/without new locking.
- Check for errors

Find and delete orphaned data

- Save on storage
- Improve performance

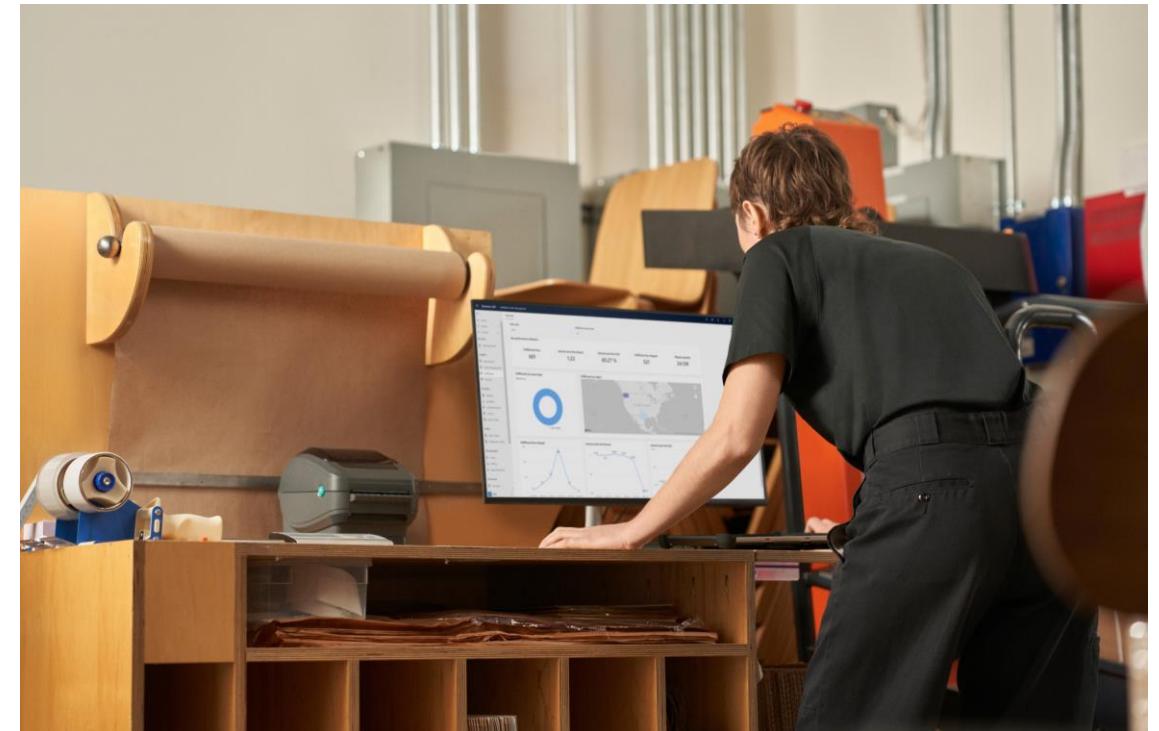
Check if you write to app db

- Only for on-premises: if you write to the app database, please stop. In 6 months, this will stop working.

Watch other relevant launch event sessions

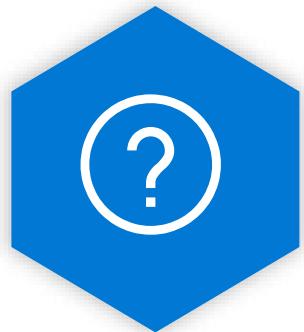


What's new: Developer tools



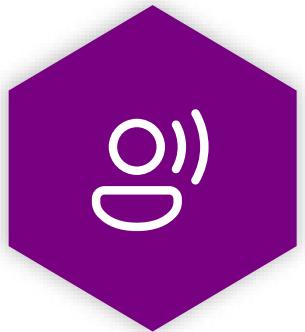
What's new: Server and database - new reporting capabilities

General Business Central resources, learn more!



**Have a
question?**

aka.ms/BCYammer

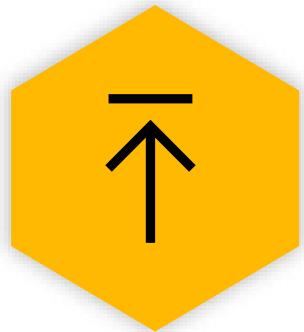


**Join the
conversation**
[twitter.com/
MSDyn365BC](http://twitter.com/MSDyn365BC)

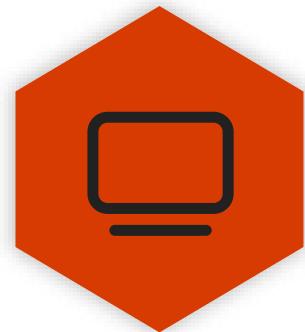


**Looking for
resources?**

aka.ms/BCAll



**Submit
your ideas**
aka.ms/BCIdeas



**Join the office
hours**

aka.ms/BCOfficeHours

Thank you

What's new in Server and Database

More stable web services



Agenda

- a faster runtime
- a faster data stack
- **more stable web services**
- new reporting capabilities

Edit-in-Excel supports more scenarios

ODataV4 pages now evaluate Editable
property like the UI does.

Before: only supported Boolean variables.

Example: Ship-to Code on Sales Orders page

```
field("Ship-to Code"; Rec."Ship-to Code")
{
    ApplicationArea = Basic, Suite;
    Caption = 'Code';
    Editable = ShipToOptions = ShipToOptions::"Alternate Shipping Address";
    ...
}
```

✖ Error Publishing Row

Details:

A row update in data set Sales_Order_Excel was not published. Error message: 'Property "Editable" for Ship-to Code is invalid. Expression: [p42p42ShipToOptions = 1]

**Web service
clients can
react on
timeouts (and
reschedule)**

If the server is too busy to run webservice requests, resulting in requests waiting in request queue to timeout (timeout is 8 minutes in the request queue), the **server now returns HTTP return code 503** (Service Temporarily Unavailable) along with a retry-after header.

Backported and available from 22.2

Webservice stack more robust towards metadata errors

If a single webservice endpoint has metadata errors, what do we do?

Before: all webservice calls would fail

Now: we ignore elements that cause errors and emit partner telemetry.

Backported and available from 22.2.

Analyze web service metadata errors in Power BI

The screenshot shows the Power BI Errors app interface. On the left, a navigation menu lists various error types, with 'Webservice metadata er...' highlighted by a red arrow. The main area displays three cards: 'Web Service Metadata Errors by Environment' (5 errors total, 1 in Sandbox, 4 in common), 'Web Service Metadata Errors by Error Type' (1 Invalid WebService Metadata, 4 Invalid Record), and 'Web Service Errors by Error Type' (80.00% Invalid Record, 20.00% Invalid WebService Metadata). Below these is a table titled 'Webservice Metadata Error log' showing timestamp, Event Id, Error type, and Metadata stack for specific error entries.

These errors occur when a web service endpoint could not be published or if a published web service endpoint isn't working correctly due to an error in creating metadata for it.

Web Service Metadata Errors by Environment

Count	Domain	AAD Tenant Id	Environment Name
1		c17619aa-4729-4dbb-9bd8-07aa663d7fa6	Sandbox
4		common	
5			

Web Service Metadata Errors by Error Type

Count	Event Id	Error type	Severity
1	RT0039	Invalid WebService Metadata	Error
4	RT0035	Invalid Record	Error
5			

Web Service Errors by Error Type

Error type	Percentage
Invalid Record	80.00%
Invalid WebService Metadata	20.00%

Webservice Metadata Error log

timestamp	Event Id	Error type	Metadata stack
16/06/2023 09:01:06	RT0039	Invalid WebService Metadata	
16/06/2023 09:01:44	RT0035	Invalid Record	Record: 1314
16/06/2023 09:02:00	RT0035	Invalid Record	Record: 1315
16/06/2023 09:02:12	RT0035	Invalid Record	Record: 1544
30/06/2023 09:14:30	RT0035	Invalid Record	

Find web service endpoints that fail at runtime and how to fix them

New TSG in docs

The screenshot shows a Microsoft Learn page for Dynamics 365 Business Central. The top navigation bar includes links for Microsoft Learn, Documentation, Training, Credentials, Q&A, Code Samples, Assessments, and Shows. Below the navigation is a breadcrumb trail: Dynamics 365 Business Central > Get started > Guidance > Troubleshooting > Release plans > Support > Resources. A search bar labeled "Filter by title" is present. The main content area has a sidebar with sections for Integration, Web services, and General, each with several sub-links. The main article title is "Troubleshooting web service errors". It includes a "Feedback" link, a "In this article" section with links to "Web service telemetry" and "Troubleshooting OData/API calls", and a "See also" section. The main text discusses HTTP status codes and provides a table of common 4xx errors with suggested solutions. The "HTTP status code" column is highlighted with a red box.

Troubleshooting web service errors

Article • 08/22/2023 • 2 contributors [Feedback](#)

In this article

[Web service telemetry](#)
[Troubleshooting OData/API calls](#)
[See also](#)

When you call a web service endpoint, either a Business Central API or from AL using `Httpclient` datatype, you get an HTTP status code as part of the response. All HTTP status codes that start with 4 (sometimes also written 4xx) are classified as client errors and it is your responsibility to react on these errors and fix them in your code.

In the following table, we list some common 4xx HTTP status codes and suggestions to how to fix them:

HTTP status code	Short name	Description	Suggested solution(s)
400	Bad Request	This status code indicates that the server can't or won't process the request due to an error on the client side. For example, it could be a malformed request syntax, header too long, or missing required parameters.	The client needs to fix things on their end. For an incoming call of category OData/API, consider using telemetry to find the error. You can also set up a debugger and debug the endpoint code. For an outgoing call, you need to review/debug the AL code that sends the request.

Troubleshoot web service calls

BCTechdays 2023

Errors

Dashboard

Login errors

Error dialogs

Permission Errors

Error messages

Feature Errors

Environment lifecycle err...

Company lifecycle errors

Extension lifecycle errors

Email errors

Authentication (web serv...

Incoming webservice er...

Connector errors

Outgoing webservice err...

Job Queue errors

Task Scheduler errors

Report errors

Go back

Object Details (in which object does the endpoint come from?)

Object Name / Id / Type	Count
Accountant Portal Finance Cues	5
Accountant Portal User Tasks	5
AccountantPortal Activity Cues	5
External Event Subscription	5
Total	20

App Details (in which app does the endpoint come from?)

Publisher / App (id) / Version	Count
Microsoft	15
Total	20

User Agents (which clients call the endpoints?)

User Agent	Count
["BusinessCentralCDSConnector/1.0.23060.1"]	5
Total	20

Which web service client experienced issues?

Failure Reasons

Count	Failure Reason	Diagnostics Message
5	Microsoft.Dynamics.Nav.Service.OData.NavODataBadRequestException	
15	Microsoft.Dynamics.Nav.Types.Exceptions.NavCSideException	While opening company 'My Company', the following error occurred: 'Preview versions are no longer available. To continue using %1, use a web client to open the company and accept the terms and conditions.' AL stack trace: "SaaS Log In Management"(CodeUnit 50).ShouldShowTermsAndConditions line 31 - Base Application by Microsoft "SaaS Log In Management"(CodeUnit 50).ShowTermsAndConditionsOnOpenCompany line 4 - Base Application by Microsoft "SaaS Log In Management"(CodeUnit 50).OnShowTermsAndConditionsSubscriber line 2 - Base Application by Microsoft LoginManagement(CodeUnit 40).OnShowTermsAndConditions(Event) line 2 - Base Application by Microsoft

Easier troubleshooting of incoming web service errors

The screenshot shows the Dynamics 365 Business Central Errors page with several highlighted features:

- Improved filters on HTTP status codes**: A red box highlights the **Filters** pane on the right, which includes sections for **HTTP response type**, **HTTP status code**, and **HTTP status description**.
- Two new visuals that show HTTP status code view on errors**: Red boxes highlight two new visualizations:
 - Incoming web service call errors (endpoints)**: A table showing endpoint categories and error counts. One row is expanded to show detailed endpoint information.
 - Incoming web service call errors over the day - (UTC-06:00) Central Time (US & Canada)**: A histogram showing the distribution of errors over time.
- Incoming web service call errors (HTTP status codes)**: A table showing a list of endpoints, their corresponding HTTP status codes, and descriptions. A red arrow points to this table from the left sidebar.
- User Agents (which clients call the endpoints?)**: A table showing user agent names and their counts.

Left Sidebar (highlighted by red arrows):

- Errors
- Incoming webservice er...
- Connector errors
- Outgoing webservice err...

Easier troubleshooting of outgoing web service errors

The screenshot shows the Dynamics 365 Business Central Errors page. A red arrow points from the left navigation bar to the 'Outgoing webservice errors' section. Another red arrow points from the bottom of the same section to a callout box.

Outgoing webservice call error statistics (endpoints)

Url / HTTP method	4xx: Client Error	Total
https://10052022-onbuyuka.myshopify.com	1	1
https://aibuildertextapiservice.us-il103.gateway.prod.island.powerapps.com	1	1
https://aibuildertextapiservice.us-il109.gateway.prod.island.powerapps.com	2	2
https://api.stripe.com	4	4
GET	4	4
Total	8	8

Outgoing webservice call error details (HTTP status codes)

Count	Url	HTTP Method	HTTP response type	HTTP status code	Description
1	https://10052022-onbuyuka.myshopify.com	DELETE	4xx: Client Error	404	Not Found
1	https://aibuildertextapiservice.us-il103.gateway.prod.island.powerapps.com	POST	4xx: Client Error	400	Bad Request
2	https://aibuildertextapiservice.us-il109.gateway.prod.island.powerapps.com	POST	4xx: Client Error	400	Bad Request
4	https://api.stripe.com	GET	4xx: Client Error	404	Not Found
8					

Web Service Errors by Error type

Error type
● 4xx: Client Error 100.00%

Web Service Errors by Http Method

HTTP Method
● DELETE 12.50%
● GET 50.00%
● POST 37.50%

Webservice errors over the Day - (UTC-06:00) Central Time (US & Canada)

0 5 10 15 20
2 4 2

Web Service Errors over the Week

0 2 4
Mon Tue Wed Thu Fri Sat Sun
2 2 2 2 2 2 2

Filters

HTTP response type is (All)
Filter type Basic filtering
Search
 Select all
 4xx: Client Error - T... 31
 5xx: Server Error - T... 12

HTTP scheme is (All)
Filter type Basic filtering
Search
 Select all
 https 24

HTTP status code is (All)
Filter type Basic filtering
Search
 Select all
 400 2
 404 5

Improved filters on HTTP status codes

Two new visuals that show HTTP status code view on errors

Does your API calls get queued up?

The screenshot shows the Dynamics 365 Business Central Performance dashboard for Usage on 29/05/2023. A red box highlights the 'Webservice Queue Statistics' section and the 'Queue time statistics, anyone?' text overlay.

Webservice Queue Statistics

Count	Endpoint	Sum queue time (sec)	Avg queue time (ms)
7	MS/api/microsoft/cloudMigration/v1.0/companies()/cloudMigrationStatus	0.00	0.00
7	MS/api/microsoft/cloudMigration/v1.0/companies()/cloudMigrationStatus()/.Microsoft.NAV.refreshStatus	0.00	0.00
1	MS/api/microsoft/dataverse/v1.0/dataverseEntityChanges	0.00	0.00
11	MS/api/microsoft/powerbi/v2.0/reportLabels	0.00	0.00
33	MS/api/microsoft/runtime/beta/apiRoutes	0.00	0.00
837	MS/api/microsoft/runtime/beta/companies	0.00	0.00
2	MS/api/microsoft/runtime/beta/webhookSupportedResources	0.00	0.00
8650		0.00	0.00

Queue time (sec) and Execution time (sec)

● Queue time (sec)
● Execution time (sec)

4K (100%)

Web service time (avg. execution time) by date

5K

Web service time (avg. queue time) by date

1

0

Queue time statistics, anyone?

Canada)

Calls over the Week

4K

2K

0K

Mon Tue Wed Thu Fri Sat Sun

Time spent over the Week

1K

0K

Mon Tue Wed Thu Fri Sat Sun

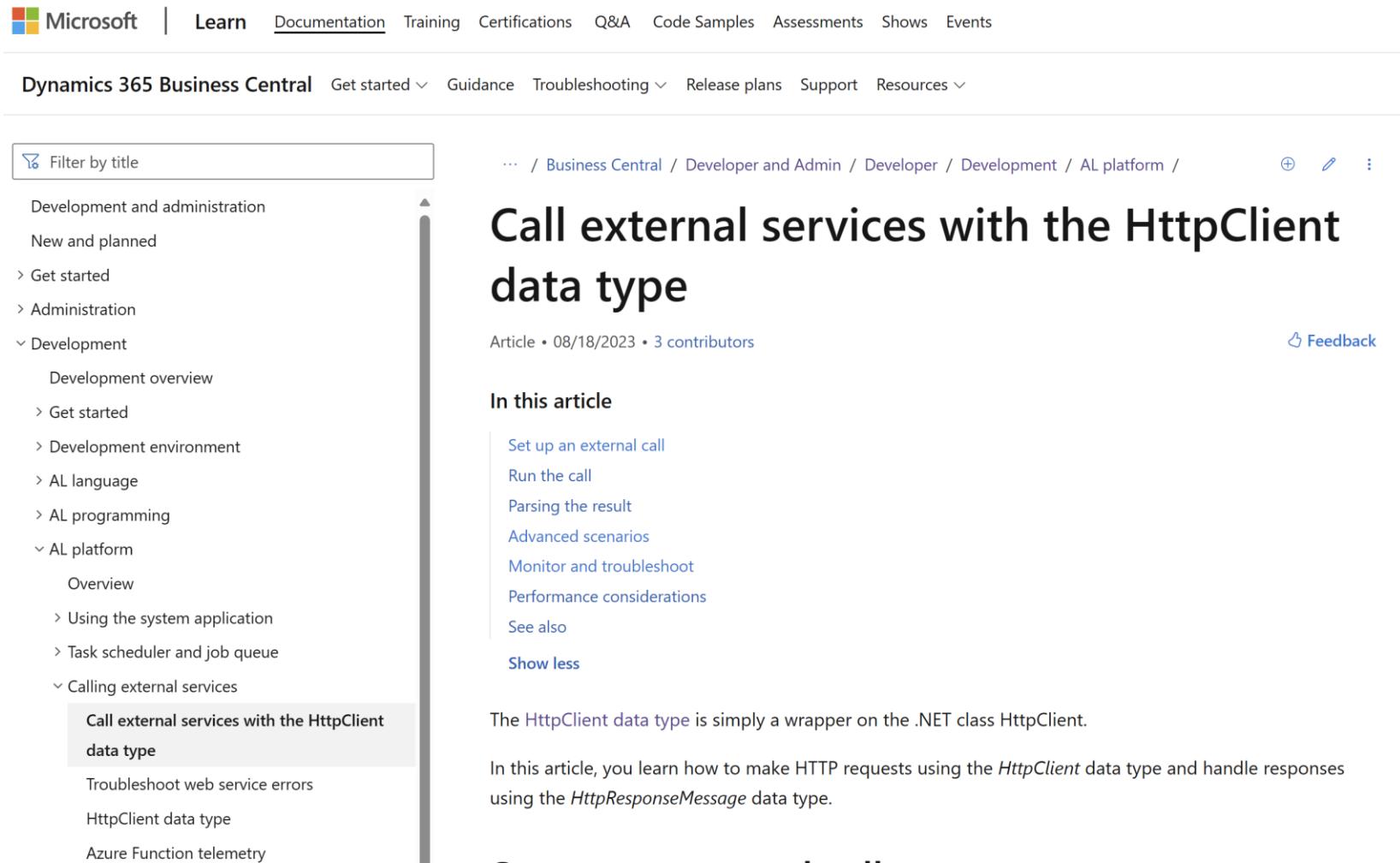
**AL developers
can write more
resilient AL
code when
calling
webservices**

Error handling best practices for Httpclient
now part of docs.

TSGs added to docs for better
troubleshooting of WS calls.

Huge makeover of Httpclient docs.

New e2e article on how to use HttpClient



The screenshot shows a Microsoft Learn article titled "Call external services with the HttpClient data type". The page is part of the "Development and administration" section under "Development". The article was published on 08/18/2023 and has 3 contributors. It includes sections on setting up an external call, running the call, parsing the result, advanced scenarios, monitoring and troubleshooting, and performance considerations. The "Show less" link is visible at the bottom of the article summary.

Microsoft | Learn Documentation Training Certifications Q&A Code Samples Assessments Shows Events

Dynamics 365 Business Central Get started Guidance Troubleshooting Release plans Support Resources

Filter by title

Development and administration

New and planned

> Get started

> Administration

Development

- Development overview
- > Get started
- > Development environment
- > AL language
- > AL programming
- AL platform

 - Overview
 - > Using the system application
 - > Task scheduler and job queue

- Calling external services

 - Call external services with the HttpClient data type
 - Troubleshoot web service errors
 - HttpClient data type
 - Azure Function telemetry

... / Business Central / Developer and Admin / Developer / Development / AL platform /

Call external services with the HttpClient data type

Article • 08/18/2023 • 3 contributors

Feedback

In this article

- Set up an external call
- Run the call
- Parsing the result
- Advanced scenarios
- Monitor and troubleshoot
- Performance considerations
- See also

Show less

The HttpClient data type is simply a wrapper on the .NET class HttpClient.

In this article, you learn how to make HTTP requests using the *HttpClient* data type and handle responses using the *HttpResponseMessage* data type.

We explain how HttpClient.Send calls can fail

Ways that HttpClient.Get calls can fail

The method HttpClient.Get can fail and return false in the following ways:

- The requestUri is not an absolute URI.
- The request failed due to an underlying issue such as network connectivity, DNS failure, server certificate validation or timeout.
- The request failed due to timeout.

 **Important**

Outbound HTTP calls from extensions are blocked by default and must be approved for each extension, otherwise instead of an external call, the system will display the following error message: The request was blocked by the runtime to prevent accidental use of production services.

To enable outbound HTTP calls, go to the **Extension Management** page in Business Central, and choose **Configure**. Then, on the **Extension Settings** page, make sure that **Allow HttpClient Requests** is selected. This setting must be enabled for each extension, including libraries.

We show how to implement error handling

Example (HTTP GET)

A GET request shouldn't send a body and is used (as the method name indicates) to retrieve (or get) data from a resource. To make an HTTP GET request, given an HttpClient and a URI, use the HttpClient.Get method:

```
AL

local procedure GetRequest() ResponseText: Text
var
    Client: HttpClient;
    IsSuccessful: Boolean;
    Response: HttpResponseMessage;
    ResponseText: Text;
begin
    IsSuccessful := Client.Get('https://jsonplaceholder.typicode.com/todos/3', Response);

    if not IsSuccessful then begin
        // handle the error
    end;

    if not Response.IsSuccessStatusCode() then begin
        HttpStatusCode := response.HttpStatusCode();
        // handle the error (depending on the HTTP status code)
    end;

    Response.Content().ReadAs(ResponseText);

    // Expected output:
    //   GET https://jsonplaceholder.typicode.com/todos/3 HTTP/1.1
    //   {
    //     "userId": 1,
    //     "id": 3,
    //     "title": "fugiat veniam minus",
    //     "completed": false
    //   }
end;
```

We show how to check HTTP status codes for errors

Parsing the result

If the HttpClient call succeeds, you get a response back from the service you called. The response is saved in the data type HttpResponseMessage, where you can parse it to use the information in your app. The service call itself might not succeed, so make sure that you check the HTTP status code in your AL code.

The following example illustrates the error handling you need to setup for handling errors from the service that you called.

```
AL Copy

local procedure GetRequest() ResponseText: Text
var
    Client: HttpClient;
    IsSuccessful: Boolean;
    ServiceCallErr: Label 'Web service call failed.';
    ErrorInfoObject: ErrorInfo;
begin
    IsSuccessful := Client.Get('https://httpcats.com/418.json', HttpResponseMessage);

    if not IsSuccessful then begin
        // handle the error
    end;

    if not HttpResponseMessage.IsSuccessStatusCode() then begin
        HttpStatusCode := HttpResponseMessage.StatusCode();
        ErrorInfoObject.DetailedMessage := 'Sorry, we could not retrieve the cat info right now.';
        ErrorInfoObject.Message := Format(ServiceStatusErr, HttpStatusCode, HttpResponseMessage);
        Error(ErrorInfoObject);
    end;

    HttpResponseMessage.Content().ReadAs(ResponseText);
end;
```

We explain common HTTP status codes for client errors

When you call a web service endpoint, either a Business Central API or from AL using `Httpclient` datatype, you get an HTTP status code as part of the response. All HTTP status codes that start with 4 (sometimes also written 4xx) are classified as client errors and it is your responsibility to react on these errors and fix them in your code.

In the following table, we list some common 4xx HTTP status codes and suggestions to how to fix them:

HTTP status code	Short name	Description	Suggested solution(s)
400	Bad Request	This status code indicates that the server can't or won't process the request due to an error on the client side. For example, it could be a malformed request syntax, header too long, or something else.	The client needs to fix things on their end. For an incoming call of category OData/API, consider using telemetry to find the error. You can also set up a debugger and debug the endpoint code. For an outgoing call, you need to review/debug the AL code that sends the request.
401	Access denied	The request failed because it lacks valid authentication credentials for the target resource.	For an incoming call, this is an authorization issue on the Business Central end of this, either in the <code>OnOpenCompany</code> trigger or a permission issue. For an outgoing call, you need to examine the AL code that sets up certificates or sets authorization header(s).
402	Payment Required	Indicates that the caller must make a payment to access the requested resource. Typically used in situations where the server requires payment before granting access to the content or service. This status code isn't returned by the Business Central server for incoming calls.	For an outgoing call, you need to examine the AL code that calls the service and setup error handling to handle this situation.
403	Forbidden	This status code is returned when there's some kind of access restriction policy implemented for the requested resource.	For an outgoing call, you need to examine the AL code that sets up certificates or sets authorization header(s). Or maybe your app shouldn't call the specified endpoint at all.
404	Not found	The resource you're calling	

AL developers can understand exceptions when calling webservices

When `HttpClient.Send` fails, we now allow AL developers to see the root cause

- Malformed URI
- Network error (it is always DNS)
- Network error (timeout)
- Network error (connectivity)
- Certificate errors

Source code example: better AL error reporting for HttpClient failures

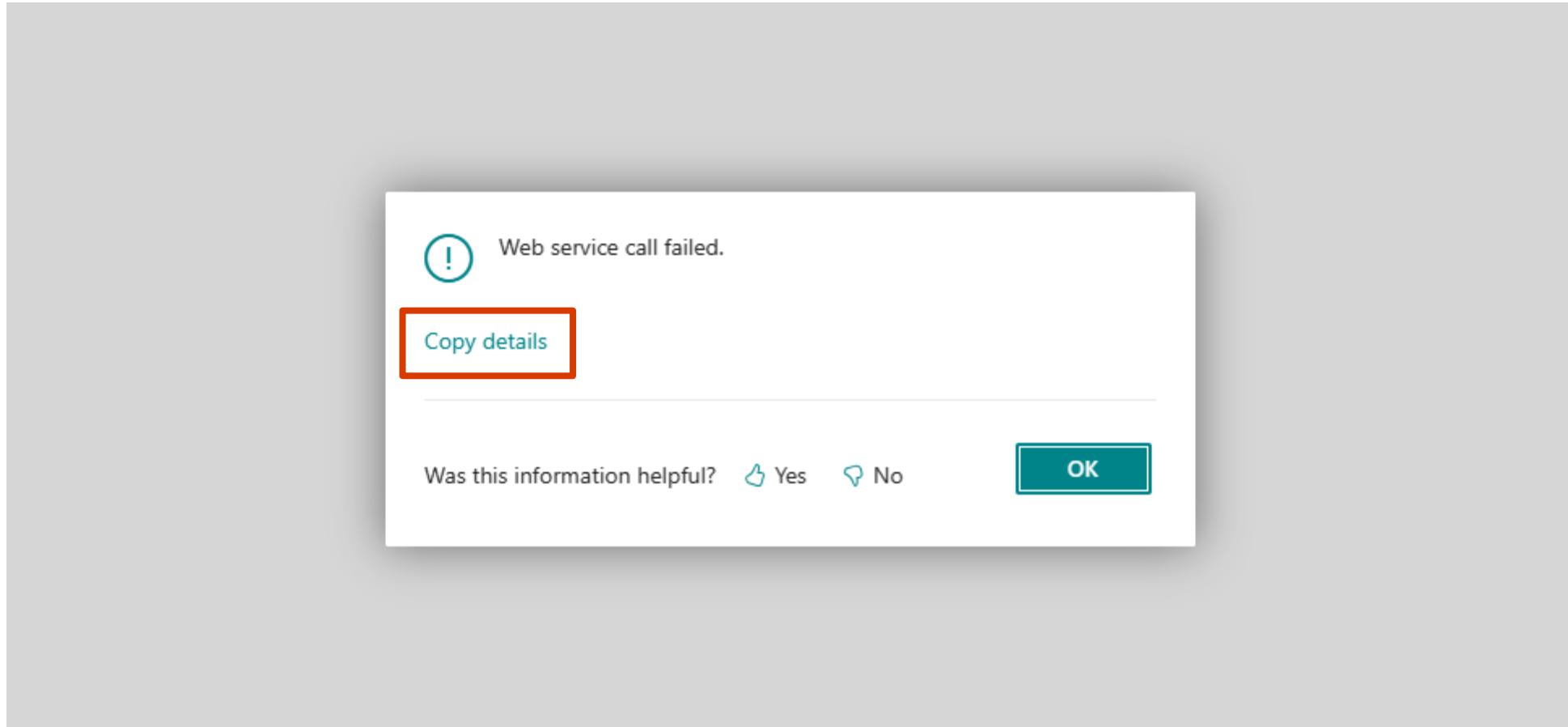
```
... setting up HttpClient ...
```

```
IsSuccessfull := HttpClient.Get(URL, Response);

if not IsSuccessfull then begin
    HttpExceptionDetails := System.GetLastErrorText();
    ErrorInfoObject.DetailedMessage := HttpExceptionDetails;
    ErrorInfoObject.Message := ServiceCallErr;
    Error(ErrorInfoObject);
end;
```

```
... more code ...
```

Developers can put the real error message in details



Example: detailed error message (HttpClient errors)

Before

If requesting support, please provide the following details to help troubleshooting:

Web service call failed.

The SSL connection could not be established, see inner exception.

Internal session ID:
d200e4f7-8f74-4ca4-aed6-94dea67fad1e

Application Insights session ID:
0394a70a-e164-4451-b1f5-93835b6c1722

Client activity id:
ac3a1952-19bf-4609-a110-c6d01c96e59a

Time stamp on error:
2023-08-23T13:18:28.4236625Z

User telemetry id:
00000000-0000-0000-000000000000

AL call stack:
APICallTest(Page 50123).
"Unhealthy Endpoint A - OnAction"(Trigger) line 40 - TestBAGCall by
Default publisher

Custom dimensions:
[]

Now (23.0)

If requesting support, please provide the following details to help troubleshooting:

Web service call failed.

The SSL connection could not be established, see inner exception.

**The remote certificate is invalid according to the validation procedure:
RemoteCertificateNameMismatch,
RemoteCertificateChainErrors**

Internal session ID:
0b9283c2-d830-445e-a9db-b7ef4e73d8ad

Application Insights session ID:
1b0bc9b1-bedb-40bb-80b9-cbc2704e9651

Client activity id:
0d2a510f-e8d6-4ac9-8eae-5b5be27221f6

Example: detailed error message

Before

If requesting support, please provide the following details to help troubleshooting:

Web service call failed.

PKCS12 (PFX) without a supplied password has exceeded maximum allowed iterations. See <https://go.microsoft.com/fwlink/?linkid=2233907> for more information.

Internal session ID:
974a987f-dcae-4abd-a99b-99ff39acb981

Application Insights session ID:
3577cf73-6798-43cc-a4ab-301cac8152d9

Client activity id:
93590c00-2931-42c3-850e-5bab2cca7a19

Time stamp on error:
2023-08-23T13:45:14.3590559Z

Now (23.0)

If requesting support, please provide the following details to help troubleshooting:

Web service call failed.

PKCS12 (PFX) without a supplied password has exceeded maximum allowed iterations. See <https://go.microsoft.com/fwlink/?linkid=2233907> for more information.

Padding is invalid and cannot be removed.

Internal session ID:
b7f38c21-a2df-4d35-a262-7e286b98c3ee

Application Insights session ID:
97d5a457-36d9-4cab-bd40-d6fcc336f2ce

Client activity id:

(Outgoing) Webservice stack more robust towards port exhaustion

Large number of outgoing calls using AL
Httpclient could starve the VM for ports.

Before: outgoing web service calls would fail.
Now: we use a pool of Httpclient objects so
that port exhaustion cannot occur

Backported and available from 21.x.

API changes coming in 2024 release wave 1

Schema version for custom APIs (changed default)

Removed or Replaced?	Why?
Replaced	The API capability in Business Central is used many places, e.g. in the APIs that come out-of-the-box. For these APIs, \$schemaversion is always set to 2.0 to get the latest features in the Business Central OData stack. For custom APIs, it is possible to get these new features by simply calling the API with \$schemaversion=2.0. Starting in version 24, the default value of \$schemaversion is set to 2.0, also for custom APIs.

Support for Delta Links with APIs (warning)

Removed or Replaced?	Why?
Replaced	Delta links are opaque, service-generated links that a web service client can use to retrieve subsequent changes to a result (see Using Delta Links with APIs). This feature was introduced in Business Central (online version only) to support the Microsoft Invoicing product, which hasn't been available for years. In version 23, the ChangeTrackingAllowed property on API pages/queries will be marked as deprecated. Starting in version 24, ChangeTrackingAllowed will be set to "false" for all APIs provided by Microsoft and support for delta links will be removed from the server. Since the delta links feature was first introduced, the webhooks functionality in Business Central has been added and improved, making it the preferred approach to change tracking. For more information about the use of webhooks, see Working with Webhooks in Dynamics 365 Business Central .

Go dos

Fix metadata issues

Use the new page in telemetry app.

Fix any naming issues etc

Write robust client code (incoming)

Handle response code 503.

Look at the TSG and harden your calling client.

Write robust AL code (outgoing)

Handle errors in AL code.

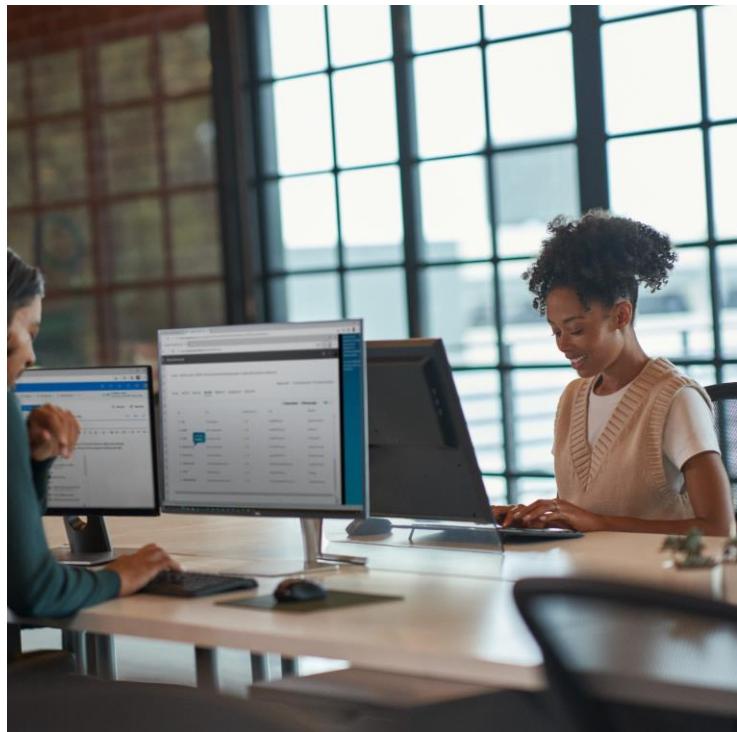
Consider retrying.

Provide relevant details as detailed information in error dialogs.

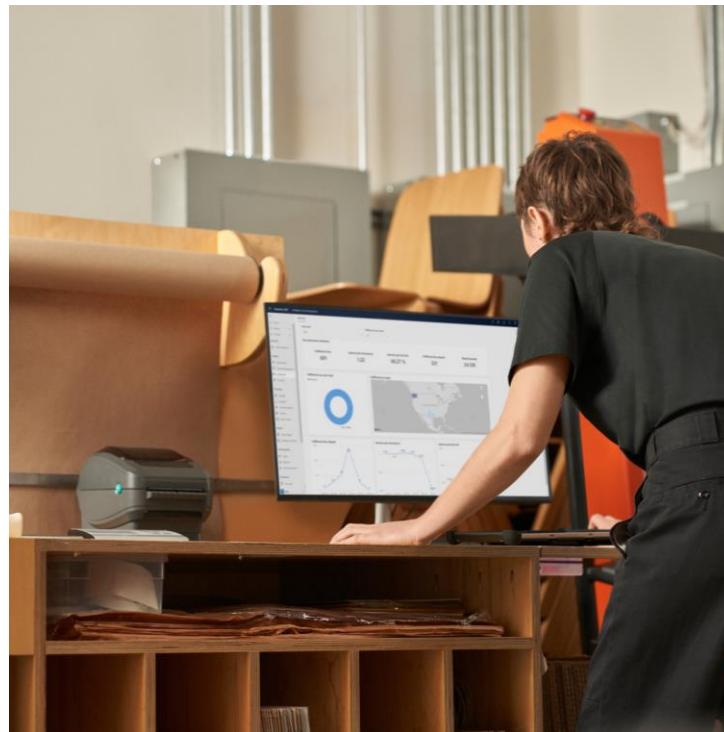
Stop using delta links

Feature is deprecated

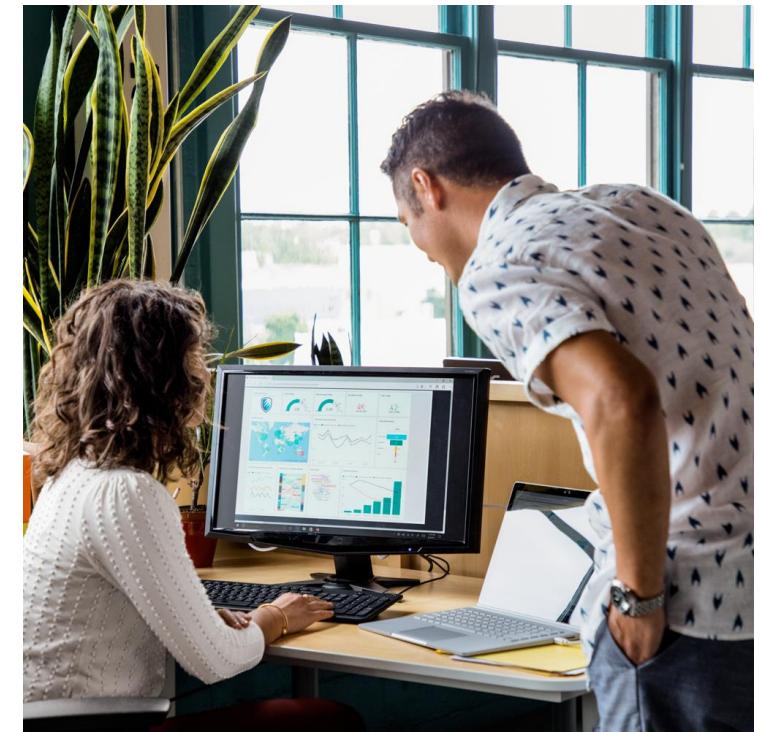
Watch other relevant launch event sessions



What's new: Developer tools

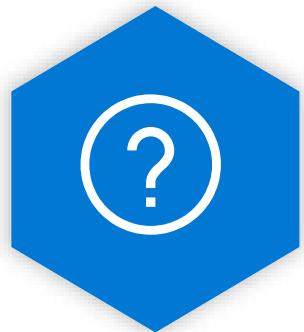


What's new: Dataverse & Dynamics 365 app integration - Part 2



What's new: Telemetry

General Business Central resources, learn more!



**Have a
question?**

aka.ms/BCYammer

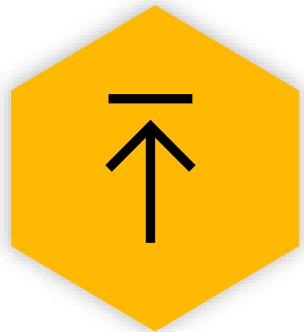


**Join the
conversation**
[twitter.com/
MSDyn365BC](http://twitter.com/MSDyn365BC)

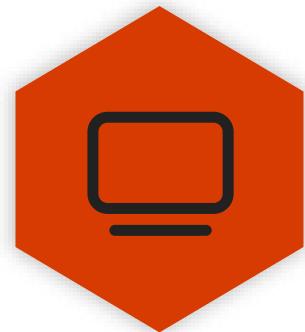


**Looking for
resources?**

aka.ms/BCAll



**Submit
your ideas**
aka.ms/BCIdeas



**Join the office
hours**

aka.ms/BCOfficeHours

Thank you

What's new in Server and Database

New reporting capabilities



Agenda

- a faster runtime
- a faster data stack
- more stable web services
- **new reporting capabilities**

New Data Analysis UI on queries

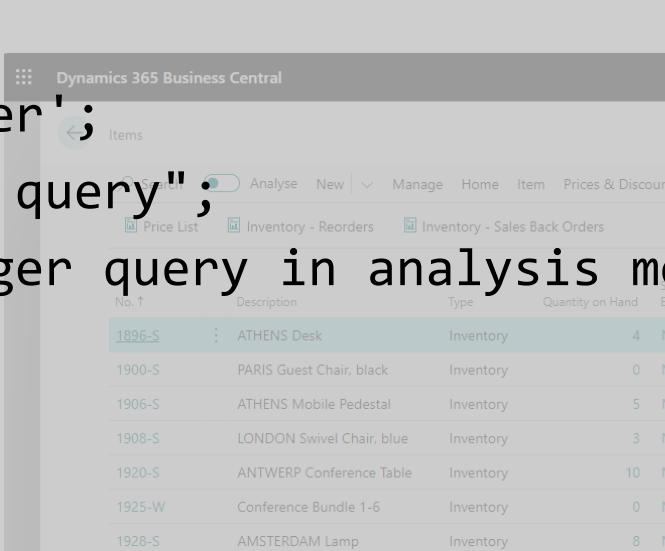
Open query URL
/?query=<query object id>

New: *Analysis* toggle button – opens new UI

New: Support for [RunObject](#)

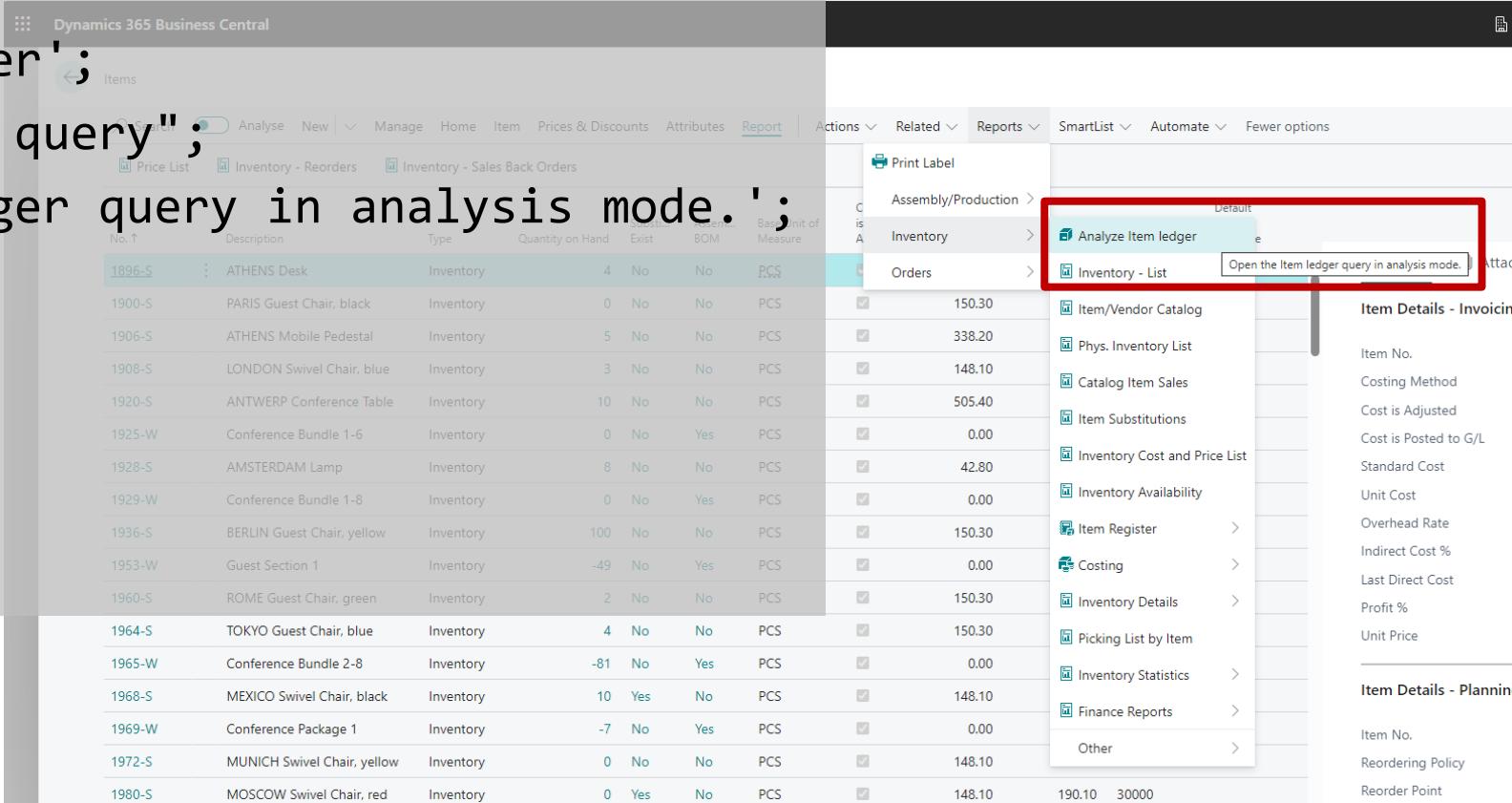
RunObject on a query

```
pageextension 50110 ItemListWithQuery extends "Item List"
{
actions {
addbefore("Inventory - List") {
action("Analyze items") {
ApplicationArea = All;
Caption = 'Analyze Item ledger';
RunObject = query "Item hero query";
Tooltip = 'Open the Item ledger query in analysis mode.';
Image = Item;
}
}
}
}
```



The screenshot shows the Dynamics 365 Business Central interface. The top navigation bar includes 'Dynamics 365 Business Central', 'Items', 'Search', 'Analyse', 'New', 'Manage', 'Home', 'Item', 'Prices & Discounts', 'Attributes', and 'Report'. Below the navigation is a toolbar with icons for 'Price List', 'Inventory - Reorders', and 'Inventory - Sales Back Orders'. The main area displays a table of item records:

No.	Description	Type	Quantity on Hand	Available Exist	Allocated BOM	Base Unit of Measure
1896-S	ATHENS Desk	Inventory	4	No	No	PCS
1900-S	PARIS Guest Chair, black	Inventory	0	No	No	PCS
1906-S	ATHENS Mobile Pedestal	Inventory	5	No	No	PCS
1908-S	LONDON Swivel Chair, blue	Inventory	3	No	No	PCS
1920-S	ANTWERP Conference Table	Inventory	10	No	No	PCS
1925-W	Conference Bundle 1-6	Inventory	0	No	Yes	PCS
1928-S	AMSTERDAM Lamp	Inventory	8	No	No	PCS
1929-W	Conference Bundle 1-8	Inventory	0	No	Yes	PCS
1936-S	BERLIN Guest Chair, yellow	Inventory	100	No	No	PCS
1953-W	Guest Section 1	Inventory	-49	No	Yes	PCS
1960-S	ROME Guest Chair, green	Inventory	2	No	No	PCS



Multi-worksheet data in Excel layouts

Excel layouts feature did not handle parent-child/overview-detail types of data very well.

With this change, such report datasets can be designed in a way that enables a very rich way to model things further in PowerPivot.

How it works

ExcelLayoutMultipleDataSheets = false

A	B	C	D	E	F	G	H	I	J	K	L	
1	IntegerColumn1	TextColumn1	Child1IntegerColumn	Child1TextColumn	IntegerColumn2	TextColumn2	Child2IntegerColumn	Child2TextColumn	IntegerColumn3	TextColumn3	Child3IntegerColumn	Child3TextColumn
2	0	100'100	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	0	101'101	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	0	102'102	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	1	100'100	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	1	101'101	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	1	102'102	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	2	100'100	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	2	101'101	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	2	102'102	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
11	NULL	NULL	NULL	NULL	0'0	100'100	NULL	NULL	NULL	NULL	NULL	NULL
12	NULL	NULL	NULL	NULL	0'0	101'101	NULL	NULL	NULL	NULL	NULL	NULL
13	NULL	NULL	NULL	NULL	0'0	102'102	NULL	NULL	NULL	NULL	NULL	NULL
14	NULL	NULL	NULL	NULL	1'1	100'100	NULL	NULL	NULL	NULL	NULL	NULL
15	NULL	NULL	NULL	NULL	1'1	101'101	NULL	NULL	NULL	NULL	NULL	NULL
16	NULL	NULL	NULL	NULL	1'1	102'102	NULL	NULL	NULL	NULL	NULL	NULL
17	NULL	NULL	NULL	NULL	2'2	100'100	NULL	NULL	NULL	NULL	NULL	NULL
18	NULL	NULL	NULL	NULL	2'2	101'101	NULL	NULL	NULL	NULL	NULL	NULL
19	NULL	NULL	NULL	NULL	2'2	102'102	NULL	NULL	NULL	NULL	NULL	NULL
20	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0'0	100'100	NULL	NULL	NULL
21	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0'0	101'101	NULL	NULL	NULL
22	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0'0	102'102	NULL	NULL	NULL
23	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1'1	100'100	NULL	NULL	NULL
24	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1'1	101'101	NULL	NULL	NULL
25	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1'1	102'102	NULL	NULL	NULL
26	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2'2	100'100	NULL	NULL	NULL
27	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2'2	101'101	NULL	NULL	NULL
28	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2'2	102'102	NULL	NULL	NULL

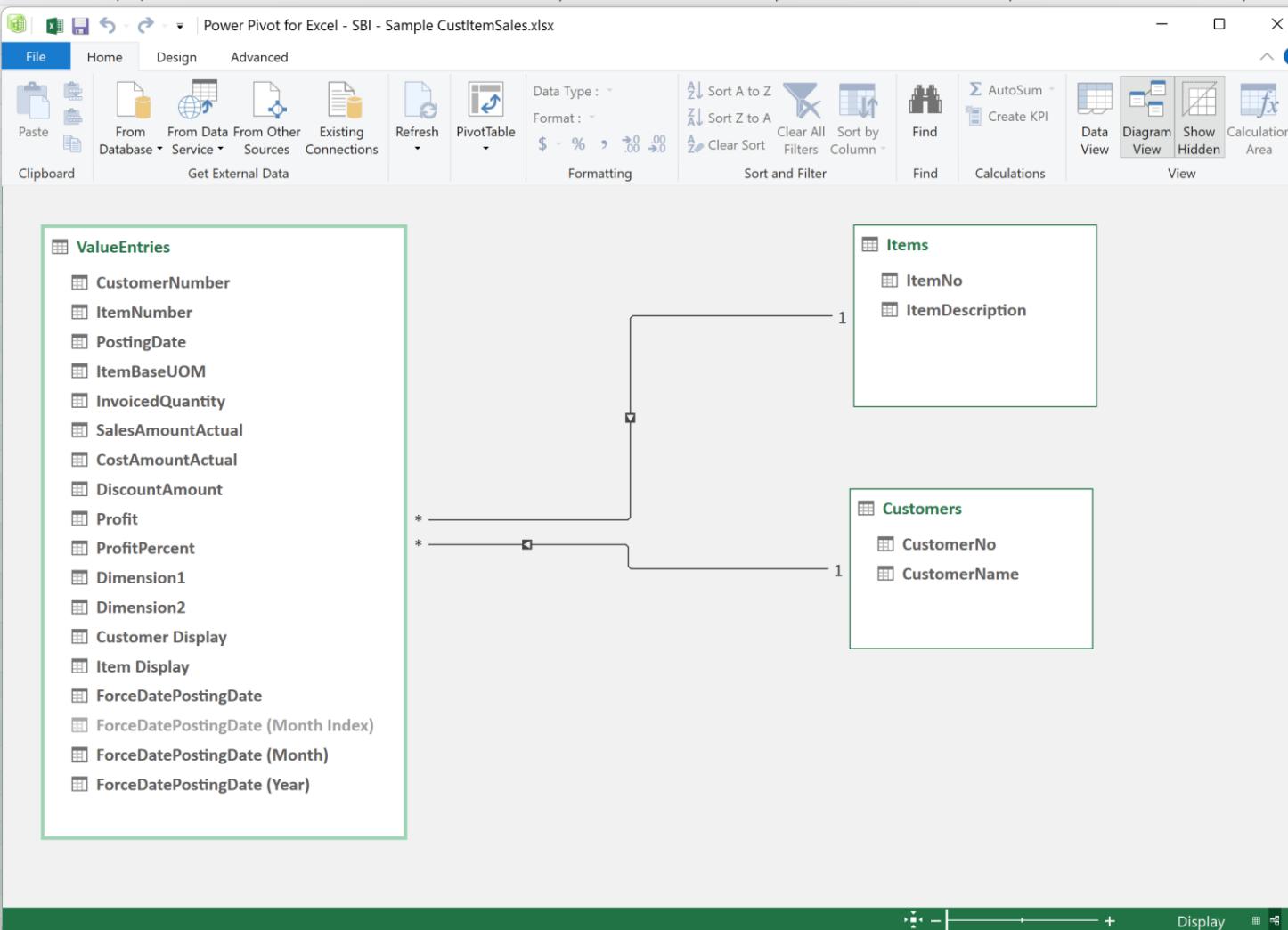
ExcelLayoutMultipleDataSheets = true

	A	B	C	D	E	F	G
1	IntegerColumn	TextColumn	ChildIntegerColumn	ChildTextColumn			
2	0 0		100	100			
3	0 0		101	101			
4	0 0		102	102			
5	1 1		100	100			
6	1 1		101	101			
7	1 1		102	102			
8	2 2		100	100			
9	2 2		101	101			
10	2 2		102	102			
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							



B4 : fx 1896-S

	A	B	C	D	E	F	G	H	I	J
1	CustomerNumber	ItemNumber	PostingDate	ItemBaseUOM	InvoicedQuantity	SalesAmountActual	CostAmountActual	DiscountAmount	Profit	ProfitPercent
2	10000	1896-S		23/01/2022 PCS	5,	3.247,00	2.533,00	0,00	714,00	C
3	10000	1896-S							999,60	C
4	10000	1896-S							1.142,40	C
5	10000	1896-S							1.428,00	C
6	10000	1896-S							1.142,40	C
7	10000	1896-S							1.142,40	C
8	10000	1896-S							856,80	C
9	10000	1896-S							856,80	C
10	10000	1896-S							714,00	C
11	10000	1896-S							714,00	C
12	10000	1896-S							571,20	C
13	10000	1896-S							428,40	C
14	10000	1896-S							856,80	C
15	10000	1896-S							1.142,40	C
16	10000	1896-S							1.428,00	C
17	10000	1953-W							163,20	C
18	10000	1953-W							1.876,80	C
19	10000	1960-S							110,40	C
20	10000	1960-S							110,40	C
21	10000	1960-S							138,00	C
22	10000	1960-S							138,00	C
23	10000	1960-S							110,40	C
24	10000	1960-S							110,40	C
25	10000	1960-S							110,40	C
26	10000	1960-S							110,40	C
27	10000	1960-S							82,80	C
28	10000	1960-S							82,80	C
29	10000	1960-S							82,80	C
30	10000	1960-S							55,20	C
31	10000	1960-S							110,40	C
32	10000	1960-S							110,40	C
33	10000	1960-S							138,00	C
34	10000	1965-W							910,20	C
35	10000	1965-W							1.668,70	C
36	10000	1965-W							151,70	C
37	10000	1969-W							1.109,00	C
38	10000	1972-S							54,40	C
39	10000	1972-S							54,40	C
40	10000	1972-S		20/01/2022 PCS	3,	369,90	288,30	0,00	81,60	C
41	10000	1972-S		21/04/2022 PCS	2,	360,00	260,20	2,20	81,60	C
42	10000	1972-S		22/05/2022 PCS	2,	360,00	260,20	2,20	81,60	C



Multi-worksheet data in Excel layouts

```
report 50101 MyReport {  
    ExcelLayoutMultipleDataSheets = true;  
  
    // more code here...  
}
```

Teaching tips and help links on queries and reports

Help users understand what reports and queries can be used for.

Give them help links for the specific report or query.

Get started

Hi, meet Business Central!

You're all set to try out our demo. Go explore on your own, or take a tour.

[Show demo tours](#)

Activities ▾

Sales This Month

\$1,906

[See more](#)

Ongoing Sales

Sales Quotes

2

Sales Orders

9

from Stock

0

Invoices

7

Next Week

8

Invoices

3

Invoices...

0

Awesome report (caption)

Awesome report

This is an awesome report. Use it to be awesome

[Show Help](#)[Got it](#)[+ Filter...](#)

Filter totals by:

[+ Filter...](#)

Filter: Item Category

[+ Filter...](#)[Advanced >](#)

AboutTitle
AboutText

[Send to...](#)[Print](#)[Preview & Close](#)[Cancel](#)

Help

Search here...

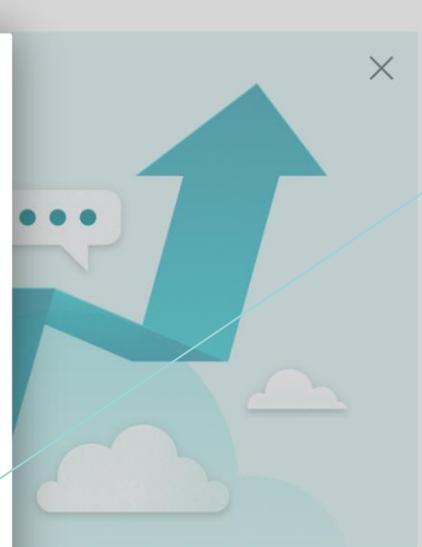
Awesome report

This is an awesome report. Use it to be awesome

Take a tour

[Go to Microsoft Learn](#)

Other resources

[Help & Support](#)[Keyboard Shortcuts](#)[Community](#)

Some features connect to other Microsoft services

ContextSensitiveHelpPage

Dynamics 365 Business Central

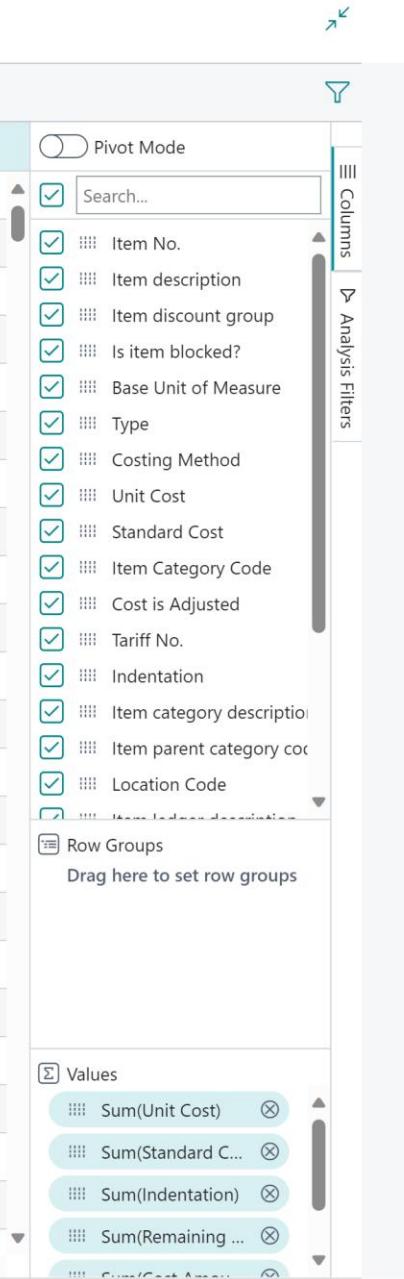
Analyze your inventory

Analyze your inventory (slice and dice your data)

Use this data cube to analyze your inventory. You can group and/or the data on any dimension you like.

Show Help

Got it



Help

 Search here...

Analyze your inventory (slice and dice your data)

Use this data cube to analyze your inventory. You can group and/or filter the data on any dimension you like.

[Take a tour](#)

 Go to Microsoft Learn

Other resources

 Help & Support

Keyboard Shortcuts

 Community

Filter by title

Development and administration

New and planned

> Get started

> Administration

Development

Development overview

> Get started

> Development environment

AL language

A tour of the AL language

FAQ for AL developers

> Program building blocks

> Language elements

> Error handling

> Formatting values

> Tables and data

> Pages and user interfaces

Developing reports

Overview

Report design overview

Report object

... / Developer and Admin / Developer / Development / AL language /

+ ⚙️ ⋮

Additional resources

 Training

Module

[Use the report design process in Dynamics 365 Business Central - Training](#)

Do you want to learn about the report design process for Microsoft Dynamics 365 Business Central? If so, this module will explain the different...

Reports Overview

Article • 09/04/2023 • 6 contributors

 Feedback

In this article

[Creating reports](#)[Getting started](#)[See Also](#)

You can use reports to print or display information from a Business Central database. Use reports to structure and summarize information to print documents, such as invoices. For example, create a report that lists all customers and orders that have been added by each customer. Also, create a report that is automatically filled with the relevant information for an invoice.

Reports can also be used to process data without printing or displaying content. For example, use a report to automate updating all prices in an item list. It can be easier to create a report to process data instead of a codeunit to do the same processing because you can use:

- Request page functionality to select options and filters for data items, which are available in a report but are difficult to add to a codeunit. For more information, see [Using request pages with reports](#).
- Report data items instead of writing code to open tables and retrieve records.
- Data modeling, which is available when you design reports.

 Documentation[Number Sequences - Business Central](#)

This topic describes how to create number sequences in AL code in Dynamics 365 Business Central.

[Simple Role Center Example - Business Central](#)

Provides AL code for a simple role center

[Defining Multiple Report Layouts - Business Central](#)

Describes how to define multiple layouts for one report in Business Central using AL.

[Show 5 more](#)

aka.ms/bcexcelsamples

The screenshot shows a GitHub repository for 'BCTech / samples / Excel /'. The repository contains several files and folders:

- guidelines**: A folder highlighted with a red box.
- layouts (ready to use)**: A folder highlighted with a red box.
- Excel-layouts.pptx**: A file highlighted with a red box.
- FAQ.md**: A file highlighted with a red box.
- VIDEOS.md**: A file highlighted with a red box.
- README.md**: A file at the bottom of the list.

A callout box labeled "Tips and tricks for datasets and layouts" points to the "guidelines" folder. Another callout box labeled "Sample layouts for existing reports" points to the "layouts (ready to use)" folder. A third callout box labeled "FAQ + list of community blog posts" points to the "FAQ.md" file. A fourth callout box labeled "List of community videos" points to the "VIDEOS.md" file. A fifth callout box labeled "Want to do presentations for events or conferences?" points to the "Excel-layouts.pptx" file.

What is Excel layouts for Business Central?

Business Central 2022 release wave 1 (version 20) introduced the feature *Excel layouts* as a way to layout and run reports using Excel.

Go dos

Understand analytics requirements

Stop developing reports.

Start supporting analytics needs.

Learn how to Power BI

Business Intelligence (BI) is closely related to ERP.

Power BI is here to stay and is the suggested platform for BI on Business Central data.

Learn how to write AL queries

AL queries are not only for performance tuning anymore.

Queries are a powerful tool to implement data mashups for self-service data analysis.

“Build me a report” might just mean “Build me a query.”

Learn how to design AL for Excel layouts

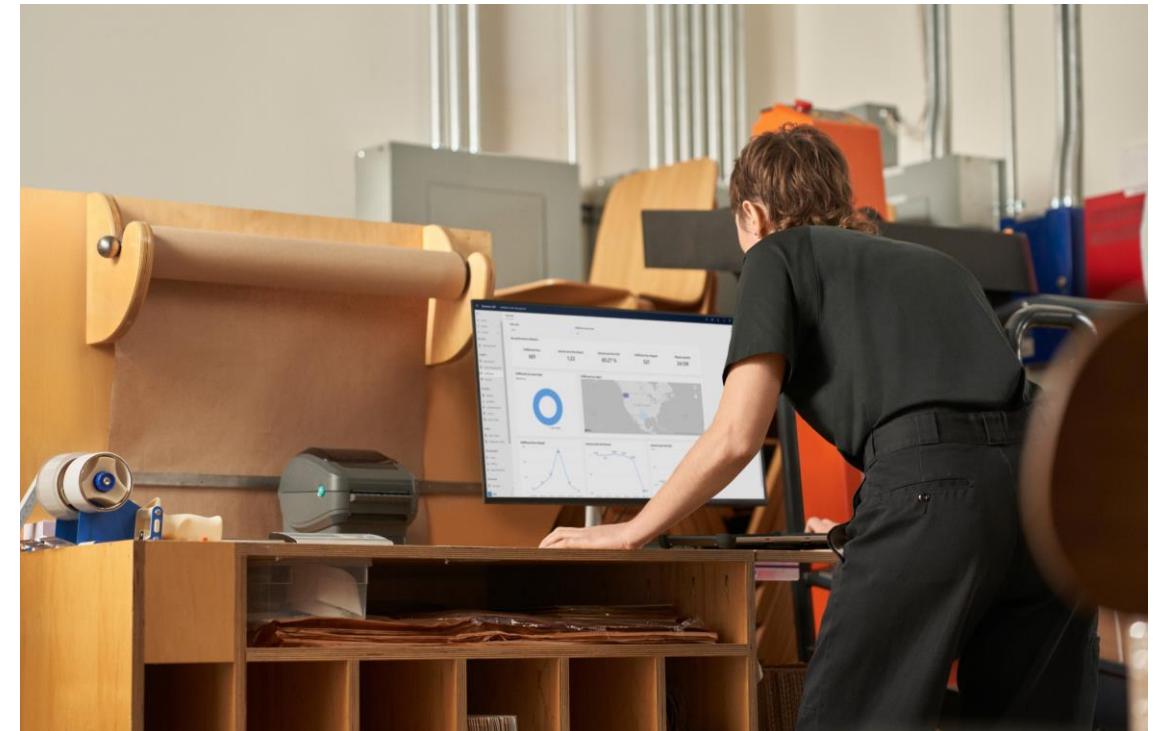
Your job as an AL report developer is changing. In many cases, a good report dataset is enough to unblock the customer.

“Build me a report” might just mean “Build me a report dataset.”

Watch other relevant launch event sessions

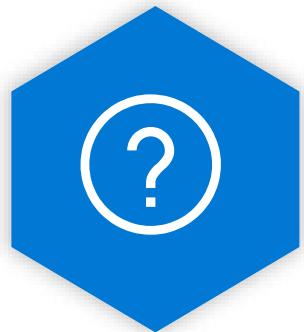


What's new: Power BI and reporting (for developers)



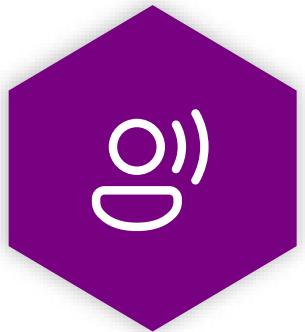
Introducing: Analyze data on lists and queries

General Business Central resources, learn more!



**Have a
question?**

aka.ms/BCYammer

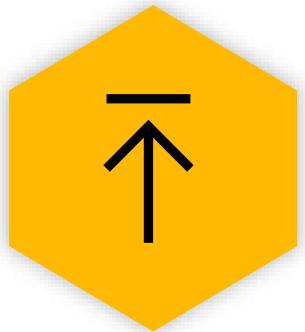


**Join the
conversation**
[twitter.com/
MSDyn365BC](http://twitter.com/MSDyn365BC)

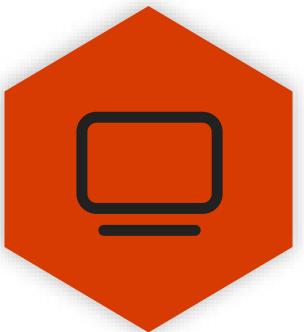


**Looking for
resources?**

aka.ms/BCAll



**Submit
your ideas**
aka.ms/BCIdeas



**Join the office
hours**

aka.ms/BCOfficeHours

Thank you