

PlanetTogether Integration Specification for Microsoft Dynamics 365 for Finance and Operations

Last updated: October 2018

Overview of the the integration between PlanetTogether and D365:

- Bi-directional flow of data between PlanetTogether and D365 (as specified in the Object Mapping section below)
- Azure SQL Server based using the Bring Your Own Database (BYOD) method, thus supporting “full push” or “incremental push” of data
- Highly configurable to adjust to various D365 configuration and usages

D365 to PlanetTogether Process

To import data into PlanetTogether from D365, select entities are exported to an Azure SQL database. From there, SQL views are used to transform data into a form that is optimized for PlanetTogether to import. PlanetTogether mappings wizard is then used to map data to PlanetTogether objects. Refer to [this Microsoft article](#) for more information on the D365 feature that enables exporting of entities into Azure SQL database (also known as the “Bring Your Own Database” method).

Deployment Steps

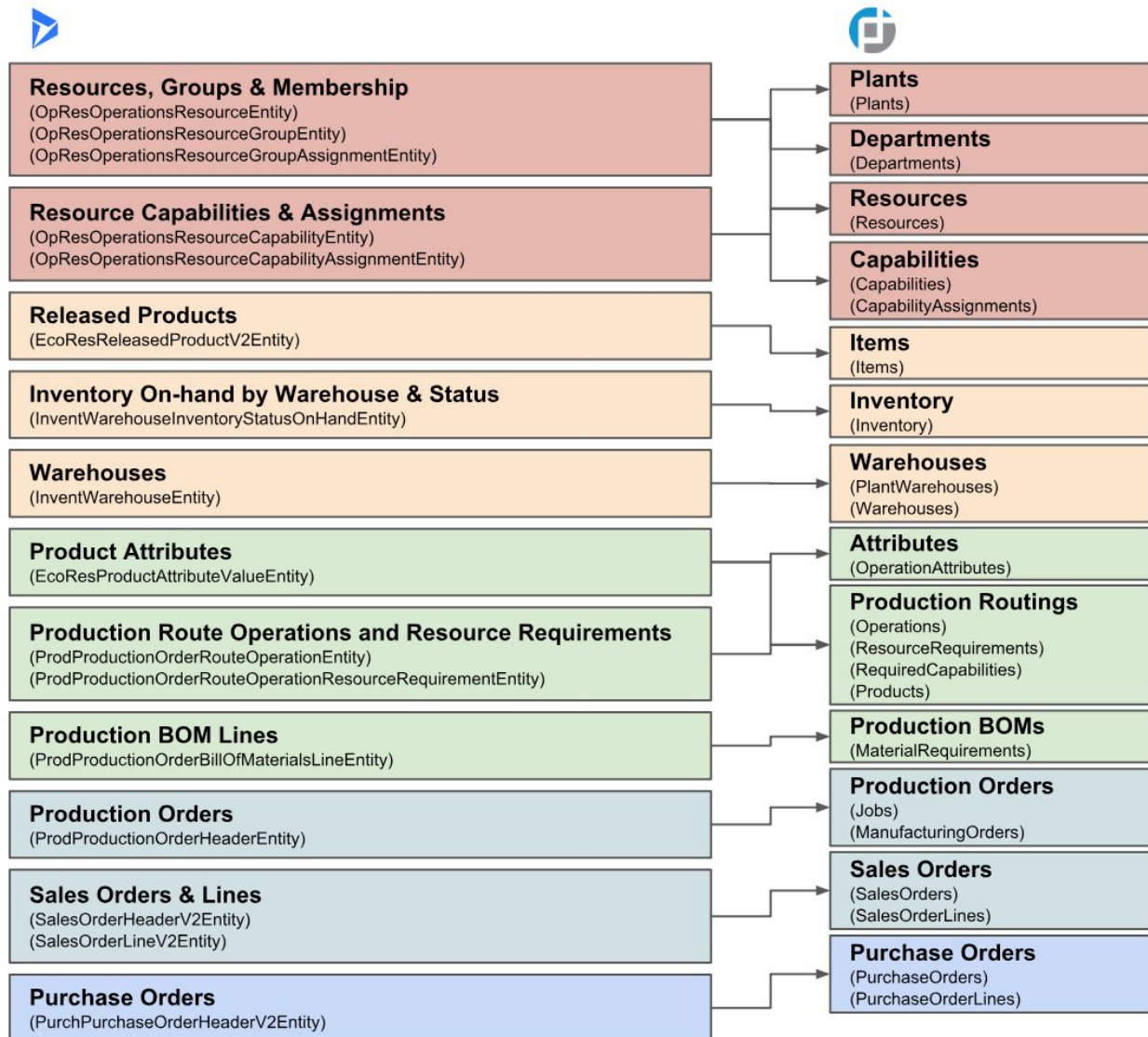
1. Create Azure SQL database
2. Configure entity export to database in D365
3. Enable Change tracking on entities below and publish them
4. Create export project and add entities below to it
5. Create views in Azure SQL database by running the provided ImportDbViews Query
6. In PlanetTogether, setup an import connection to the Azure SQL database
7. In PlanetTogether, load the provided mappings file APSInterfaceSettings.xml

Entities

Name	Target Entity	Standard
Sales order headers V2	SalesOrderHeaderV2Entity	Yes
Sales order lines V2	SalesOrderLineV2Entity	Yes
Purchase order headers V2	PurchPurchaseOrderHeaderV2Entity	Yes

Purchase order lines V2	PurchPurchaseOrderLineV2Entity	Yes
Warehouses	InventWarehouseEntity	Yes
Production bill of materials lines	ProdProductionOrderBillOfMaterialsLineEntity	Yes
Production route operations	ProdProductionOrderRouteOperationEntity	Yes
Production orders	ProdProductionOrderHeaderEntity	Yes
Inventory on-hand by warehouse and inventory status	InventWarehouseInventoryStatusOnHandEntity	Yes
Released Products V2	EcoResReleasedProductV2Entity	Yes
Capabilities	OpResOperationsResourceCapabilityEntity	Yes
Operations resource	OpResOperationsResourceEntity	Yes
Operations resource capability assignment	OpResOperationsResourceCapabilityAssignmentEntity	Yes
Resource groups	OpResOperationsResourceGroupEntity	Yes
Resource group membership	OpResOperationsResourceGroupAssignmentEntity	Yes
Production route operation resource requirements	ProdProductionOrderRouteOperationResourceRequirementEntity	Yes
Product attribute values	EcoResProductAttributeValueEntity	Yes

Mapping of D365 Entities to PlanetTogether Objects¹



¹ Standard Routings, Standard BOMs, and Forecasts are currently not supported

PlanetTogether to D365 Process

D365 can be updated with the latest scheduling data from PlanetTogether by running the Publish process from PlanetTogether. This process updates data in the PlanetTogether SQL database and then calls a program which queries this data and calls Web Services that are provided for D365.

1. Install custom services by loading the provided PTInboundService.axpp file.
2. Place PlanetTogether publish program PTAXPublish somewhere on the server.
3. Configure PlanetTogether to call program after publishing to database.
4. Program Arguments: PtPublishDbConnectionString AxURL ActiveDirectoryTenant
ActiveDirectoryClientAppId ActiveDirectoryClientAppSecret

D365 Objects and fields updated²

Procedure calls: UpdateDate, UpdateDeliveryDate, UpdateQuantity³, UpdateResource, UpdateStatus⁴

ProdTable			
Associated Procedure	D365 Table.Field	PT Table.Field	Description
UpdateDeliveryDate	ProdTable.DlvDate	Job.ScheduledEndTime	Requested date of delivery or finish report
	ProdTable.DlvTime	Job.ScheduledEndTime	Delivery time
UpdateStatus	ProdTable.ProdStatus	Custom	Production status 0 = Created 1 = CostEstimated 2 = Scheduled 3 = Released 4 = StartedUp 5 = ReportedFinished 6 = Completed
UpdateQuantity	ProdTable.QtyCalc	Job.Qty	Quantity cost estimated
	ProdTable.QtySched	ADD (Job.Qty - ProdTable.QtySched)	Scheduled production quantity
	ProdTable.RemainInventPhysical	ADD ((Job.Qty - ProdTable.QtySched) + ProdTable.RemainInventPhysical)	Quantity not yet reported as finished
UpdateDate	ProdTable.SchedStart	Job.ScheduledStartTime	The date that the project is scheduled to begin
	ProdTable.SchedFromTime	Job.ScheduledStartTime	Scheduled start time
	ProdTable.SchedEnd	Job.ScheduledEndTime	The date that the project is scheduled to finish

² D365 Documentation; <https://docs.microsoft.com/en-us/dynamics365/>

³ Used only if Quantities are being updated manually in PlanetTogether

⁴ Used only if D365 ProdTable.ProdStatus needs to be updated, this is not typical

	ProdTable.SchedToTime	Job.ScheduledEndDateTime	Scheduled end time
--	-----------------------	--------------------------	--------------------

ProdRoute			
Associated Procedure	D365 Table.Field	PT Table.Field	Description
UpdateQuantity	ProdRoute.CalcQty	Operation.RequiredFinishedQty	Calculated quantity in for this operation
UpdateDate	ProdRoute.FromDate	Operation.ScheduledStart	The date that the project is scheduled to begin
	ProdRoute.FromTime	Operation.ScheduledStart	Scheduled start time
	ProdRoute.ToDate	Operation.ScheduledEnd	The date that the project is scheduled to finish
	ProdRoute.ToTime	Operation.ScheduledEnd	Scheduled end time
UpdateResource	ProdRoute.WrkCtrlIDCost	JobResourceBlock.Resource.Name	The resource used for costing purposes

ProdRouteJob			
Associated Procedure	D365 Table.Field	PT Table.Field	Description
UpdateDate	ProdRouteJob.FromDate	Operation.ScheduledStart	The date that the project is scheduled to begin
	ProdRouteJob.FromTime	Operation.ScheduledStart	Scheduled start time
	ProdRouteJob.ToDate	Operation.ScheduledEnd	The date that the project is scheduled to finish
	ProdRouteJob.ToTime	Operation.ScheduledEnd	Scheduled end time
UpdateResource	ProdRouteJob.WrkCtrlID	JobResourceBlock.Resource.Name	Identify the resource