

* Customer Support - 1place - Data Conversion How To's (OneSource (V4) to 1place (V5))

Last Modified on 08/25/2026 6:15 pm EDT

Overview

How to perform a 'data conversion' from our existing customer's data files into our NEW cloud version.

Terminology

- **ERP:** Enterprise Resource Planning. (Everything in 1 database)
- **Customers** (Someone you sell something to)
 - Individual Contacts? Phone numbers, emails, etc...
 - Custom 'Prices' for specific items or 'types' of items?
 - Templates....
- **Vendors** (Someone you buy something from)
 - Individual Contacts?
- **Items (also known as 'Parts')**
 - Item Prices (Item X can have more than 1 price)
 - Item Vendors (Item X can have more than 1 vendor)
 - Item Catalog (Ways to find things by Category, Subcategory, or like Year, Make, Model, etc...)
 - Item Receipts (What I have in stock this very moment---who I bought it from, how much I paid for X quantity, where it is stored in the warehouse, etc)
 - Item Sales (To whom did I sell this item to, when and for how much for X number of Items)
 - Item Purchases (who I bought the item from, when, what the PO# is...)
- **Quotes**
 - Who did we quote, what item(s) to, when? For how much? Did they buy? (If so there will be a Sales Order #)
 - Customer Name, Date, who you spoke to, their PO # (unique # for them), which items? What prices? What quantities?
- **Sales Order**
 - Who bought what item(s) to, when? For how much?, Did they buy?
 - Customer Name, Date, who you spoke to, their PO # (unique # for them), which items? What prices? What quantities?
- **Invoice**
 - Who did quote what item(s) to, when? For how much?, Did they buy?
 - Customer Name, Date, who you spoke to, their PO # (unique # for them), which items? What prices? What quantities?
- **Invoice Payments**
 - Payments made on Invoices.
 - What payment came in, what date, how much was the payment? (\$45.67) What type of payment (Check, Credit Card, Venmo), what did it pay? (applied toward Invoice #123, #456)
- **Credit Memo**
 - Issued to the buyer to reduce the amount that the buyer owes under the terms of an earlier invoice.
- **Purchase Orders**
 - Who did I buy what from? What did I buy? How much did it cost? How many did I buy? Has it been 'received'?
- **Jobs**

- Projects
- Support Requests
- Tickets
- Etc...
- Documents
- Tasks

Big Picture Data Conversion Process (related to the GoLive on 1place)

Initial Data Conversion (Weeks or Months before GoLive)	Final GoLive Data Conversion (Hours before or after GoLive)	Fixing Issues Found By Us or Customer	Data Re- Conversion (Trial Run GoLive)?
Create Company	(done already)		
Users	(should be done)		
Warehouses	(no changes nec)		
Customers	(update any missing--day before)		
(related) Customer 'Contacts' (if any)	(update any missing--day before)		
(related) Customer 'Tasks' (if any)			
(related) Custom 'Pricing Templates' (if any)	(tell customer to maintain)		
(related) Customer 'Custom Pricing'	(tell customer to maintain)		
Vendors	(tell customer to maintain)		
(related) Vendor Contacts (if any)	(tell customer to maintain)		
Items	(update any missing--day before)		
(related) Item Quantities in Stock (per item, if any)	NIGHT or EARLY MORNING BEFORE		
(related) Item Price Levels (per item, if any)	(tell customer to maintain)		
(related) Item Vendors Items (Their Item #, Cost, etc) (if any)	(update any missing--day before)		
(related) Item Catalog (if any--if they sell auto parts)	(update any missing--day before)		
(related) Item Assemblies (if any--if they assembly items into kits)			
Quotes (We many times do <u>not</u> bring in quotes, or only 30 days worth)	?? Ask Customer?		
Quote Line Items	?? Ask Customer? Necessary? Do before or after?		

Sales Orders (Open, active, un-invoiced)	??? May or may not be necessary??		
Sales Order Line Items (Open)	???		
Invoices / Credits (Credits are in same table but have - values)	EARLY MORNING BEFORE OR NEXT EVENING AFTER		
Invoice / Credit Line Items	EARLY MORNING BEFORE OR NEXT EVENING AFTER		
Purchase Orders	EARLY MORNING BEFORE OR NEXT EVENING AFTER		
Purchase Order Line Items	EARLY MORNING BEFORE OR NEXT EVENING AFTER		

Training Videos for Data Conversion

Where to Locate?

- Go to Company Shared Drives > OneSource-Screencastify (Training Videos)

Training Topics:

- **1-V5 Data Conversion Training Videos (Folder1)**
 - 1 - 2020-DataConversion-Part1-How to Get V4 Customer Data File and Intro to DC Tool
 - 2 - 2020-DataConversion-Part2-Data Conversion Tool
 - 3 - 2020-DataConversion-Part3-Access Queries and Data Importation
 - 4 - 2020-DataConversion-Part4 (Resolving Importing Issues and Table Join Training)
 - 5 - 2020-DataConversion-Part5 (Resolving Problems for Imports)
 - 6 - 2020-DataConversion-Part6 (Initial Data Conversion, Other Tools)
 - 7 - 2020-DataConversion-Part7 (Overview and Recap of How to Start and Do).
 - 8 - 2020-DataConversion-Part8 (First Follow-up) 7-10-20
 - 9 - 2020-DataConversion-Reconciling AR Balances-7-17-2020
 - 10 - 2020-DataConversion-Updated the tool for FileLocations-LinkingTables--8-14-20.webm
 - 11-2020-DataConversion-Reconciliation-Ideas+Totals-Queries+DiffQueries-7-28-20
 - 12 - 2020-DataConversion-Invoices
 - 13 - 2021--DATA-CONVERSION-Advanced-HowTos--MergeUndatedPartsToANewPart#,FindXCharactersInAValue--5-13-21--sbc
 - 14 - 2020-DataConversion-DealingWithMaxLocksError-ImportingSmallerBatchesOfData--8/19/20 sbc
 - 15 - 2020-DataConversion-HowToConvertCustomPricingAndTemplates--8-21-20-sbc
 - 16 - 2020-DataConversion-HowToDeleteDataFromSQL+QueryHelp--2020.1111--sbc
 - 17 - 2020-DataConversion-Troubleshooting-LinItemsWontImport--2020.1209--sbc
- **1-V5 Data Conversion Training Videos (Folder1) for Non-V4 Clients**
 - 20 - 2020-DataConversion-Combining-Grouping-Totaling-Queries-NON-V4- Append Query etc./Servers--20
 - 21 - 2020-Alliance-DataConversion-NON-V4-Catalog,ItemSuppliers--2020-11-10--sbc
 - 22 - 2020--DataConversion--(NON-V4) HighLevelHowTos--BuildingQueries--BestPractices--20.1125--sbc.webm
 - 23 - 2020-Alliance-DataConversion-NON-V4-HowToBuildTheInventoryTableForExport--2020.1120--sbc.webm

(2026) CUSTOMERS table - DATA CONVERSION NOTES (PRIMARY TABLE)

Importance	1place Table	Field Name	Field Type	Field Notes
High	Customers	CustomerNumber	nvarchar(50), not null	This value must be unique. -- Duplicates not allowed. (If duplicate found, wizard will not ADD but rather UPDATE) -- NULL not allowed. -- The character length must not exceed 50 characters.
High	Customers	AddressType	nvarchar(50), null	This value must be one of these 3 hard coded options: Both (Bill To & Ship To) , or Bill To , or Ship To (If empty we will auto insert: 'Both (Bill To & Ship To)'
	Customers	IsSubCustomer	Yes/No	0 (for NO) or 1 (for YES) --Default will be 0
	Customers	ParentCustomer	nvarchar(50), null	This is rarely used, but when used must be another 1place Customer 'ID' field (which will be a GUID). --Default will be NULL
High	Customers	Active	Yes/No	0 (for NO) or 1 (for YES) --Default will be 1
High	Customers	Company	nvarchar(150), null	--If null or "" then company up with a way to create a Company NAME --If Duplicate...
	Customers	Prefix	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters.
	Customers	FirstName	nvarchar(50), null	-- Default null (OK to be empty). -- The character length must not exceed 50 characters.
	Customers	LastName	nvarchar(50), null	-- Default null (OK to be empty). -- The character length must not exceed 50 characters.
Low	Customers	Title	nvarchar(50), null	-- Default null (OK to be empty). -- The character length must not exceed 50 characters.
	Customers	Street	nvarchar(250), null	-- Default null -- The character length must not exceed 250 characters.
	Customers	Suite	nvarchar(50), null	This is 'Address Line 2' on Customer Detail screen. -- Default null -- The character length must not exceed 50 characters.

	Customers	City	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters.
	Customers	State	nvarchar(50), null	Usually a 2 digit code, such as CA for California, UT for Utah, etc... -- Default null -- The character length must not exceed 50 characters.
	Customers	Zipcode	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters.
	Customers	Region	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	Country	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	Phone	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	PhoneExt	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	Fax	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	CustomerType	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	CustomerSubType	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	Status	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	CompanyEmail	nvarchar(250), null	-- Default null -- The character length must not exceed 250 characters
	Customers	CompanyWebsite	nvarchar(100), null	-- Default null -- The character length must not exceed 100 characters
	Customers	AccountRep	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	Salesperson	nvarchar(50), null	This is a person's name, not a UserID. -- Default null -- The character length must not exceed 50 characters

High	Customers	ShippedVia	nvarchar(50), null	A user defined 'Text' value which must be equal to 1 of the values in the ShippedVia!ShippedVia (table!field). Might read like: Truck 1, Truck 2, etc... -- Default null -- The character length must not exceed 50 characters
High	Customers	Warehouse	nvarchar(50), null	This is NOT the GUID or the Warehouse Code, but rather than Warehouse Name. WarehouseLocations!CompanyName -- Default null -- The character length must not exceed 50 characters
High	Customers	DefaultPricing	nvarchar(150), null	Must be one of these hard coded values: Discount, Markup, Multiplier, Item List Price, Item Price Level -- Default null -- The character length must not exceed 50 characters
High	Customers	DefaultPricingDiscount	decimal, null	If Default Pricing = Discount, then this field stores the discount as a WHOLE number. 50 would = 50% -- Default 0
High	Customers	DefaultPricingMarkup	decimal, null	-- Default 0
High	Customers	DefaultPricingMultiplier	decimal, null	-- Default 0
High	Customers	DefaultPricingCode	nvarchar(50), null	If <u>Default Pricing</u> = Item Price Level, then this field stores the 'Price Level' created in ItemPricing This query shows the UNIQUE set of Price Levels in use: SELECT [MatrixPriceCode] FROM [dbo].[PricingLevels] Group By MatrixPriceCode
High	Customers	PaymentTerms	nvarchar(50), null	User defined Payment term name, such as Net 30. These values must match with one of the distinct values in the PaymentTerms!PaymentTerms Select PaymentTerms from Customers Group By PaymentTerms
	Customers	PaymentMethod	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	CreditLimit	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	CreditDays	bigint	-- Default 0
	Customers	TaxCode1	nvarchar(250), null	-- Default null -- The character length must not exceed 250 characters

	Customers	TaxExempt	Yes/No	-- Default null
	Customers	TaxExemptNum	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	QBOTaxExemptReasonForExemption	number, null	-- Default null
	Customers	FedTaxId	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	CreatedDate	date	-- Default null --The field must contain a valid date format.
	Customers	CreatedBy	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	ModifiedDate	date	-- Default null --The field must contain a valid date format.
	Customers	ModifiedBy	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Customers	QBOSync	Yes/No	-- Default null
	Customers	Comments	nvarchar(10000), null	-- Default null
	Customers	ReceivablesNotes	nvarchar(10000), null	-- Default null
	Customers	InternalNotes	nvarchar(10000), null	-- Default null
	Customers	FreightNotes	nvarchar(10000), null	-- Default null
	Customers	SICCode	nvarchar(255), null	-- Default null -- The character length must not exceed 255 characters
	Customers	LastCampaignDate	date	-- Default null --The field must contain a valid date format.
	Customers	LastCampaignMethod	nvarchar(255), null	-- Default null -- The character length must not exceed 255 characters
	Customers	LastCampaignPiece	nvarchar(255), null	-- Default null -- The character length must not exceed 255 characters
	Customers	NextCampaignMethod	nvarchar(255), null	-- Default null -- The character length must not exceed 255 characters
	Customers	NextCompaaignPiece	nvarchar(255), null	-- Default null -- The character length must not exceed 255 characters
	Customers	LastContactDate	date	-- Default null --The field must contain a valid date format.

	Customers	LastContacted	nvarchar(255), null	-- Default null -- The character length must not exceed 255 characters
	Customers	LastContactBy	number	-- Default 0
	Customers	NextContactDate	date	-- Default null --The field must contain a valid date format.
	Customers	NextContactBy	number	-- Default 0
	Customers	DeliveryFee	decimal	-- Default 0
	Customers	W9LastUpdated	date	Not available for import
	Customers	ParentCustomerID	nvarchar(150), null	--1place Customer 'ID' field (which will be a GUID). -- Default null -- If you are uploading from a Child company, please enter only the Parent Company ID in this field. It will create a customer in Parent Comapny as well.
	Customers	ChildCustomerCompanyID	number	--1place Company 'ID' field (like 4443, 4444). -- Default null -- If you are uploading from a Parent company, please enter only the Child Company ID in this field. It will create a customer in Child Comapny as well.

(2026) CUSTOMER CONTACTS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Customer Contacts	Customer Number	nvarchar(150), not null	
	Customer Contacts	PrimaryContact	Yes/No	
	Customer Contacts	FirstName	nvarchar(50), null	
	Customer Contacts	LastName	nvarchar(50), null	
	Customer Contacts	ContactType	nvarchar(50), null	
	Customer Contacts	Email	nvarchar(50), null	
	Customer Contacts	WorkPhoneDirLine	nvarchar(50), null	
	Customer Contacts	PhoneOther	nvarchar(50), null	
	Customer Contacts	WorkFax	nvarchar(50), null	
	Customer Contacts	CellPhone	nvarchar(50), null	
	Customer Contacts	PagerPhone	nvarchar(50), null	
	Customer Contacts	HomePhone	nvarchar(50), null	
	Customer Contacts	Birthday	date	
	Customer Contacts	ContactNotes	nvarchar(10000), null	

(2026) CUSTOMER CUSTOM PRICING table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Customer Custom Pricing	CustomerID	nvarchar(150), not null	

	Customer Custom Pricing	PricingTemplatesID	nvarchar(150), null	
	Customer Custom Pricing	ItemID	nvarchar(150), null	
	Customer Custom Pricing	PricingCategoryID	nvarchar(150), null	
	Customer Custom Pricing	PriceLevel	nvarchar(150), null	
	Customer Custom Pricing	QtyLow	real	
	Customer Custom Pricing	QtyHigh	real	
	Customer Custom Pricing	Discount	decimal	
	Customer Custom Pricing	Markup	decimal	
	Customer Custom Pricing	Multiplier	decimal	
	Customer Custom Pricing	SpecificPrice	decimal	
	Customer Custom Pricing	Lock	Yes/No	

(2026) CUSTOMER CUSTOM PRICING TEMPLATES table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Customer Custom Pricing Templates	PricingTemplatesID	nvarchar(50), not null	
	Customer Custom Pricing Templates	Description	nvarchar(255), null	
	Customer Custom Pricing Templates	ItemNumber	nvarchar(150), not null	
	Customer Custom Pricing Templates	PricingCategoryID	nvarchar(150), not null	
	Customer Custom Pricing Templates	PriceLevel	nvarchar(255), null	
	Customer Custom Pricing Templates	QtyLow	real	
	Customer Custom Pricing Templates	QtyHigh	real	
	Customer Custom Pricing Templates	Discount	real	
	Customer Custom Pricing Templates	Markup	real	
	Customer Custom Pricing Templates	Multiplier	decimal	
	Customer Custom Pricing Templates	SpecificPrice	decimal	

(2026) CUSTOMER API 3RD PARTY MAPPING table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
	Customer API 3rd Party Mapping	CustomerID	nvarchar(150), not null	
	Customer API 3rd Party Mapping	ThirdPartyType	nvarchar(150), null	
	Customer API 3rd Party Mapping	ThirdPartyCustomerID	nvarchar(150), null	

(2026) CUSTOMERS MERGE table - DATA CONVERSION NOTES (NOT PART OF DATA CONVERSION IMPORT WIZARD)

Importance	1place Table	Field Name	Field Type	Field Notes
High	Customers Merge	FromCustomerNumber	nvarchar(250), not null	
	Customers Merge	FromCustomerName	nvarchar(250), null	
High	Customers Merge	ToCustomerNumber	nvarchar(250), not null	
	Customers Merge	ToCustomerName	nvarchar(250), null	
Low	Customers Merge	ShipToCustomerNumber	nvarchar(250), null	
	Customers Merge	ShipToCustomerName	nvarchar(250), null	

(2026) VENDORS table - DATA CONVERSION NOTES (PRIMARY

TABLE)

Importance	1place Table	Field Name	Field Type	Field Notes
High	Vendors	VendorNumber	nvarchar(50), not null	This value must be unique. -- Duplicates not allowed. (If duplicate found, wizard will not ADD but rather UPDATE) -- NULL not allowed. -- The character length must not exceed 50 characters.
High	Vendors	Active	Yes/No	0 (for NO) or 1 (for YES) --Default will be 1
	Vendors	PoType	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	VendorAccountNumber	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
High	Vendors	VendorName	nvarchar(50), not null	-- The character length must not exceed 50 characters
	Vendors	Salutation	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	FirstName	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	LastName	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	Title	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	Street	nvarchar(255), null	-- Default null -- The character length must not exceed 255 characters
	Vendors	Suite	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	City	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	State	nvarchar(25), null	-- Default null -- The character length must not exceed 25 characters
	Vendors	PostalCode	nvarchar(75), null	-- Default null -- The character length must not exceed 75 characters
	Vendors	Country	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters

	Vendors	Phone	nvarchar(30), null	-- Default null -- The character length must not exceed 30 characters -- Must not contain non-numeric characters in order to import properly
	Vendors	Fax	nvarchar(30), null	-- Default null -- The character length must not exceed 30 characters
	Vendors	PaymentTerms	nvarchar(150), null	-- Default null -- The character length must not exceed 150 characters
	Vendors	CreditLimit	money	-- Default null
	Vendors	ProductsServices	nvarchar(255), null	-- Default null -- The character length must not exceed 255 characters
	Vendors	SICCode	int, null	-- Default null
	Vendors	EmailAddressCompany	nvarchar(100), null	-- Default null -- The character length must not exceed 100 characters
	Vendors	WebSite	nvarchar(100), null	-- Default null -- The character length must not exceed 100 characters
	Vendors	CreatedDate	date	-- Default null -- The field must contain a valid date format.
	Vendors	CreatedBy	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	ModifiedDate	date	-- Default null -- The field must contain a valid date format.
	Vendors	ModifiedBy	nvarchar(50), null	-- Default null -- The character length must not exceed 50 characters
	Vendors	QBOSync	Yes/No	-- Default null
	Vendors	Comments	nvarchar(10000), null	-- Default null

(2026) VENDOR CONTACTS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Vendor Contacts	VendorID	nvarchar(150), not null	
	Vendor Contacts	PrimaryContact	Yes/No	
	Vendor Contacts	FirstName	nvarchar(25), null	
	Vendor Contacts	LastName	nvarchar(25), null	
	Vendor Contacts	ContactType	nvarchar(50), null	
	Vendor Contacts	Email	nvarchar(50), null	
	Vendor Contacts	WorkPhoneDirLine	nvarchar(15), null	
	Vendor Contacts	PhoneOther	nvarchar(15), null	
	Vendor Contacts	WorkFax	nvarchar(15), null	

	Vendor Contacts	CellPhone	nvarchar(15), null	
	Vendor Contacts	PagerPhone	nvarchar(15), null	
	Vendor Contacts	HomePhone	nvarchar(15), null	
	Vendor Contacts	ContactNotes	nvarchar(10000), null	

(2026) ITEM LIST table - DATA CONVERSION NOTES (PRIMARY TABLE)

Importance	1place Table	Field Name	Field Type	Field Notes
High	Item List	Item Number	nvarchar(50), not null	
	Item List	Active	Yes/No	
High	Item List	Type	nvarchar(50), not null	
	Item List	Category	nvarchar(255), null	
	Item List	Sub Category	nvarchar(255), null	
	Item List	QualityIndicator	nvarchar(50), null	
	Item List	Item Description	nvarchar(10000), null	
	Item List	Manufacturer	nvarchar(50), null	
	Item List	OEMModelNumber	nvarchar(50), null	
	Item List	AverageCost	decimal	
	Item List	ListPrice	decimal	
	Item List	SpecialNotes	nvarchar(10000), null	
	Item List	LastOrderDate	date	
	Item List	UnitofMeasure	nvarchar(15), null	
	Item List	Bin Location	nvarchar(50), null	
	Item List	Weight	decimal	
	Item List	WeightUnit	nvarchar(15), null	
	Item List	UPCCode	nvarchar(100), null	
	Item List	Assembly	Yes/No	
	Item List	Taxable	Yes/No	
	Item List	QIS (Qty in Stock)	decimal	
	Item List	Min	real	
	Item List	Max	real	
	Item List	SpecialOrder	Yes/No	
	Item List	PricingCategory	nvarchar(150), null	
	Item List	PO Category	nvarchar(50), null	
	Item List	eCommerce	Yes/No	
	Item List	SubCategory2	nvarchar(255), null	
	Item List	SubCategory3	nvarchar(255), null	
	Item List	SubCategory4	nvarchar(255), null	
	Item List	Commissionable	Yes/No	
	Item List	MasterItemNumber	nvarchar(255), null	
	Item List	ItemNumAlt1	nvarchar(50), null	
	Item List	YearsListing	nvarchar(10000), null	
	Item List	OrigItemNum	nvarchar(50), null	
	Item List	OEMPrice	money	
	Item List	InterchangeNumber	nvarchar(50), null	
	Item List	FixedCostForMarkupPricing	money	
	Item List	ShipLength	decimal	
	Item List	ShipWidth	decimal	

	Item List	ShipHeight	decimal	
	Item List	DataConversionNotes	nvarchar(255), null	
	Item List	ItemNumAlt2	nvarchar(50), null	
	Item List	CreatedBy	number	
	Item List	CreatedDate	date	
	Item List	ModifiedBy	number	
	Item List	ModifiedDate	date	
	Item List	IncomeAccountRef	nvarchar(150), null	
	Item List	ExpenseAccountRef	nvarchar(150), null	
	Item List	AssetAccountRef	nvarchar(150), null	
	Item List	QBOSync	Yes/No	

(2026) ITEM QUANTITY IN STOCK table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Item Quantity in Stock	ItemID	nvarchar(150), null	
High	Item Quantity in Stock	QIS (Qty in Stock)	decimal, not null	
	Item Quantity in Stock	QtyReceived	decimal	
	Item Quantity in Stock	DateReceived	date	
High	Item Quantity in Stock	WarehouseID	nvarchar(150), not null	
High	Item Quantity in Stock	Last PO Cost	money, not null	
	Item Quantity in Stock	AdditionalCost	money	
	Item Quantity in Stock	Lot1	nvarchar(100), null	
	Item Quantity in Stock	Lot2	nvarchar(100), null	

(2026) ITEM PRICE LEVELS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Item Price Levels	ItemID	nvarchar(150), not null	
High	Item Price Levels	MatrixPriceCode	nvarchar(50), not null	
High	Item Price Levels	QtyLow	real, not null	
High	Item Price Levels	QtyHigh	real, not null	
	Item Price Levels	Price	money	
	Item Price Levels	DiscountPercent	real	
	Item Price Levels	FuturePrice	decimal	
	Item Price Levels	FuturePriceDate	date	
	Item Price Levels	LastPrice	decimal	
	Item Price Levels	LastPriceDate	date	

(2026) ITEM SEARCH CATALOG table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Item Search Catalog	ItemNumber	nvarchar(150), null	
	Item Search Catalog	Item Description	nvarchar(10000), null	
	Item Search Catalog	SubCategory3	nvarchar(100), null	
	Item Search Catalog	SubCategory4	nvarchar(100), null	
	Item Search Catalog	YearsListing	nvarchar(500), null	

	Item Search Catalog	Notes	nvarchar(10000), null	
	Item Search Catalog	YearRange	nvarchar(100), null	
	Item Search Catalog	Category	nvarchar(75), null	
	Item Search Catalog	Sub Category	nvarchar(75), null	
	Item Search Catalog	CatalogSource	nvarchar(50), null	
	Item Search Catalog	PlinkNumber	nvarchar(30), null	
	Item Search Catalog	OEMNumber	nvarchar(500), null	

(2026) ITEM BUNDLE (ASSEMBLY) COMPONENTS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
	Item Bundle (Assembly) Components	BundleComponentID	nvarchar(150), null	
	Item Bundle (Assembly) Components	ItemID	nvarchar(150), null	
	Item Bundle (Assembly) Components	QtyNeeded (Qty suggested to buy)	decimal	
	Item Bundle (Assembly) Components	AssemblyNotes	nvarchar(10000), null	

(2026) VENDOR ITEM NUMBERS (ALL VENDORS) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Vendor Item Numbers (All Vendor)	ItemID	nvarchar(150), not null	
High	Vendor Item Numbers (All Vendor)	VendorID	nvarchar(150), not null	
High	Vendor Item Numbers (All Vendor)	VendorItemNum	nvarchar(50), not null	
	Vendor Item Numbers (All Vendor)	VendorCost	money	
	Vendor Item Numbers (All Vendor)	VendorLeadTime	number	
	Vendor Item Numbers (All Vendor)	Notes	nvarchar(10000), null	
	Vendor Item Numbers (All Vendor)	CasePack	decimal	
	Vendor Item Numbers (All Vendor)	CostLastChangeDate	date	
	Vendor Item Numbers (All Vendor)	CubicSquareFt	decimal	
	Vendor Item Numbers (All Vendor)	Duty	decimal	
	Vendor Item Numbers (All Vendor)	RoundUpQtyToCaseQtyOnPO	Yes/No	

	Vendor Item Numbers (All Vendor)	ConvertQtyToCaseQtyOnPO	Yes/No	
	Vendor Item Numbers (All Vendor)	LastOrderDate	date	
	Vendor Item Numbers (All Vendor)	QtyOnOrder	decimal	
	Vendor Item Numbers (All Vendor)	VendorQtyInStock	real	
	Vendor Item Numbers (All Vendor)	VendorQtyInStockLastUpdated	date	

(2026) VENDOR ITEM NUMBERS (1 VENDOR) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Vendor Item Numbers (1 Vendor)	ItemID	nvarchar(150), not null	
High	Vendor Item Numbers (1 Vendor)	VendorItemNum	nvarchar(50), not null	
	Vendor Item Numbers (1 Vendor)	VendorCost	money	
	Vendor Item Numbers (1 Vendor)	VendorLeadTime	number	
	Vendor Item Numbers (1 Vendor)	Notes	nvarchar(10000), null	
	Vendor Item Numbers (1 Vendor)	CasePack	decimal	
	Vendor Item Numbers (1 Vendor)	CostLastChangeDate	date	
	Vendor Item Numbers (1 Vendor)	CubicSquareFt	decimal	
	Vendor Item Numbers (1 Vendor)	Duty	decimal	
	Vendor Item Numbers (1 Vendor)	RoundUpQtyToCaseQtyOnPO	Yes/No	
	Vendor Item Numbers (1 Vendor)	ConvertQtyToCaseQtyOnPO	Yes/No	
	Vendor Item Numbers (1 Vendor)	LastOrderDate	date	
	Vendor Item Numbers (1 Vendor)	QtyOnOrder	decimal	
	Vendor Item Numbers (1 Vendor)	VendorQtyInStock	real	
	Vendor Item Numbers (1 Vendor)	VendorQtyInStockLastUpdated	date	

(2026) ITEM CROSS REFERENCE table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Item Cross Reference	ItemNumber	nvarchar(250), not null	
High	Item Cross Reference	CrossReferenceNumber	nvarchar(250), not null	
	Item Cross Reference	ItemVendor	nvarchar(250), null	
	Item Cross Reference	ItemManufacture	nvarchar(250), null	
	Item Cross Reference	Notes	nvarchar(2000), null	

(2026) QUOTATION HEADERS (IMPORT 1ST) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Quotation HEADERS (IMPORT 1st)	QuotationNum	nvarchar(50), not null	
High	Quotation HEADERS (IMPORT 1st)	QuoteDate	date, not null	
High	Quotation HEADERS (IMPORT 1st)	BillToCustomerNumber	nvarchar(150), not null	
	Quotation HEADERS (IMPORT 1st)	BillToContactName	nvarchar(10000), null	
	Quotation HEADERS (IMPORT 1st)	BillToCompanyName	nvarchar(255), null	
	Quotation HEADERS (IMPORT 1st)	BillToStreet	nvarchar(250), null	
	Quotation HEADERS (IMPORT 1st)	BillToCityStateZipPostalCode	nvarchar(250), null	
High	Quotation HEADERS (IMPORT 1st)	ShipToCustomerNumber	nvarchar(150), not null	
	Quotation HEADERS (IMPORT 1st)	ShipToContactName	nvarchar(10000), null	
	Quotation HEADERS (IMPORT 1st)	ShipToCompanyName	nvarchar(255), null	
	Quotation HEADERS (IMPORT 1st)	ShipToStreet	nvarchar(250), null	
	Quotation HEADERS (IMPORT 1st)	ShipToCityStateZipPostalCode	nvarchar(250), null	
	Quotation HEADERS (IMPORT 1st)	SubTotal	money	
	Quotation HEADERS (IMPORT 1st)	FreightOrOtherCharges	money	
	Quotation HEADERS (IMPORT 1st)	TotalTax	money	
High	Quotation HEADERS (IMPORT 1st)	GrandTotal	money, not null	
High	Quotation HEADERS (IMPORT 1st)	PaymentTerms	nvarchar(150), not null	
	Quotation HEADERS (IMPORT 1st)	CustomerPONumber	nvarchar(50), null	
	Quotation HEADERS (IMPORT 1st)	SourceOfOrder	nvarchar(50), null	

	Quotation HEADERS (IMPORT 1st)	QuotationComments	nvarchar(10000), null	
	Quotation HEADERS (IMPORT 1st)	EnteredBy	nvarchar(50), null	
	Quotation HEADERS (IMPORT 1st)	ShippedVia	nvarchar(150), null	
	Quotation HEADERS (IMPORT 1st)	Warehouse	nvarchar(150), null	

(2026) QUOTATION LINE ITEMS (IMPORT AFTER HEADERS) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	QuotationNumber	nvarchar(150), not null	
High	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	ItemNumber	nvarchar(150), not null	
	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	Category	nvarchar(50), null	
High	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	Quantity	decimal, not null	
High	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	ItemDescription	nvarchar(10000), not null	
	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	UnitPrice	money	
	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	Discount	real	Set to 0 if none
	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	Taxable	Yes/No	
	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	Cost	decimal	
High	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	LineTotal	money, not null	
	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	ListPrice	money	
High	Quotation LINE ITEMS (IMPORT AFTER HEADERS)	NetPrice	money, not null	

(2026) SALES ORDERS HEADERS (UNSHIPPED) (DO THIS 1ST) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	SalesOrderNumber	nvarchar(22), not null	

High	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	SalesOrderDate	date, not null	
High	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	BillToCustomerNumber	nvarchar(150), not null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	BillToContactName	nvarchar(10000), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	BillToCompanyName	nvarchar(255), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	BillToStreet	nvarchar(250), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	BillToCityStateZipPostalCode	nvarchar(250), null	
High	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	ShipToCustomerNumber	nvarchar(150), not null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	ShipToContactName	nvarchar(10000), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	ShipToCompanyName	nvarchar(255), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	ShipToStreet	nvarchar(250), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	ShipToCityStateZipPostalCode	nvarchar(250), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	Warehouse	nvarchar(150), null	

	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	SubTotal	money	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	FreightOrOtherCharges	money	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	TotalTax	money	
High	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	GrandTotal	money, not null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	BalanceDue	money	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	SalesOrderStatus	nvarchar(150), null	The value in this field can only be 'New', 'To Print & Pick', 'Picking', 'To Invoice', or 'To Load/Ship'. If left blank the default value will be 'To Print & Pick'.
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	PaymentTerms	nvarchar(150), null	If this field is left blank the Net 30 payment term will be auto inserted.
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	CustomerPONumber	nvarchar(30), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	SourceOfOrder	nvarchar(50), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	SalesOrderComments	nvarchar(10000), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	EnteredBy	nvarchar(50), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	JobID	nvarchar(150), null	

	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	Salesman	nvarchar(30), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	Shippedvia	nvarchar(150), null	
	Sales Orders HEADERS (UNSHIPPED) (Do This 1st)	TaxExempt	Yes/No	

(2026) SALES ORDER LINE ITEMS (UNSHIPPED) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Sales Order LINE ITEMS (UNSHIPPED)	SalesOrderNumber	nvarchar(150), not null	
High	Sales Order LINE ITEMS (UNSHIPPED)	ItemNumber	nvarchar(150), not null	
	Sales Order LINE ITEMS (UNSHIPPED)	QtyOrdered	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	QtyShipped	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	BOQty	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	PrevShipped	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	ItemDescription	nvarchar(10000), null	
	Sales Order LINE ITEMS (UNSHIPPED)	ListPrice	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	UnitPrice	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	Discount	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	NetPrice	decimal	
High	Sales Order LINE ITEMS (UNSHIPPED)	Taxable	Yes/No, not null	
	Sales Order LINE ITEMS (UNSHIPPED)	LineTotal	money	=QtyOrdered x NetPrice
Low (optional)	Sales Order LINE ITEMS (UNSHIPPED)	DateShipped	date	
	Sales Order LINE ITEMS (UNSHIPPED)	LineItemComments	nvarchar(10000), null	
Low (optional)	Sales Order LINE ITEMS (UNSHIPPED)	SerialNumber	nvarchar(50), null	

	Sales Order LINE ITEMS (UNSHIPPED)	UnitCost	money	
	Sales Order LINE ITEMS (UNSHIPPED)	SOStatus	nvarchar(150), null	

(2026) SALES ORDER HEADER + LINES (1 FILE) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Sales Order HEADER + LINES (1File)	OrderSource	nvarchar(150), not null	This is the marketplace name like eBay, Amazon.
High	Sales Order HEADER + LINES (1File)	OrderNumber	nvarchar(150), not null	
High	Sales Order HEADER + LINES (1File)	ShipToName	nvarchar(250), not null	
	Sales Order HEADER + LINES (1File)	ShipToCompany	nvarchar(250), null	
High	Sales Order HEADER + LINES (1File)	ShipToAddress1	nvarchar(250), not null	
	Sales Order HEADER + LINES (1File)	ShipToAddress2	nvarchar(250), null	
High	Sales Order HEADER + LINES (1File)	ShipToCity	nvarchar(250), not null	
	Sales Order HEADER + LINES (1File)	ShipToState	nvarchar(250), not null	
	Sales Order HEADER + LINES (1File)	ShipToPostalCode	nvarchar(250), not null	
	Sales Order HEADER + LINES (1File)	ShipToPhone	nvarchar(250), null	
	Sales Order HEADER + LINES (1File)	ItemNumber	nvarchar(250), not null	
	Sales Order HEADER + LINES (1File)	ItemDescription	nvarchar(250), not null	
	Sales Order HEADER + LINES (1File)	ItemPrice	money, not null	
	Sales Order HEADER + LINES (1File)	ItemQty	decimal, not null	
	Sales Order HEADER + LINES (1File)	CustomerEmail	nvarchar(250), null	

(2026) SALES ORDER LINE ITEMS (for 1 Order) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Sales Order LINE ITEMS (UNSHIPPED)	SalesOrderNumber	nvarchar(150), not null	
High	Sales Order LINE ITEMS (UNSHIPPED)	ItemNumber	nvarchar(150), not null	
	Sales Order LINE ITEMS (UNSHIPPED)	QtyOrdered	decimal	

	Sales Order LINE ITEMS (UNSHIPPED)	QtyShipped	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	BOQty	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	PrevShipped	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	ItemDescription	nvarchar(10000), null	
	Sales Order LINE ITEMS (UNSHIPPED)	ListPrice	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	UnitPrice	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	Discount	decimal	
	Sales Order LINE ITEMS (UNSHIPPED)	NetPrice	decimal	
High	Sales Order LINE ITEMS (UNSHIPPED)	Taxable	Yes/No, not null	
	Sales Order LINE ITEMS (UNSHIPPED)	LineTotal	money	=QtyOrdered x NetPrice
Low (optional)	Sales Order LINE ITEMS (UNSHIPPED)	DateShipped	date	
	Sales Order LINE ITEMS (UNSHIPPED)	LineItemComments	nvarchar(10000), null	
Low (optional)	Sales Order LINE ITEMS (UNSHIPPED)	SerialNumber	nvarchar(50), null	
	Sales Order LINE ITEMS (UNSHIPPED)	UnitCost	money	
	Sales Order LINE ITEMS (UNSHIPPED)	SOStatus	nvarchar(150), null	

(2026) INVOICE & CREDIT MEMO HEADERS (IMPORT 1ST) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Invoice & Credit Memo HEADERS (IMPORT 1st)	InvoiceNumber	nvarchar(20), not null	
High	Invoice & Credit Memo HEADERS (IMPORT 1st)	InvoiceDate	date	

High	Invoice & Credit Memo HEADERS (IMPORT 1st)	BillToCustomerNumber	nvarchar(150), not null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	BillToContactName	nvarchar(200), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	BillToCompanyName	nvarchar(255), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	BillToStreet	nvarchar(250), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	BillToCityStateZipPostalCode	nvarchar(250), null	
High	Invoice & Credit Memo HEADERS (IMPORT 1st)	ShipToCustomerNumber	nvarchar(150), not null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	ShipToContactName	nvarchar(200), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	ShipToCompanyName	nvarchar(255), null	

	Invoice & Credit Memo HEADERS (IMPORT 1st)	ShipToStreet	nvarchar(250), not null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	ShipToCityStateZipPostalCode	nvarchar(250), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	Warehouse	nvarchar(150), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	SubTotal	money	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	FreightOrOtherCharges	money	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	TotalTax	money	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	1placeTaxCode	nvarchar(250), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	TaxExempt	nvarchar(250), null	

High	Invoice & Credit Memo HEADERS (IMPORT 1st)	GrandTotal	money, not null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	TotalPayments	money	If this value is left blank the BalanceDue will be equal to GrandTotal.
	Invoice & Credit Memo HEADERS (IMPORT 1st)	BalanceDue	money	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	InvoiceType	nvarchar(10), null	The value in this field could either be 'I' (for Invoice) or 'C' (for Credit Memo). If left blank the value can be derived from field TotalPayments. For a transaction where TotalPayments is a 'positive' number this field will automatically set to 'I' (and will be imported as an Invoice). If 'negative' number then 'C' (and will be imported as a Credit Memo).
	Invoice & Credit Memo HEADERS (IMPORT 1st)	PaymentTerms	nvarchar(150), null	If this field is left blank the Net 30 payment term will be auto inserted.
	Invoice & Credit Memo HEADERS (IMPORT 1st)	CustomerPONumber	nvarchar(200), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	SourceOfOrder	nvarchar(50), null	

	Invoice & Credit Memo HEADERS (IMPORT 1st)	Shippedvia	nvarchar(50), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	InvoiceComments	nvarchar(10000), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	EnteredBy	nvarchar(50), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	JobID	nvarchar(150), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	SalesMan	nvarchar(30), null	
	Invoice & Credit Memo HEADERS (IMPORT 1st)	InvoicePaymentDate	nvarchar(250), null	

(2026) INVOICE & CREDIT MEMO LINE ITEMS (IMPORT AFTER HEADERS) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	InvoiceNumber	nvarchar(150), not null	
High	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	ItemNumber	nvarchar(150), not null	
High	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	QtyOrdered	decimal, not null	

High	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	QtyShipped	decimal, not null	
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	BOQty	decimal	This is the quantity that have been Back Ordered and not Shipped yet. Set this value to 0 if none.
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	PrevShipped	decimal	
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	PrevQtyReturned	decimal	
High	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	ItemDescription	nvarchar(10000), not null	
Low (optional)	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	ListPrice	money	
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	UnitPrice	money	
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	Discount	real	
High	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	NetPrice	money, not null	
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	Taxable	Yes/No	
High	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	LineTotal	money, not null	=QtyOrdered x NetPrice
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	DateShipped	date	
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	LineItemComments	nvarchar(10000), null	
Low (optional)	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	SerialNumber	nvarchar(10000), null	
	Invoice & Credit Memo LINE ITEMS (IMPORT AFTER HEADERS)	UnitCost	money	

(2026) INVOICE & CREDIT MEMO (IMPORT Payments) table - DATA CONVERSION NOTES

Importance	Table	Field Name	Field Type	Field Notes
High	Invoice & Credit Memo (IMPORT Payments)	CustomerNumber	nvarchar(250), not null	
High	Invoice & Credit Memo (IMPORT Payments)	InvoiceNumber	nvarchar(250), not null	
High	Invoice & Credit Memo (IMPORT Payments)	InvoiceType	nvarchar(50), not null	Value in this field can be 'I' for Invoice payment or 'C' for Credit Memo payment.
High	Invoice & Credit Memo (IMPORT Payments)	InvoicePaymentDate	date, not null	
High	Invoice & Credit Memo (IMPORT Payments)	TotalPayment	decimal, not null	Nearly all payments (for both Invoice and Credit Memo payments) will normally be a positive number. Negative payments are only allowed to reverse (offset) positive payments (such as NSF/Bounced/Rejected payments).
High	Invoice & Credit Memo (IMPORT Payments)	PaymentTerms	nvarchar(250), not null	
	Invoice & Credit Memo (IMPORT Payments)	PaymentRefNumber	nvarchar(250), null	
	Invoice & Credit Memo (IMPORT Payments)	PaymentMethod	nvarchar(250), not null	
	Invoice & Credit Memo (IMPORT Payments)	DepositToAccount	nvarchar(250), null	
	Invoice & Credit Memo (IMPORT Payments)	ReceiptNumber	nvarchar(250), null	
	Invoice & Credit Memo (IMPORT Payments)	CreditCardNumber	nvarchar(250), null	

	Invoice & Credit Memo (IMPORT Payments)	AuthorizationNumber	nvarchar(250), null	
--	---	---------------------	---------------------	--

(2026) INVOICE & CREDIT MEMO (RECEIVE PAYMENTS SCREEN) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Invoice & Credit Memo (Receive Payments Screen)	InvoiceOrCreditMemoNumber	nvarchar(50), not null	
High	Invoice & Credit Memo (Receive Payments Screen)	AmountApplied	money, not null	This is the amount to apply toward the Invoice or Credit Memo.

(2026) PURCHASE ORDERS HEADER (IMPORT 1ST) table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Purchase Orders HEADER (IMPORT 1st)	PurchaseOrderNum	nvarchar(50), not null	
High	Purchase Orders HEADER (IMPORT 1st)	VendorID	nvarchar(50), not null	
High	Purchase Orders HEADER (IMPORT 1st)	PODate	date, not null	
	Purchase Orders HEADER (IMPORT 1st)	POStatus	nvarchar(150), null	If this field is left blank then 'Waiting For Delivery' will automatically be inserted in this field.
	Purchase Orders HEADER (IMPORT 1st)	OrderedBy	nvarchar(50), null	Enter a User Name that is included in your list of 1place users, or this field will be left blank
	Purchase Orders HEADER (IMPORT 1st)	Terms	nvarchar(50), null	Enter a Payment Term that is in 1place or this field will be left blank.

	Purchase Orders HEADER (IMPORT 1st)	FOB	nvarchar(50), null	
	Purchase Orders HEADER (IMPORT 1st)	ShippedVia	nvarchar(50), null	
	Purchase Orders HEADER (IMPORT 1st)	DateExpected	date	
	Purchase Orders HEADER (IMPORT 1st)	POReceived	Yes/No	This value needs to be = 'True' or 'Yes' or '1' if the PO has been received. Or 'False' or 'No' or '0' if the PO has NOT been received. If this field is left blank the value will be set to 'True'.
	Purchase Orders HEADER (IMPORT 1st)	Comments	nvarchar(10000), null	
High	Purchase Orders HEADER (IMPORT 1st)	BillToWarehouseCode	nvarchar(50), not null	
High	Purchase Orders HEADER (IMPORT 1st)	ShipToWarehouseCode	nvarchar(50), not null	
	Purchase Orders HEADER (IMPORT 1st)	ContainerNum	nvarchar(50), null	
	Purchase Orders HEADER (IMPORT 1st)	CountryOfOrigin	nvarchar(50), null	

	Purchase Orders HEADER (IMPORT 1st)	CreatedBy	nvarchar(150), null	This field must be a valid 1place User Name. If not it will be left blank
High	Purchase Orders HEADER (IMPORT 1st)	POTotal	money, not null	
	Purchase Orders HEADER (IMPORT 1st)	POWeight	decimal	
	Purchase Orders HEADER (IMPORT 1st)	POVolume	decimal	
	Purchase Orders HEADER (IMPORT 1st)	YourOrderNumber	nvarchar(50), null	
	Purchase Orders HEADER (IMPORT 1st)	BillToWarehouseStreet	nvarchar(250), null	
	Purchase Orders HEADER (IMPORT 1st)	BillToWarehouseCityStateZip	nvarchar(250), null	
	Purchase Orders HEADER (IMPORT 1st)	ShipToWarehouseStreet	nvarchar(250), null	
	Purchase Orders HEADER (IMPORT 1st)	ShipToWarehouseCityStateZip	nvarchar(250), null	
	Purchase Orders HEADER (IMPORT 1st)	VendorStreet	nvarchar(250), null	

	Purchase Orders HEADER (IMPORT 1st)	VendorCityStateZip	nvarchar(250), null	
	Purchase Orders HEADER (IMPORT 1st)	VendorFirstName	nvarchar(250), null	
	Purchase Orders HEADER (IMPORT 1st)	VendorLastName	nvarchar(250), null	

(2026) PURCHASE ORDERS LINE ITEMS (IMPORT AFTER HEADERS) table - DATA CONVERSION NOTES

Importance	Table	Field Name	Field Type	Field Notes
High	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	PurchaseOrderNumber	nvarchar(150), not null	
High	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemItemNumber	nvarchar(150), not null	
	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemReceived	Yes/No	
High	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemQtyOrdered	decimal, not null	

High	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemQtyReceived	decimal, not null	If you are importing OPEN / UNRECEIVED POs the Qty Received must be set to 0 and the Qty Ordered must be set to the amount of item not already received on the PO. In other words you cannot import partially received POs.
	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemDescription	nvarchar(10000), null	
	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemAddedCost	decimal	
High	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemCost	decimal, not null	
High	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemPOLineTotal	money, not null	
	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemWeight	real	
	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	LineItemVolume	real	
	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	Comment	nvarchar(10000), null	

	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	Lot1	nvarchar(50), null	
	Purchase Orders LINE ITEMS (IMPORT AFTER HEADERS)	Lot2	nvarchar(50), null	

(2026) BINS LOCATIONS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Bins Locations	WarehouseCodeName	nvarchar(150), not null	
High	Bins Locations	BinLocation	nvarchar(150), not null	
High	Bins Locations	BinPickingOrder	decimal, not null	
High	Bins Locations	BinZone	nvarchar(150), not null	

(2026) JOBS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
	Jobs	JobNumber	nvarchar(50)	
	Jobs	JobCustomerID	nvarchar(150)	
	Jobs	JobContactID	nvarchar(150)	
	Jobs	JobCompleted	Yes/No	
	Jobs	JobValue	money	
	Jobs	JobType	nvarchar(50)	
	Jobs	JobShortDesc	nvarchar(255)	
	Jobs	JobOrigDate	date	
	Jobs	JobExpCompDate	date	
	Jobs	JobCompDate	date	
	Jobs	JobStatus	nvarchar(50)	
	Jobs	JobLastResults	nvarchar(255)	
	Jobs	JobLastContactDate	date	
	Jobs	JobLastMeetingDate	date	
	Jobs	JobNextContactDate	date	
	Jobs	JobNextMeetingDate	date	
	Jobs	PriorityGrouping	nvarchar(50)	
	Jobs	PriorityItem	real	
	Jobs	EstTime	real	
	Jobs	JobSubType	nvarchar(50)	
	Jobs	ActualTime	real	
	Jobs	SalesMan	nvarchar(50)	
	Jobs	Comments	nvarchar(10000)	
	Jobs	FromEmailAddress	nvarchar(75)	
	Jobs	AssignedTo	nvarchar(50)	
	Jobs	Entered By User	nvarchar(50)	

	Jobs	EnteredDate	date	
	Jobs	ModifiedBy	nvarchar(50)	
	Jobs	ModifiedDate	date	
	Jobs	EmailToAddress	nvarchar(10000)	
	Jobs	CompanyName	nvarchar(150)	
	Jobs	CustomerCompanyName	nvarchar(150)	

(2026) TASKS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
	Tasks	TaskNumber	nvarchar(50)	
	Tasks	OrganizationID	nvarchar(150)	
	Tasks	ContactID	nvarchar(150)	
	Tasks	DateOfActivity	date	
	Tasks	ActivityTypeID	nvarchar(50)	
	Tasks	ActivityDescription	nvarchar(10000)	
	Tasks	ActivityResultsNotes	nvarchar(10000)	
	Tasks	UserSchedFor	nvarchar(50)	
	Tasks	EstTime	nvarchar(50)	
	Tasks	UserSchedBy	nvarchar(50)	
	Tasks	GroupPriority	nvarchar(50)	
	Tasks	ItemPriority	real	
	Tasks	FromEmailAddress	nvarchar(75)	
	Tasks	ActivityCompleted	Yes/No	
	Tasks	ToBeBilled	Yes/No	
	Tasks	StartTime	nvarchar(50)	
	Tasks	StopTime	nvarchar(50)	
	Tasks	TotalTime	decimal	
	Tasks	BillableTime	decimal	
	Tasks	DateTimeCompleted	date	
	Tasks	EmailToAddress	nvarchar(10000)	
	Tasks	TaskBilled	Yes/No	
	Tasks	BillableItem	nvarchar(150)	
	Tasks	Subject	nvarchar(10000)	
	Tasks	CompanyName	nvarchar(150)	
	Tasks	CustomerCompanyName	nvarchar(150)	
	Tasks	VendorCompanyName	nvarchar(150)	

(2026) USERS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Users	SecurityLevel	nvarchar(150), not null	
	Users	Department	nvarchar(50), null	
	Users	Phone	nvarchar(50), null	
High	Users	Email	nvarchar(50), not null	
	Users	Copy from Parent	bit	
	Users	Copy from User	nvarchar(100), null	
	Users	Password	nvarchar(100), null	
	Users	Salesperson	Yes/No	

	Users	DashBoard	Yes/No	
	Users	MaximumGrossMargin	real	
	Users	RestrictSOChangesAfterPrint	Yes/No	
	Users	RestrictInvoiceChangesAfterPrint	Yes/No	
	Users	RestrictSODeleteAfterPrint	Yes/No	
	Users	RestrictInvoiceDeleteAfterPrint	Yes/No	
	Users	HideCostFields	Yes/No	
	Users	AllowUpdatePOAfterAPBillCreated	Yes/No	
	Users	AllowExportingCustomerData	nvarchar(50)	
	Users	AllowExportingVendorData	nvarchar(50)	
	Users	AllowExportingJobData	nvarchar(50)	
	Users	AllowExportingPOData	nvarchar(50)	
	Users	AllowExportingTasksData	nvarchar(50)	
	Users	AllowExportingQuotesData	nvarchar(50)	
	Users	AllowExportingSOData	nvarchar(50)	
	Users	AllowExportingInvoiceData	nvarchar(50)	
	Users	AllowExportingCMDData	nvarchar(50)	
	Users	AllowExportingItemData	nvarchar(50)	
	Users	RestrictPriceChanges	Yes/No	
	Users	EnforceQuotationSinglePrint	Yes/No	
	Users	EnforceSalesOrderSinglePrint	Yes/No	
	Users	EnforceInvoiceSinglePrint	Yes/No	
	Users	EnforcePurchaseOrderSinglePrint	Yes/No	
	Users	EnforceCreditMemoSinglePrint	Yes/No	
	Users	HomeAddress	nvarchar(50)	
	Users	HomePhone	nvarchar(50)	
	Users	CellPhone	nvarchar(50)	
	Users	DateOfBirth	date	
	Users	DateOfHire	date	
	Users	AdditionalNotes	nvarchar(10000)	
	Users	LockSalesman	Yes/No	
	Users	DefaultWarehouseCode	nvarchar(150)	
	Users	WareHouseWorker	Yes/No	
	Users	Driver	Yes/No	
	Users	IsActive	Yes/No	
	Users	UserName	nvarchar(75)	
	Users	EnablePriceControls	Yes/No	
	Users	PriceControlsType	nvarchar(100)	

(2026) PORTAL USERS table - DATA CONVERSION NOTES

Importance	1place Table	Field Name	Field Type	Field Notes
High	Portal Users	CustomerEmail	nvarchar(250), not null	
High	Portal Users	CustomerNumber	nvarchar(250), not null	
	Portal Users	CompanyName	nvarchar(250), null	
	Portal Users	CustomerFirstName	nvarchar(250), null	
	Portal Users	CustomerLastName	nvarchar(250), null	

(2026) (INCOMPLETE) Step by Step Data Conversion OneSource

V4 to 1place (for our existing customers)

- Get copy of Customer's OneSource (ACCESS) data file.
- Connect to a COPY of YOUR own MS Access Data Conversion tool.
- Run Queries to create 'Work' Tables and transform the data IN the work tables.
- Export each data type into an Excel (or Comma or Tab delimited) File.
- In SSMS.exe
 - Import file into new table.
- ...

(2020+) Step by Step Data Conversion for OneSource (V4 / Access Version of 1place) Customers.

1. Hold a meeting with the Customer to determine (and take note of) which features they are currently using in OneSource V4 (to make sure there no 'show stoppers' -- meaning crucial features that are not part of the new cloud version of OneSource called 1place).
2. Create new 1place Company (at www.onesourcesoftware.com). Goto the Pricing and click on the link on the very bottom....to create a new company). (NOTE: This company may already exist from months or years before if we tried to get them to GoLive on 1place before....)
3. Login to Customer EXISTING OneSource Windows Server.
 - Open OUR 1place and look up the Customer name and find the login instructions in the NOTES of the customer. We usually login using RDP (Remote Desktop) and/or AnyDesk.
4. Find the OneSource Data files (listed below) and then zip them up into 1 or more upload file(s).
 - **xCustomerData.mdb** (where X is a variable name and data may or may not be in the file name). The file is usually a large file with their company name in the filename, and is usually located here:
C:\OneSource\Data or any other Drive letter like D or E, etc.. (In same cases the data file(s) will be located in another location designated by their network admin).
 - tblPricingGroups (Item Pricing Categories)
 - tblBuyingGroups (Pricing Template HEADERS)
 - tblBuyingGroupDefaults (Pricing Templates Line Items)
 - tblGroupPricing (Customer's Custom Pricing - usually copied over from Templates....)
 - Another Concept (to mention about PRICING tables...):
 - The OLD V4 version of OneSource...had 'Pricing Method' used by the Customer Default Pricing (field), AND/OR by the Custom Pricing Templates (to an extent....). The Pricing Method included methods like: Discount, Markup, Matrix By Code, Matrix by Qty, Markup by Average Cost, etc....
 - The NEW 1place version tries to simplify this....in the following 3 ways.
 - There are fewer of them.
 - The ones that are there are shorter and named more clearly and are less confusing.
 - The TEMPLATES no longer have a FIELD called Pricing Method:

- Example: V4 OneSource Version, template: Pricing Method: May have a value called *Matrix By Code*. Then have a field called Code where the Code was inserted.
- Example: 1place Version does NOT have a Pricing Method field. It just assumes it will be easier to understand and easier to use by just 'SHOWING' the columns that have the same type of settings. In the example above there IS NO Pricing Method such as Matrix by Code. There is just a field called Code (or Price Level) that is used makes it work like the V4 Matrix by Code--without having to specify a particular Pricing Method.
- Example 2: Rather than having a Pricing Method = Discount, and a discount field to store the Discount. Now you just enter a discount in the discount field and the system (and the user) know that a discount is what the user wants to do....
- Here is a list of the OLD pricing methods and what the equivalent is now:
 - V4 Pricing Methods 1place Pricing Methods
 - Discount Discount
 - Markup Markup
 - Markup By Average Cost Markup
 - Markup by X? Markup
 - Markup by Y? Markup
 - Matrix By Code Item Price Level
 - Matrix by Choice (dont support it any more)
 - Matrix by Qty Item Price Level Quantity
 - Code + Discount Item Price Level + Discount
- (These FILES MAY OR MAY NOT EXIST) If their 'data' file has been 'distributed' to split the data into 13-14 smaller databases then you will need to look for the same 'company data file' listed above AND these data files as well:
 - **osar.mdb** (AR)
 - Payments
 - **oscsa.mdb** (Contacts)
 - Customers
 - CustomerSpecific
 - Suppliers
 - tblActivities
 - tblContacts

- **osinv1.mdb** (Inventory1)
 - All tables that start with Inventory...
- **osinv2.mdb** (Inventory2)
 - All tables that start with tblInventory
- **osicm.mdb** (Invoices/Credit Memos)
 - All tables that start with Invoice
- **ospo.mdb** (Purchasing)
 - Purchase Order
 - Purchase Order Line Items
- **osqu.mdb** (Quotes)
 - Quotation
 - Quotation Lineitems
- **ossa.mdb** (Sales Orders)
 - Sales Orders
 - Sales Order Lineitems
 - Sales Order Lost Sales

5. Open up Chrome (or other browser) ON THE CUSTOMERS SERVER and log into our Google Account (drive.google.com) called **osFTPFiles@gmail.com** and drag/drop the zip file up to the FTP Files folder.

- **VERY IMPORTANT NOTE: You must SIGN OUT of our FTP Files account (on the Customer's server) or they will have access to our FTP Site, and very possibly other customers data being moved.**
- PLAN B: Login to your onesourcesoftware.NET Google account. Start a meeting. Send the user the meeting info. And then have them share their screen, and you advise them how to find, zip, copy over to Google Drive, and how to share the FOLDER the file is in with you.
- PLAN C: If the customer doesn't have a way to allow you to login, then send them STEP BY STEP instructions on how to find, zip up, and copy the zip file to their own Google Drive and how to share with you.
-

6. Log in to OUR Azure-V4 (using Remote Desktop) and IP address: 13.90.135.70 (NOTE: YOUR IP ADDRESS ON YOUR OWN PC WILL HAVE TO BE WHITELISTED on our Azure-V4 server in order to be able to login to the Azure server.

- Enter your user and password.

1. Then login to the osFTPFiles google account and drag and drop the zipped files to here:

- OneSource Data Conversions

- Make a new Folder with the Customers name and drag the zip file into that folder.
- Explore the files in the zip files and COPY and PASTE a COPY of the files to the same folder (retaining the zip file in its original state).
- Copy the data files over into the C:\OneSource Data Conversions\V4toV5 and then rename the Company Data File to: V4ToV5DataConversion-DATA.mdb (this will make the V4ToV5DataConversion-TOOL link the tables properly and prevent from a lot of manual table linking).
- MAKE A SUBFOLDER in the folder above, for the Customer you are converting data for.

7. Make a COPY of the **V4ToV5DataConversion-TOOL.mdb** (if there is only ONE company data file), or **V4ToV5DataConversion-TOOL-DIST.mdb** (if the company also has 13-14 '**distributed**' data files as well). (As of 4/27/20 this file does not exist yet...)

- Then add the Customer NAME to the end of the TOOL db so you will know which data you are linked to.... (since YOU are going to rename their data file to the generic database name above).
- For example, if doing a data conversion for ABCCompany, then you would rename the tool to:
V4ToV5DataConversion-TOOL-ABC.mdb

8. Open your (OneSource) Google Drive and search for **V4 to V5 - Import Tool - Data Conversion Tool TEMPLATE** and COPY it and add the Customer Name to the end of the File name and then (if necessary) clear out any record counts and totals.

9. Open the Google Doc called 'GoLive and Data Conversion - Master Checklist', then click on the Data Conversion 'tab' on the bottom, and follow these steps:

- Add the Company you are working on as a new column (just to the right of the instructions columns)
- Run all of the queries in the numeric Order, reading the notes in the Objective and Notes column.
- Update the results after executing each query with an X (meaning I completed this...) or a record count or total, following the example of past data conversions...
- Each time you come across a query that has EXPORT FILE in the name--right click on that file and select Export > Excel (to export the data for importing).
- Log into the Customers version of V5 and IMPORT the data from the Spreadsheet(s) that you just exported.
- If there are issues with the import (meaning x of the records could not be imported), look closer at the data and queries to see if any queries need to be added or modified to better prepare the data for importing.
- NOTE: In most cases if there are issues you will probably need to delete the data from V5 and re-run the queries for that data type once again--after you have modified or added new queries...
- NOTE: If you need to make a fix or change to any of the queries and then re-export be sure to go find the original exported Excel file and delete it. (If you don't it will just add another 'tab' inside of the file and you will just end up importing the exact set of older/incorrect data).

- Look at the data in V5 to get a good feel for if it looks correct.
- NOTE: Run 1 or more reports or lists to compare the # records imported, or the bottom line totals, such as the OPEN AR TOTALS, etc (they should match, except where certain exceptions apply).
- REPEAT THE PROCESS FOR EACH DATA TYPE.

Step by Step Data Conversion for NON-OneSource Customers.

- (NOTE: Please see the notes for the V4 data conversion steps and concepts above that will NOT be duplicated here).
- Hold a meeting with the Customer to determine (and take note of) which features they will NEED to use in V5.
- Create new OneSource (V5) Company (at www.onesourcesoftware.com)
- Ask the Customer to create an MSAccess file (with tables for each data type) or separate Excel files for each data type. (Remind them that we will need to get the data 2 or more times so they need to make good notes of how to make the same database, table, column, field, and excel NAMES--since we will be 'connecting' to these files by the actual database or excel and table/column/field names...)
- Make folder for Customer Data (to Store a virgin/untouched copy of the data)
- Make folder for Customer Import Tool. Place a COPY of the TOOL in that folder. Place COPY of the data in that folder. Make a NEW MS Access db called CustomerXData.mdb
- Open the CustomerXData.mdb and make links to the other db or excel files. Make numerous new queries (with logical naming conventions, to keep the ORDER of execution in Ascending order) that perform the operations to get that data from all the customer data sources into tables that have the same Table Names and Field Names as V4 data files--so you can use the tool to do final data conversion and exports.

How to Mass Delete ALL Records For A Customer Data Conversion (to Start Over)

See * [1place - Data Conversion - Mass Deletion \(Scripts\)](#)

(ADVANCED TOPIC) How to delete data in LOOPS (which is more controlled to STOP, but which is SLOWER)

How to DELETE TOP X RECORDS in a LOOP (1 table)

WHILE EXISTS (select top 1 * from TableName where CompanyID=101)

BEGIN

delete top(100) from TableName where CompanyID=101

END;

(Where 101 is the real companyID and TableName is the real table name).

How to DELETE TOP X RECORDS in a LOOP (When having to LINK to another table to get the related CompanyID)

(Note: Where x = to the COMPANY ID).

```
use dbOS000
```

```
go
```

```
Declare @CompanyID int = x
```

```
WHILE EXISTS (select top 1 * from InvoicePaymentLineItems where InvoicePaymentid in (select ID from InvoicePayments where CompanyID=@CompanyID))
```

```
BEGIN
```

```
delete top(1000) from InvoicePaymentLineItems where InvoicePaymentid in (select ID from InvoicePayments where CompanyID=@CompanyID)
```

```
END;
```

NOTE: Deleting in batches using BETWEEN 'X' and 'Y' is the FASTEST way to delete large amounts of data AND still break the job into smaller chunks. (Delete 100,000+ at a time runs as a full PASS / FAIL. If you stop it it fails. If it struggles with field constraints it may take 20 min to fail. Its best to break into smaller, more managable chunks.

How to Troubleshoot when data will NOT import properly

When the data you import is duplicated

- The Problem:
 - Sometimes when you EXPORT data to an Excel file and then try to open it you might get an error such as: We found a problem with some content in 'x file'. Do you want to try to recover as much as we can?
 - If you click NO then the file does not open up.
 - If you click YES then Excel tries to FIX the file by DUPLICATING the sheet and SAVING the original sheet to another HIDDEN sheet that is not visible. Then when you try to import the file it can try to DUPLICATE the records.
- The Solution.
 - Make a NEW Excel file. Highlight, copy, and paste the visible rows into the new file and save it.

When the 1place Import Wizard doesn't import ALL (or ANY) of the records in the Excel file

- Idea 1:
 - Each FIELD in 1place holds a certain 'type' of data. Each column on your spreadsheet must contain data that is acceptable for that column. You can VIEW the required data type (and the maximum field length) for each field when you open the Import Wizard, select an Import type, and then select a file to import. This then displays each database field that you can map to fields on the spreadsheet you have attached. To the right of the field name are some notes, including the data type and field length requirements.
 - The following are some common field types to become familiar with:
 - Text: Virtually ANY type of data will import into a Text field. However the 1place database field will be set to hold up to x characters, such as 150 characters in the ItemNumber field. If you try to import 101 or more characters SQL Server will reject the record.
 - Long Text field: This field is usually a NOTES or Comments type field that can store up to 64,000 characters.
 - Date: If you try to import text or numbers into a Date field SQL Server will record the record.
 - Integer: An 'integer' only holds NUMBER values between: -2,147,483,648 to 2,147,483,647. If you have a number outside of that number range, including a DECIMAL field, such as 12.12 SQL Server will reject the record.

- Boolean: A Boolean (or sometimes called a Bool or True/False field) only holds 0 or 1. 0 = No. 1 = Yes. An MS Access YES = -1. SQL is usually able to properly convert a -1 to a 1. It is also usually able to convert True or Yes to 1, and False or No to 0.
 - Single: A Single field can hold a larger whole or decimal value.
 - Double: A Double field can hold an even larger whole or decimal value.
- Solution #1: Make sure each column has the proper data type.
- **Idea 2:**
 - You've tried your best to match the proper data types and lengths, but it still doesn't import all of the records.
 - Run a NEW query in SSMS.exe (on the database that contains the potential error) with this text:
 - select top 100 * from ErrorLog where CompanyID=4287 order by Date desc
 - (Where the 1place company in this case is 4287)
- **Idea 3:**
 - If the ERROR leads you to believe it is NOT a data validation (improper data error), if the file is rather large, such as 10,000 or more records, try duplicating the file x times to import the data in smaller chunks.
- **Other Ideas:**
 - Make sure there are no DUPLICATE records with the same 'Primary Key' (such as the Item #, etc), unless the data type has duplicates like Invoice Line Items.
 - Make sure there are no DUPLICATE records because you have 1 or more records that have a BLANK field in the 'Primary Key' field.
 - Make sure that if you have Line Item linked to Transaction (header) X that that transaction does exist in the header table. (If Invoice # 123 on the Line Item import, make sure there is an Invoice # 123 in the Header table).
 - Try importing fewer columns, in order to narrow down which column(s) has the invalid data that cannot be imported.

Data Conversion How To's for OneSource V4 to OneSource V5

- Have a meeting with Customer. Make a google doc to make a plan of action. Decide on which data types will be imported.
- Make a COPY of the customer's OS Company Data file. Upload to Google. Download to C drive and store here: C:\OneSource\Data\{customername}
- Make ANOTHER copy of the data file and place it in this folder: C:\OneSource\Data\DataConversion\ and then rename it to: DataConversionData.mdb
 - Create Excel Export Files for each of the desired data types in the sections below.
 - (HOLD) Open up the data conversion DB created for these data conversions located here: S:\Dropbox\Databases\V5-SQL.accdb
 - This should AUTO LINK to the primary tables in the DataConversionData.mdb db.
- Run the queries that start with @DB...

Customer Data

SQL to Export key fields to a WORKING table (query called: @Export_100_MT_Customers_WorkingTable):

```
SELECT Customers.[Customer Number], Customers.Active, Customers.[Customer Record Type], Customers.[Type of Customer], Customers.[SubType of Customer], Customers.Region, Customers.Status, Customers.EmailAddressCompany, Customers.[First Name], Customers.[Last Name], Customers.Title, Customers.Company, Customers.Street, Customers.Suite, Customers.City, Customers.State, Customers.[Zip/Postal
```

Code], Customers.Country, Customers.Phone, Customers.Fax, Customers.EmployeeRepresentative, Customers.Comments, Customers.[Payment Terms], Customers.CreditHold, Customers.AdminCreditHold, Customers.[Shipped Via], Customers.[Bill To Customer Number], Customers.wwwSite, Customers.DefaultWarehouse, Customers.DateAdded, Customers.DateEdited, Customers.SourceTypeID, Customers.SICCode, Customers.ProductsServices, Customers.NumEmp, Customers.[Tax Exempt], Customers.[Tax Exempt Number], Customers.FedTaxID, CustomerSpecific.[Pricing Type], CustomerSpecific.[Discount Percent], CustomerSpecific.[Markup Percent], CustomerSpecific.Multiplier, CustomerSpecific.Code, CustomerSpecific.Salesman, Customers.[Credit Limit], Customers.CreditPastDueDays, Yes AS QBOSync

FROM Customers LEFT JOIN CustomerSpecific ON Customers.[Customer Number] = CustomerSpecific.[Customer Number]

WHERE (((Customers.Active)=True));

QUESTIONS / ISSUES TO RESOLVE:

- How to map the Pricing Types? (What are the values?)
- How to map the Warehouse 'Main'? Will it work with just the word Main?
- How to map QBO Payment Terms? (When we bring in that is not already in OS it does NOT show it and it does NOT add it).
- ~~How to map QBO Shipped Via?~~ (There is no table or list in QBO for Shipped Via)
- How to map QBO Sales Tax Codes? (Just leave blank?)
- Credit Limit? Does it sync with QBO?
- Credit Days? Does it sync with QBO?
- QBOTaxExemptReasonForExemption? 'Resale'
- Tax Exempt vs Fed Tax ID. (When add a customer it shows the Tax Resale #, but it does NOT show up on the Customer screen, but the Fed Tax ID # does)....
- How link up User Names (like for Sales Rep field or Customer Service / Employee Representative)?
- Need to Add 'NextCampaignDate'
- Need to Add 'CreditHold', 'AdminCreditHold', 'SourceType', 'ProductsServices', 'NumEmployees', CustServiceRep, SalesRep,
- Do we have a field or a way to see a POPUP for the Comments field (like in v4?)

Working Table, Queries to 'clean up' data.

-

Default Pricing

- This data is stored in the CustomerSpecific table. The following table explains V4 vs V5:

V4 - Default Pricing Type	V5 - Default Pricing Type	Notes
Discount	Discount	
Discount Code	Item Price Level + Discount (Also copy the Discount % over to the Discount field.)	
Markup	Markup (copy over the markup % to the markup % field).	
Multiplier	Multiplier (copy over the Multiplier % to the Multiplier % field).	
Specific	Not supported as a Default Pricing Method. Make a query to see if any exist and inform the customer.	
Matrix by Code	Item Price Level (Copy the Price Level or Code into the field that stores the price level)	
Matrix by QTY	Not supported as a Default Pricing Method. Make a query to see if any exist and inform the customer.	

Matrix by Choice	Not supported as a Default Pricing Method. Make a query to see if any exist and inform the customer.	
First	Not supported as a Default Pricing Method. Make a query to see if any exist and inform the customer.	
Markup up by Divisor	Not supported as a Default Pricing Method. Make a query to see if any exist and inform the customer.	
Markup up by Divisor Actual	Not supported as a Default Pricing Method. Make a query to see if any exist and inform the customer.	
Markup up by Average Cost	Markup	

Custom Pricing Templates (v4 TABLE NAMES)

- **tblGroupPricing**
 - This is where individual customer 'custom' prices are stored.
 - The user can HAND enter or BULK copy prices from a 'Template', which are stored in the tblBuyingGroups & tblBuyingGroupDefaults
 - The user can also put a Pricing Group on the template, to price all parts with x group/category with 1 simple discount or markup. These groups or categories are stored in the tblPricingGroups table (in V4)
 - **tblBuyingGroups**
 - This is the (header) table that stores the NAME of the template and the unique template ID.
 - **tblBuyingGroupDefaults**
 - This is the (details) table that stores the individual 'template' prices.
 - To save lots of time, a Customer can COPY ALL of the Custom Prices for any ONE or MORE of the Templates to Any particular Customer. When the Copy takes place it copies the rows from Template X to the tblGroupPricing table for that Customer, with that Customer's # and the tblBuyingGroup # in the table.
 - **tblPricingGroup**
 - This was called a 'Product Pricing Group' but is now called 'Pricing Category'.
 - This is a table that stores various NAMES to Categorize parts, by the part type for Pricing purposes.
 - Generally ALL items have a Pricing Category assigned to them, but do not have to. Note: 1 item can only be in 1 Pricing Category.

Vendor Data

Item List Data

Item Inventory Levels

Item Vendor Prices

How the Data Conversion Tool 2015 Works - Setup & Big Picture

- NOTE: The original copies of these objects should be in the Version Control - Apps\DataConversionTool2015\DataDefs.mdb
- TABLES: Import these objects into the NEW DATA file you are going to give the new customer (if they are not already in the DataDefs.mdb):
 - @DataConversionTool_Parameters
 - @DataConversionTool_Steps

- @DataConversionTool_TableViews
- QUERIES: Import ALL queries that start with @dc15- AND @DCTool_
- FORMS:
 - @frmDataConversionTool
- MODULES:
 - DataConversionTool
 - fnDCExecute (this is the function that you call from the Immediate windows.
 - Note: Just open the data file these objects are in, press Ctrl+G, type fnDCExecute in the 'Immediate' window and then press ENTER.
 - This will show, in the immediate window, each of the steps the function completes. It will also create a detailed LOG file in the same directory called DataConLog.txt.
 -
- NOTE: Some of the queries have HARD CODED Values. These need to be changed to Variables which are listed on the first tab of the form/tool.

Data Conversion Extraction Techniques and Ideas

Find the Data Files...

- Ask the customer if they know what 'type' of database stores their data. If they know this info this will be of help.
- Right click on the ICON for the application that creates and uses the data. Make note of the 'Target' path and click Open File Location...
- Look at each of the folders to see if an 'obvious' data folder exists...
 - If not, open the Windows Explorer and browse to the C drive (or whichever drive the app is using) and do a search (maybe for file *.*) to find a bunch of files that have been changed within the last few seconds, or minutes. Active accounting systems have data added and changed every x seconds or Minutes...
 - The goal is to 1--find a batch of files that are large enough to hold significant amount of data (usually several MB's to a few Gigabytes in size), 2--Find files that have been changed within the last x seconds or minutes (assuming users are actively using the system), and 3--Find files that have logical file names like Cust or Customer, Ven, Vendor, AR..., AP... INV, Invoices, Trans, Transactions, Inventory, etc... basically file names that hold that types of data you are looking for...

Figure out how to 'open' the Data File...

- Once you have located the data files look at the 'extension' of the files to see if you recognize it, such as .mdb (microsoft access), .sql (mysql), .mdf (MS SQL Server), .dbf (DBase and others), etc. If it is any of these you can typically use Access to open and view the data (using the ODBC driver applet below).
 - If you do NOT recognize the file format (an example is .nx1 format for a data file I was trying to figure out while writing this kba) then search the internet for the file format and odbc driver. For instance: nx1 odbc driver (This will help you find an ODBC driver that can be used by MS Access to provide a proper connection to the file to extract data from the file into MS Access.

Create an ODBC 'DSN' (Data Source Name) that connects to the file

- Open Control Panel > Administrative Tools > ODBC Data Sources
- Follow steps (on the internet) on how to create an ODBC DSN that you can 'link' to from MS Access. (This article is an example of that...: <https://www.mssqltips.com/sqlservertip/1470/microsoft-access-and-sql-server-integration/>)

Select Queries

How to Make a Query (Select Query)

1. Select Create > Query Wizard
2. Select Simple Query, and then OK.
3. Select the table that contains the field, add the Available Fields you want to Selected Fields. and select Next.
4. Choose whether you want to open the query in Datasheet view or modify the query in Design view, and then select Finish.

How to Filter for a fixed value

- 1.

How to Filter for a value contained in a string value

More to come...

How to add a Line Feed or Carriage Return in a query

To select records with a carriage return at the end of a field, create a query and enter the following in the Criteria line for that field: Like "*" & Chr(13) & Chr(10) (Access used carriage return + line feed, characters 13 and 10, for a new line).

Update Queries

Make Table Queries

Append Queries

Delete Queries

Using VBA Functions IN your Queries to Help Transform or Convert Data

How to see a LIST of VBA Functions

- http://www.techonthenet.com/excel/formulas/index_vba.php

Nz function: How to replace NULL values with a value, such as 0.

This is a great function to use on QTY, and Price, and Total fields to find any fields that have a NULL value and replace the NULL with another value such as 0.

Label: nz([FieldName],0) Where the label is whatever you want to call the cell, and field name is whatever field name you want to use.

Trim function: To strip off leading and trailing blank spaces off a field.

Left function: How to view the LEFT x Characters in a field

Right function: How to view the RIGHT x Characters in a field

Mid function: How to view X characters in the MIDDLE of the field

InStr function:

Function InStr([Start], [String1], [String2], [Compare As VbCompareMethod = vbBinaryCompare])

InStrRev function:

Function InStrRev(StringCheck As String, StringMatch As String, [Start As Long = -1], [Compare As VbCompareMethod = vbBinaryCompare]) As Long

StrComp function:

This 'compares' the values of field x with field y. Such as: strcmp([FieldX],[FieldY]). If the result = 0 then the value are the SAME. If the result = -1 then the values are NOT the same.

Split function:

IsNumeric function: (How to determine if a value is a Numeric, or in other words a NUMBER.

Example: Item number: HO1000254N

We know that PartsLink #s are 2 TEXT CHARACTERS Followed by 7 NUMBERS.. So if you want to determine if the part IS (or is not) an item with the exact format of a Partslink #, then you could run a few functions combined that look like this:

isNumeric(Left([ItemNum],2))

Using the combination of the isNumeric and Left functions above you could analyze to see if the LEFT 2 characters of the ItemNum field are Numeric. The return value would be -1 if TRUE or 0 if False.

How to take the dash "-" out of a phone field.

- Replace(PhoneNumberField, "-", "")

How to strip a variable length number or value off the end of a field.

- As long as they are all followed by a particular character or space, then you can use the following expression to return just the value you want into another field:
- QueryColumnTitleX: Right([Field X],Len([Field X])-InStrRev([Field X],""))

How to strip the cents off of a price (the digits to the right of the decimal point)

- x: `CCur(Left([Payment Amount],InStr(CStr([Payment Amount]),".")))`

How to link 2 tables together by fields that are of different types (when you get a type mismatch error when trying to combine them)

- Example:
 - Table A value: Number (Integer, meaning it is a number field that cannot have any decimal values)
 - Table B value: String (text)
- Solution:
 - Create a query that converts either of the Table values to both of them.
 - In the query put as a field (column) the field that is a number:
 - ValXNum: (this will be the normal field that is a number already)
 - ValYText: `cstr([number field])`
 - Note
 - To convert a value to a double (if that is the field 'type') then the function would be: `cdbl()`
 - To a single: `csgl()`
 - To a date: `cdate()`
 - To a yes/no, boolean, `cbool()`
 - To an integer: `cint()`

How to Determine if x charaters in a field are numeric or not.

- Example A: Left x charaters: `x: IsNumeric(Left([FieldX],2))`
- Example B: Mid x chars...: `y: IsNumeric(Mid([FieldX2],3,7))`

Example: We want to identify which parts in the Item list are 'Partslink'. Partslink parts have 2 TEXT Characters in position 1 and 2 of the item #, and then 7 numbers in position 3-9. So, in order to find all parts that have TTNNNNNNNN (where T = Text and N = Number) we would need to put these two columns in the query:

X: `IsNumeric(Left([ItemNumber],2))` Then set the Criteria = 0 (In this case we are looking at the LEFT 2 characters, where the Characters are TEXT values. In this case the IsNumeric returns a 0, or in other words NOT a Numeric Value, so we set the Criteria on the field = 0.

Y: `IsNumeric(Mid([ItemNumber],3,7))` Then set the Criteria = -1 (In this case we are looking at the MIDDLE of the field, starting at Character 3 and then looking at the 7 Characters from the 3rd position (including the 3rd position). In this case the IsNumeric returns a -1, or in other words YES (-1) the value IS Numeric. So we set the Criteria on the field = -1.

See the picture below. Then we UPDATE the Field X with Value X, if it meets our criteria...

How to COMPARE VALUES

best access vba function to compare to values
ChatGPT said:

In Access VBA the “go-to” comparison function is `StrComp` (for strings). For numbers/variants you usually just use the normal comparison operators. Here’s the breakdown so you can pick the right one.

1. Comparing strings (best choice): `StrComp`

```
If StrComp(value1, value2, vbTextCompare) = 0 Then  
    ' they match (case-insensitive)  
End If
```

- 3rd argument controls case:
 - `vbBinaryCompare` → case-sensitive
 - `vbTextCompare` → case-insensitive
- Returns:
 - `0` = equal
 - `-1` = first < second
 - `1` = first > second
 - `Null` if either is Null

This is nicer than `If a = b Then` when you care about **case** or **locale**.

2. Comparing numbers/dates: just use `=`, `<`, `>`

```
If num1 = num2 Then ...  
If d1 > d2 Then ...
```

Access VBA handles numeric/date comparisons well without a special function.

3. Dealing with Nulls: use `Nz` or `StrComp`

If there's a chance of Null (common in Access):

```
If StrComp(Nz(v1, ""), Nz(v2, ""), vbTextCompare) = 0 Then ...  
If Nz(v1, 0) = Nz(v2, 0) Then ...
```

or

4. Quick inline compare for fields: `IIf`

```
IIf(StrComp(f1, f2, vbTextCompare) = 0, "Same", "Different")
```

So: use **StrComp** for strings, plain operators for numbers/dates, and wrap with **Nz** if Nulls are possible.

OTHER 'CUSTOM' Data Conversion Query Functions / Tools (NOTE: These FUNCTIONS are Custom made and MUST be in your Access DB to work)

CleanPhoneNumber

CleanPhoneNumber([x])

ParsePersonName (Misc Functions in OneSource)

ParseCityStateZip (Misc Functions in OneSource)

'ltk 06.1102

'ParseCityStateZip("Huntsville AL 35803","City") : 'returns 'Huntsville'

'ParseCityStateZip("Huntsville AL 35803","State") : 'returns 'AL'

'ParseCityStateZip("Huntsville AL 35803","Zip") : 'returns '35803'

CompressToYearRange (Misc Functions in OneSource)

'This function will take a list of years and produce a year range

'ltk 07.0827

,

'Example: CompressToYearRange("94,95,96,97,98") would output: "94-98"

Function CompressToYearRange(Str As Variant, Optional digits As Integer = 2) As Variant

ExpandYearRange (Misc Functions in OneSource)

'This function will build a list of years given a range

'ltk 08.0701 - Fix when range end with a comma, ",".

'ltk 08.0701 - added option to handle multiple year ranges and a mixture of single years in same string

'Example: ExpandYearRange("91-94,9900") would output: 91,92,93,94,99,00

'Example: ExpandYearRange("91-94,96-99") would output: 91,92,93,94,96,97,98,99

,

'ltk 07.0827 - added 4 digit years option (this feature does not force the output to 4 digits it just handles the input of 4 digits)

'ltk 07.0402 - added additional error handling if there is no range: i.e. input is 98,02

'ltk 06.0927 - modified by LTK - originally written by Craig and Andy

,

'Example: ExpandYearRange("1996,1997") would output: "96,97"

'Example: ExpandYearRange("94-8") would output: "94,95,96,97,98"

'Example: ExpandYearRange("94-98") would output: "94,95,96,97,98"

'Example: ExpandYearRange("94-02") would output: "94,95,96,97,98,99,00,01,02"

'Example: ExpandYearRange("1994-98") would output: "94,95,96,97,98"

'Example: ExpandYearRange([SubCategory1]):' if run from within a query

Function ExpandYearRange(Str As Variant, Optional digits As Integer = 2) As Variant

Getting Familiar With MS Access / VB / VBA FUNCTIONS to help you Convert or Transform Data in your MS Access Queries

In MS Access you can browse to: Create > Visual Basic > View > Object Browser to OPEN up the list of OBJECTS (including ALL functions used in VBA that you can use in your MS Access Queries to 'Convert' data. The example highlighted below is for the CStr (which is a short abbreviated FUNCTION for CONVERTING the data to a STRING (or in other words TEXT).

See Also...

[VBA How To's and Coding Tip's](#)

SQL TRAINING Classes

- SQL FULL COURSE for Beginners (30 Hours) from Zero to Hero: <https://www.youtube.com/watch?v=SSKVgrwhzus>
 - Learn SQL Server, and Structured Query Language (SQL) inside and out - so you can create queries on the fly in SQL (SQL Server Management Studio) - Which adds, deletes, and updates data in SQL Server 100x faster than MS Access (via ODBC).
- Type Conversion Functions in Microsoft Access: CCur, CDate, CDbt, CDec, CLng, CStr: <https://www.youtube.com/watch?v=f0f4xo8S-3Y>
- MS ACCESS COURSES: <https://www.youtube.com/@599CD/courses>

How to Extract Data from a DBF File with the Headers:

- Open the DBF File using the DBF Plus (a free DBF Viewer). -Check SJ Trading Folder under Nivea
- Choose the DBF File to view.
- Export the File as Excel Sheet (This file will not include the Headers so you need to copy and paste them manually using the following steps).
- Open the Excel file after the exporting.
- Go back to the DBF Viewer.
- Click "Table Info" from the options above.
- After, a pop-up will be shown (Check image below):

- Click the "Clipboard" icon to copy all the Header columns.
- Paste to the Excel sheet and choose "Transpose" so the Header will be arranged as columns.
- Save the file and you can start using it with MS Access.

Older Notes =====NOT APPLICABLE FOR NOW=====

- Create new (or get access to existing) QuickBooks Online (QBO) account
 - Setup Lists:
 - Chart of Accounts
 - Product Categories (Make these match the Product Categories in existing OneSource)
 - Locations (If applicable...Make these match the list of Warehouse Codes in existing OneSource)
 - Payment Methods (In OneSource the Payment Methods and Payment Terms were one in the same, so no matching up with OneSource for this one...)
 - Terms (Make these match the list of Payment Terms in OneSource).
 - Setup Bank Accounts (including online banking sync)
- Determine HOW we will BALANCE the 'BALANCE SHEET' from V4 into V5 (Cloud Version):
 - OPTION 1: IMPORT ONLY THE 'OPEN, UNPAID INVOICES'...
 - This option is the best balance of getting the QuickBooks Online BALANCE SHEET in Balance AND still being able to send out statements (for Invoices and Credits imported in from OneSource) FROM QuickBooks Online.
 - Drawback: Use WILL be able to send 'Open Item' Statements but WILL NOT be able to send an accurate 'Transaction' or 'Balance Forward' type statements.
 - STEPS:
 - 1 - User will EXPORT ALL UNPAID Invoices and Credit Memo's from OneSource to Excel.
 - 2 - User will IMPORT the Excel file using the Import Wizard option (for Invoices and Credit Memo's) called: Invoices & Credit Memo's (UNPAID ONLY)
 - 3 - (OPTIONAL) If 'Sales History is desired in OneSource then the user will:
 - EXPORT ALL FULLY PAID Invoices and Credit Memo's from OneSource to Excel.
 - IMPORT the Excel files using the Import Wizard option (for Invoices and Credit Memo's) called: Invoices & Credit Memo's (PAID ONLY) (NOTE: NO PAYMENTS will be imported and it will NOT Sync these PAID transactions with QuickBooks Online)
 - GL: Hand enter a SINGLE JOURNAL ENTRY to represent the BALANCE SHEET (less the OPEN AR BALANCE--since the OPEN INVOICES will represent the AR balance on the Balance Sheet).
 - OPTION 2: IMPORT ALL OPEN AND PAID INVOICES, CREDITS, and PAYMENTS (after a certain cut-off date)
 - This is NOT a good option because it bringing in several months or years of historical info ALL affect the QuickBooks Online Balance Sheet--which makes it very, very difficult to get the Balance Sheet accurate and properly balanced.
 - OPTION 3: IMPORT NONE of the INVOICE, CREDIT MEMO'S or PAYMENTS..
 - This is really, really simple to do (after creating a simple, single journal entry that represents the Balance Sheet as of that moment in time).
 - However this is NOT a good option for the following reasons:
 - ALL AR Payments would have to be entered in BOTH systems.
 - Statements would have to be printed out and sent to the customer from BOTH systems--for an undetermined length of time (most likely several months or even years).

Data Exports

OneSource SUPPLIERS

- Create a special EXPORT query IN the customers 'Company Data File' database, following these steps:
 - Creation of SUPPLIER EXPORT LIST:
 - Create (tab) > Query Design > (Then click Close on the Show Table popup window) > Design (tab) > SQL (Then back-space over the text in the Query, where it says: **SELECT**; so the window will be completely blank when done).
 - Next Copy and Paste the following SQL code into the blank SQL Query screen:
 -
 - `SELECT Suppliers.[Supplier Number], Suppliers.Active, Suppliers.[Supplier Account Number], Suppliers.[Type of Vendor], Suppliers.[Supplier Name], Suppliers.[First Name], Suppliers.[Last Name], Suppliers.Title, Suppliers.Street, Suppliers.Suite, Suppliers.City, Suppliers.State, Suppliers.[Zip/Postal Code], Suppliers.Country, Suppliers.Phone, Suppliers.Fax, Suppliers.[Payment Terms], Suppliers.[Credit Limit], Suppliers.[Fed Tax ID], Suppliers.Comments, Suppliers.wwwSite, Suppliers.DateAdded, Suppliers.DateEdited, Suppliers.SICCode, Suppliers.ProductsServices, Suppliers.OrigSupID, Suppliers.ShipVia, Suppliers.RemitToSupplierName, Suppliers.RemitToFirstName, Suppliers.RemitToLastName, Suppliers.RemitToAddress1, Suppliers.RemitToAddress2, Suppliers.RemitToCity, Suppliers.RemitToSate, Suppliers.RemitToZipPostalCode, Suppliers.RemitToCountry, Suppliers.EmailAddressCompany, Suppliers.[Entered By], Suppliers.DataConversionNotes FROM Suppliers WHERE (((Suppliers.Active)=Yes));`
 - Then click the Design (tab) and then click the RED exclamation point to run the SQL Code.
 - Then right-click your mouse over the rows and columns (in the top left corner) and select Copy.
 - Then open up a new Excel file and paste the contents into the excel file.
 - Creation of Different REMIT TO addresses comparison list (to give to the customer to manually review and update/fix):
 - Create (tab) > Query Design > (Then click Close on the Show Table popup window) > Design (tab) > SQL (Then back-space over the text in the Query, where it says: **SELECT**; so the window will be completely blank when done).
 - Next Copy and Paste the following SQL code into the blank SQL Query screen (then click the Design (tab) and then click the RED exclamation point to run the SQL Code):
 -
 - `SELECT Suppliers.[Supplier Number], Suppliers.[Supplier Name], Suppliers.Active, Suppliers.Street, Suppliers.RemitToAddress1, Suppliers.RemitToAddress2, Suppliers.RemitToCity, Suppliers.RemitToSate AS RemitToState, Suppliers.RemitToZipPostalCode, StrComp([Street],[RemitToAddress1]) AS x FROM Suppliers WHERE (((Suppliers.Active)=True) AND ((StrComp([Street],[RemitToAddress1]))<>0));`

OneSource CUSTOMERS

- Step by Step (with Video):
 -
- Step by Step (without a video):
 - Create special EXPORT query IN the customers 'Company Data File' database, following these steps:
 - CREATE NEW QUERY WITH NO TABLES:
 - Create (tab) > Query Design > (Then click Close on the Show Table popup window) > Design (tab) > SQL
 - Then back-space over the text in the Query called **SELECT**; (so it will be completely blank when done).
 - PASTE SQL INTO BLANK QUERY:
 -
 -

- SELECT Customers.[Customer Number], Customers.[Bill To Customer Number], Customers.Active, Customers.[Customer Record Type], Customers.[Type of Customer], Customers.[SubType of Customer], Customers.Region, Customers.[First Name], Customers.[Last Name], Customers.Title, Customers.Company, Customers.Street, Customers.Suite, Customers.City, Customers.State, Customers.[Zip/Postal Code], Customers.Country, Customers.Phone, Customers.Fax, Customers.EmployeeRepresentative, Customers.[Tax Exempt], Customers.[Tax Exempt Number], Customers.[Credit Limit], Customers.CreditPastDueDays, Customers.Comments, Customers.[Payment Terms], Customers.CreditHold, Customers.AdminCreditHold, Customers.DefaultWarehouse, Customers.[Shipped Via], Customers.wwwSite, Customers.EmailAddressCompany, Customers.DateAdded, Customers.DateEdited, Customers.SICCode, Customers.ProductsServices, Customers.NumEmp, Customers.NextCallDate, Customers.LastCalldate, Customers.LastCallResults, Customers.[Pricing Type], Customers.[Discount Percent], Customers.[Markup Percent], Customers.Multiplier, Customers.Code, Customers.AddedBy, Customers.EditedBy, Customers.Status, 1 AS QBOSync, 1 AS QBOResync INTO [@OS_Export_Customers_WorkTable] FROM Customers INNER JOIN CustomerSpecific ON Customers.[Customer Number] = CustomerSpecific.[Customer Number] WHERE (((Customers.Active)=True));
- CREATE WORKING TABLE (So you can see the data and update the date, where necessary)
 - Right-click on the new Query (tab) and select 'Design View'. (This will show the tables on top and the fields on the bottom).
 - Then right-click on the open gray area and select: Query Type > Make Table Query
 - Then type (in the Table Name field): **@OS_Export_Customers_WorkingTable** and then click OK.
 - Then click on the Design (tab) and then click the Run icon. Then click Yes when prompted. (Note the # of records to be exported should be a number larger than 0).
 - (VIEW THE DATA) Now on the left side, upper left corner, there will be a drop down list called something like 'All Access Objects' or 'Tables' etc. Drop the list down and select 'Tables'.
 - Now scroll down and and OPEN the newly created table called:
@OS_Export_Customers_WorkingTable
- VIEW DATA AND UPDATE KEY FIELDS(to more accurate data that works properly with QuickBooks Online)
 - Then open up a new Excel file and paste the contents into the excel file.
- Update (Special consideration to...)
 -
 - Payment Terms
 - 1 = Cash (QBO default)
 - 2 = Check (QBO default)
 - 3 = Credit Card (QBO default)
 - Warehouse
 - Tax Exempt
 - Default Pricing Type
 - Discount
 - Markup
 - Multiplier
 - Item Price Code
 - Item List Price
 -

OneSource INVENTORY PRODUCTS

OneSource INVENTORY PRICE

OneSource INVOICES

OneSource PURCHASE ORDERS

Data Imports

- Import Data into new OneSource
 - Customers > Customers
 - Suppliers > Vendors
 - Inventory > Items
 - Inventory Products > ItemStock
 - Inventory Prices > ItemVendors
 - Invoices > Invoices
 - Purchase Order > Purchase Orders

Keywords

OneSource Data Conversion Momentum Notes
