Microsoft Dynamics 365

What's new in
Server and Database

Agenda

New server features in

- Data stack

- Web service stack

- AL runtime

- Reporting

- Security

Data stack

Faster upgrades with DataTransfer

New AL type DataTransfer

Allows for

- bulk transfer of fields between two given tables
- copies all rows (matching a filter) from one table to another

How-to use DataTransfer

Setup from and to tables

- `SetTables(<fromTable>, <totable>);`

Select fields / values to copy

- `AddFieldValue(<fromFieldNo>, <toFieldNo>);`
- `AddConstantValue(<value>, <toFieldNo>);`

Define relationship between tables

- `AddJoinCondition(<fromFieldNo>, <toFieldNo>);`

Filter on source

- `AddSourceFilter(<fromFieldNo>, <filterString>);`

Invoke operation

- `CopyFields();`
- `CopyRows();`

DataTransfer: move a field

Before

```
local procedure MoveField()
var
  from: Record FromTable;
  to: Record ToTable;
begin
  if from.Find() then
  repeat
    to.Get(from.RecordId);
    to.IntField := from.IntField;
    to.SmallCodeField := from.SmallCodeField;
    to.Modify();
  until (from.Next() = 0)
end;
```

With DataTransfer

```
local procedure MoveField()
var
  dt : DataTransfer;
  from: Record FromTable;
  to : Record ToTable;
begin
  dt.SetTables(Database::FromTable, Database::ToTable);

  dt.AddFieldValue(from.FieldNo("SmallCodeField",to.FieldNo("SmallCodeField"));
  dt.AddFieldValue(from.FieldNo("IntField", to.FieldNo("IntField"));

  dt.AddJoinCondition(from.FieldNo("id"), to.FieldNo("id"));

  dt.CopyFields();
end;
```

DataTransfer: copy a row

Before

```
local procedure CopyRows()
var
  from: Record FromTable;
  to: Record ToTable;
begin
  if from.Find() then
  repeat
    to.SmallCodeField := from.SmallCodeField;
    to.IntField := from.IntField;
    to.id  := from.id;
    to.Insert();
  until (from.Next() = 0)
end;
```

With DataTransfer

```
local procedure CopyRows()
var
  dt : DataTransfer;
  to : Record ToTable;
begin
  dt.SetTables(Database::FromTable, Database::ToTable);

  dt.AddFieldValue(2, to.FieldNo("SmallCodeField"));
  dt.AddFieldValue(3, to.FieldNo("IntField"));
  dt.AddFieldValue(1, to.FieldNo("id"));

  dt.CopyRows();
end;
```

DataTransfer use cases

Common upgrade scenarios:

Move a field from one table to another (obsolete a field).

Move data from one table to another (deprecate a table).

DataTransfer limitations

Only allowed in upgrade code
(Events are not raised)

Disabled for the following table types
Non-SQL tables
System and virtual tables
Audited tables (destination)
Obsoleted (destination)

Disabled for the following field types
System and Timestamp fields (destination)
Media and MediaSet
Calculated fields

New system field System Rowversion

Rowversion field that are on all tables now exposed as system field SystemRowversion.

Field name (SQL): timestamp

Field name (AL): SystemRowVersion

Read-only.

New functions

```
// Lowest row version part of some active transaction
// (anything lower has been committed)
MinimumActiveRowVersion()


// Last used RowVersion from the database.
LastUsedRowVersion
```

Rowversion use cases

Create fully covering indexes.

Read changes since last recorded rowversion for faster data integration.

Show SQL Locks in debugger

Non-blocking number series (in demo data)

Adjusted demo/setup data to allow gaps by default (where allowed).

Changed default settings for non-blocking number series

No. Series | Work Date: 1/25/2024 — Saved

Search + New Edit List Delete Navigate | Actions Related Fewer options

Lines Relationships

Code ↑	Description	Starting No.	Ending No.	Last Date Used	Last No. Used	Defa... Nos.	Man... Nos.	Date Order	Allow Gaps in Nos.
IJNL-REC	Recurring Item Journal	T03001	T04000	_	_	☑	☑	☐	☐
IJNL-REVAL	Revaluation Journal	T04001	T05000	12/31/2023	T04001	☑	☑	☐	☐
I-MOVE+	Registered Invt. Movement	RIM000001	RIM999999	_	RIM000001	☑	☑	☐	☑
I-MOVEMENT	Inventory Movement	IM000001	IM999999	_	IM000001	☑	☑	☐	☑
INT-MOVE	Internal Movement	RINTM000001	RINTM999999	_	RINTM000001	☑	☑	☐	☑
I-PICK	Inventory Pick	IPI000001	IPI999999	_	IPI000001	☑	☑	☐	☑
I-PICK+	Posted Invt. Pick	PPI000001	PPI999999	_	PPI000001	☑	☑	☐	☑
I-PUT	Inventory Put-away	IPU000001	IPU999999	_	IPU000001	☑	☑	☐	☑
I-PUT+	Posted Invt. Put-away	PPU000001	PPU999999	_	PPU000001	☑	☑	☐	☑
I-RCPT	Inventory Receipt	IR000001	IR999999	_	IR000001	☑	☑	☐	☑
I-RCPT+	Posted Inventory Receipt	IR000001	IR999999	_	IR000001	☑	☑	☐	☑
I-SHPT	Inventory Shipment	IS000001	IS999999	_	IS000001	☑	☑	☐	☑
I-SHPT+	Posted Inventory Shipment	IS000001	IS999999	_	IS000001	☑	☑	☐	☑
ITEM1	Partially Manufactured	70000	70099	_	70060	☑	☑	☐	☐
ITEM2	Paint	70100	70199	_	70104	☐	☑	☐	☑
ITEM3	Loose Hardware	70200	70299	_	70201	☐	☑	☐	☑
ITEM4	Finished	1896-S	2996-S	_	2000	☐	☑	☐	☑
ITEM5	Assembly BOM	1924-W	2096-W	_	1992	☐	☑	☐	☑
JJNL-GEN	Job Journal	J00001	J01000	_	J00001	☑	☑	☐	☑
JJNL-REC	Recurring Job Journal	J01001	J02000	_	J01001	☑	☑	☐	☑
JOB	JOB	J00010	J99990	_	J00010	☑	☑	☐	☑
JOB-WIP	Job-WIP	WIP0000001	WIP9999999	_	WIP0000001	☑	☑	☐	☑

New version of SQL Server Express

Installer now deploys SQL Server 2019 Express.

Data stack performance improvements

Partial records now skip obsoleted fields outside upgrade code.
Partial records load fewer fields in calcformulas.

Bulk ModifyAll involving fields from table extensions is faster.

Faster transaction ending in multi-tenancy mode.

Data search is faster.

Web services stack

Other web services stack improvements

Avoiding JIT Load in pages referenced through $expand
   Large reduction in SQL calls
   (already shipped with update 2 for Business Central 2022 release wave 1).

New server setting - ODataMaxBodySize
   Default is 350MB (also for SaaS)
   Automation API now allows larger files.

Better filter handling for Edit-In-Excel.

Added HTTP Retry-After to 429 responses in Odata (when throttled)

AL runtime

Discover orphan media in the database

AL developers can discover orphan media in the database.

New methods

- Media.FindOrphans
- MediaSet.FindOrphans.

Returns list of Media/MediaSet GUIDs not referenced anywhere.

Coming soon: NST on .NET Standard 2.0/core

We are done moving the server to .NET standard. Releasing in 2023w1.

Start conversion of your (on-premises) .NET add-ins now...

.NET addins running .NET framework will not work in 2023 release wave 1.

Move to services (Azure functions and friends) where possible.

New system application module: Azure Function Integration

Integrate to Azure Functions from AL using the new system module Azure Function Integration.

We used it for two modules that we could not migrate to .NET standard.

Security

Permission system

<table>
<tr><td>

2022 release wave 1

Composing permission sets
Excluding permissions/permission sets
InherentPermissions
Isolated Events

</td><td>

New in 2022 release wave 2

UI for composing permissions
InherentPermissions available for apps
Exclude permission sets in metadata
InherentEntitlements for objects

</td></tr>
</table>

Compose permission sets in the UI

PS Name (Type)

Included permissions for the currently selected PS

Permission Set | Work Date: 1/25/2024 — ✓ Saved

TECHDAYS BASIC (Tenant)

Include/Exclude permissions

🔲 View all permissions

General

Permission Set	TECHDAYS BASIC	Name	Techdays Basic

Permissions | Manage

Type	Object Type ↑	Object ID ↑	Object Name	Read Permission	Insert Permission	Modify Permission	Delete Permission	Execute Permission	Security Filter
Include	Table Data	472	Job Queue Entry	Yes	Yes	Yes	Yes		
→ Include	⋮ Table Data	474	Job Queue Log Entry	Yes	Yes	Yes	Yes	-	

Included permissions ∨

Object Type ↑		Object ID ↑	Object Name
Table Data	⋮	3	Payment Terms
Table Data		4	Currency
Table Data		5	Finance Charge Terms
Table Data		6	Customer Price Group
Table Data		7	Standard Text

Permission Sets ∨

Type ↑	Permission Set ↑	Scope
Include	D365 READ	System
Include	EMAIL - ADMIN	System
Include	FEATURE MGT. - ADMIN	System
→ Include		System
		System

Result ∨

| Permission Set	Scope	Inclusion Status
→ D365 READ	⋮ System	**Full**
EMAIL - ADMIN	System	**Full**
FEATURE MGT. - ADMIN	System	**Full**

Include permission sets

Result of included permission sets and hierarchy

Defining your own permissions hierarchy
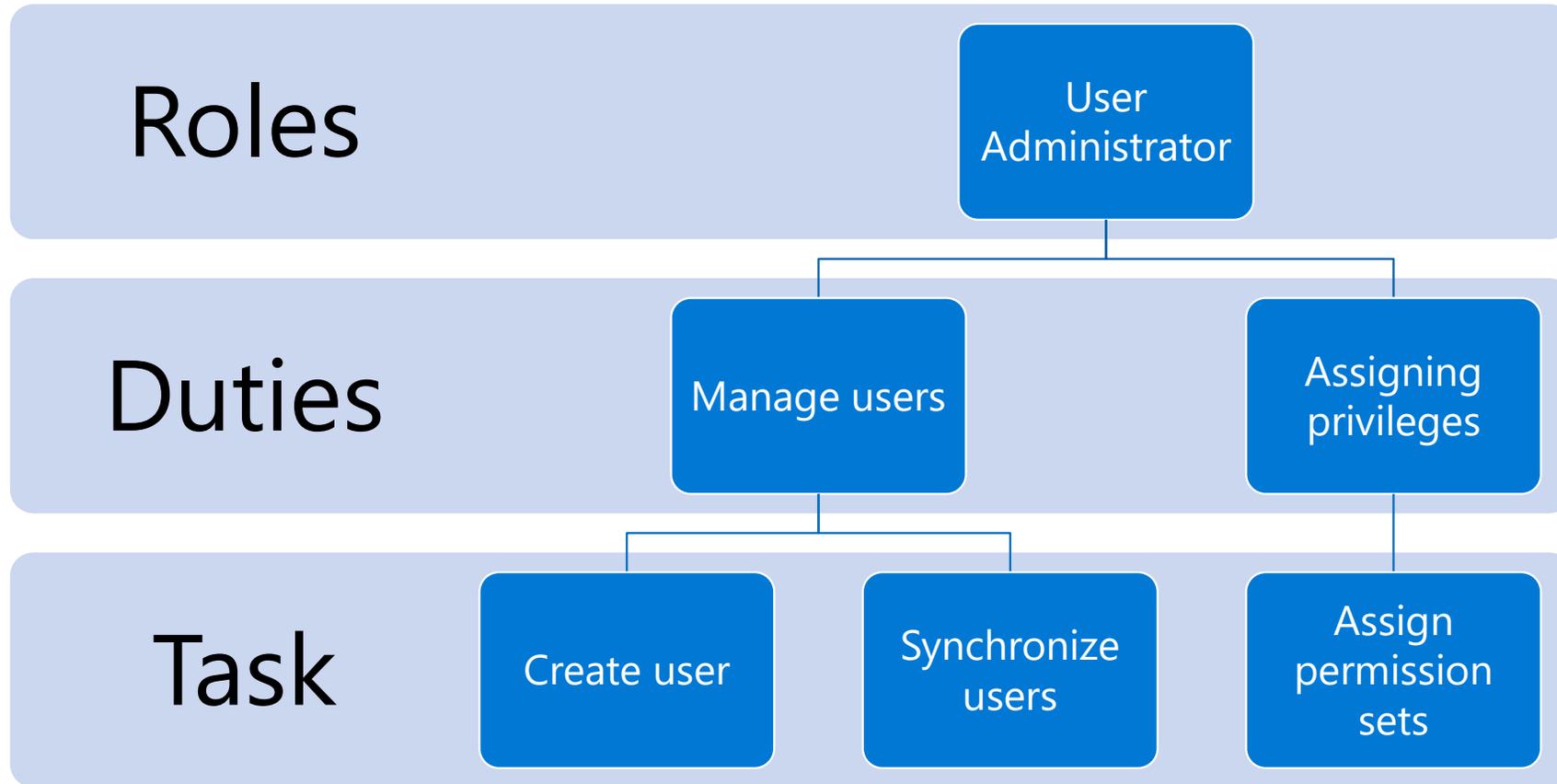
How inclusion and exclusion of permission sets work

Warning – Math notation
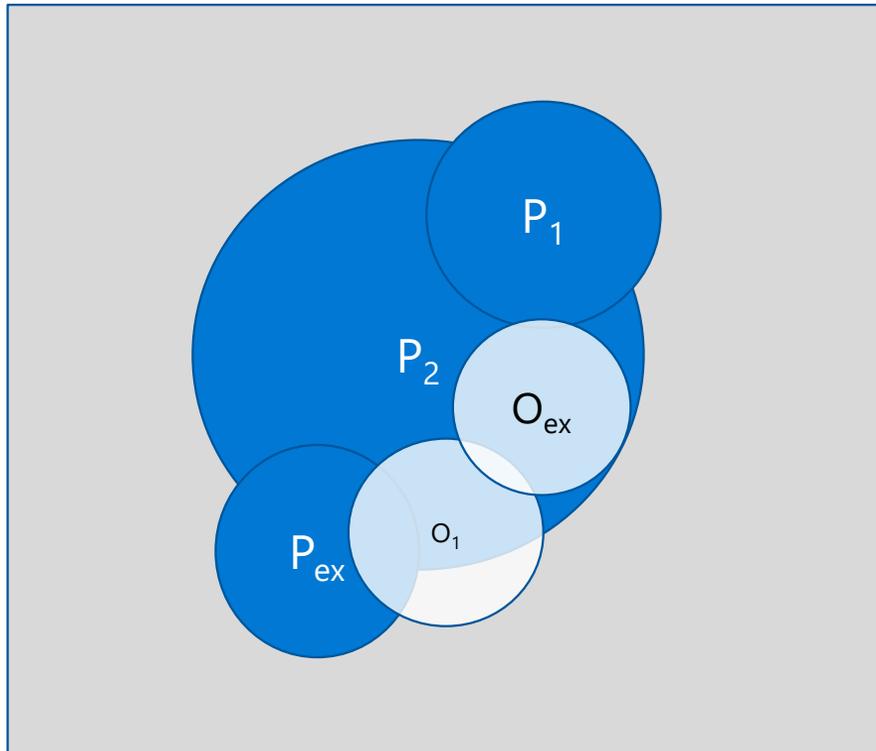
Permission set P_{top}
   Includes permission sets P_1 and P_2
   Includes explicit permissions P_{ex}
   Excludes permission set O_1
   Excludes explicit permissions O_{ex}

$$P_{top} = (P_1 \cup P_2 \cup Pex) \mathbin{/} (O_1 \cup Oex)$$

$\cup$ is set union
$/$ is set complement

Dark blue area are your permissions.

Exclude permission sets in the UI

Exclude by updating the type

The PS results update with their inclusion status

Exclude permission sets in metadata

Exclude permission sets in permission set definitions.

New property ExcludedPermissionSets

Excluding permission set in metadata

In extensions you can only exclude permission *sets.*

```
permissionset 132218 "Test Tables - Restricted"
{
    Assignable = true;

    IncludedPermissionSets = "System Application Test Tables",
                             "Local Test Tables";

    ExcludedPermissionSets = "Restricted Local Test Tables"

    Permissions = tabledata "All-Keys Type" = RIMD,
                  tabledata "Amount Auto Format Test Table" = RIMD,
```

Example: Create an accountant role who can only read from General Ledger Accounts

Exclude read permissions from the "Edit" permission set to get only read permissions.

Exclude all edit permissions provided by the accountant permission set.

```
permissionset 100299 AccountantReadOnly
{
    IncludedPermissionSets = "D365 ACCOUNTANTS";
    ExcludedPermissionSets = GeneralLedgerEditOnly;



}

1 reference
permissionset 100300 GeneralLedgerEditOnly
{
    IncludedPermissionSets = "General Ledger Accounts - Edit";
    ExcludedPermissionSets = "General Ledger Accounts - View";
}
```

Security filters

Composing permission sets supports security filters.

Excluding permission sets currently *does not* apply security filters.

Old-time favorite behavior changed:

Allow sorting on flow fields when security filters are unrelated.

Inherent Permissions available for Apps

AL developers can elevate permissions for specific code paths.

New method annotation

- InherentPermissions

Can only elevate permissions to your own objects.

InherentPermissions available for Apps

Method annotation

```
[InherentPermissions(PermissionObjectType:Table, Database:MyTable, 'x', InherentPermissionScope:Entitlements)]

[InherentPermissions(PermissionObjectType:Table, Database:MyTable, 'x', InherentPermissionScope:Permissions)]

[InherentPermissions(PermissionObjectType:Table, Database:MyTable, 'x', InherentPermissionScope:Both)]



// Default value : InherentPermissionScope:Both
```

When and how to use Inherent permissions

Small dedicated procedures performing a task

- Avoid adding InherentPermissions at top level
- Ideally avoid calling other methods or raising events

Critical code paths

- Code that should run, regardless of the user

'System tasks' that do not expose data to users

- Initialization of default values
- Derived information like: Counts of records, is feature enabled

No need for Stan to restrict access

- Stan cannot remove privileges to run this code

Inherent Entitlements available for Apps

AL developers can elevate entitlements for specific objects.

New property InherentEntitlements

Can only elevate entitlements to your own objects

Reporting

Layout picker on the request page

End users can change the layout for the report on the request page

Possible to navigate to report layouts page from request page

Layout picker on the request page

Before

Now (2022 release wave 2)

… opens the Report Layouts page

Report ID ↑		Layout Name	Description	Extension	Type ↑
111		./CustomerTop10List.rdlc	./CustomerTop10List.rdlc	Base Application by Microsoft	RDLC
111		Customer Top-10 (Printing)	Use for printing.		RDLC
→ 111	⋮	Customer Top-10 (Visualization)	Use this report layout for mon…		Excel

Customer - Top 10 List

Report Layouts

Shows all layouts from the current report.

OK   Cancel

Email Status

Send to…   Print   Preview   Cancel

After choosing a new layout...

Customer - Top 10 List

Printer	(Handled by the browser)
Report Layout	Customer Top-10 (Visualization)
Use default values from	Last used options and filters

Options

Show	Sales (LCY)
Number of Customers	10
Chart Type	Bar chart

...ner

Request page exit controls are dynamically updated depending on the layout format

× Customer Posting Group

Send to...   Download   Cancel

Bonus: administer layouts on the Report Layouts page

Reduced number of steps for basic report operations

Operation	Before	2022 release wave 2
Adding a layout	- Exit the request Page - Search and open the Report Layouts page - Filter the layouts for your desired report. - Use action 'New Layout'.	- Open the layout selector - Use action 'New Layout'
Deleting a layout	- Exit the request Page - Open the Report Layouts page - Filter the layouts for your desired report. - Select the layouts and delete.	- Open the layout selector - Select layouts and delete
Changing the default layout	- Exit the request page. - Open Report Layout Selection page. - Filter for your report. - Select desired layout format. - Select from available layouts.	- Open the layout selector - Select desired layout and use 'Set Default' action.

Miscellaneous reporting improvements

PrintOnlyifDetail is now available in SaaS

New Word reporting trigger OnIntermediateDocumentReady

Better filenames in OneDrive for reports sent to inbox

Go dos

Update your upgrade code

Uptake the new DataTransfer feature to make your upgrade code faster (and more stable).

Move .NET addins to .NET Standard

.NET addins using .NET framework will no longer work in 2023 release wave 1.

Reduce database locks

Use the debugger to display database locks. Optimize your code to reduce locking issues.

Redesign your app permissions

Stop copying our permission sets for excluding permissions (the old way).

Use the new permission system building blocks to make your app permissions easier to understand and maintain.

Thank you