Quantum Utility Quick Guide

Contents

[Overview 3](#_Toc27388840)

[Main Features 7](#_Toc27388841)

[Other Options 8](#_Toc27388842)

[Suggested Future Modifications 8](#_Toc27388843)

[Issues 8](#_Toc27388844)

[Questions 9](#_Toc27388845)

[Referenced Documents 10](#_Toc27388846)

[Code change record log 11](#_Toc27388847)

# Overview

Quantum Utility is a command-line utility that automatically FTPs files down from the UPS server for the current day and backs them to a local AWS drive (\\172.31.251.161\sdixdata\ftp\). The Windows Task Scheduler should run the executable on a daily basis at 11:45pm (to catch all daily exports provided by hour).

#### QuantumView Overview

UPS generates numerous plain text files per day/per hour containing shipping information, unique tracking numbers and variations of the SDI purchase order number. UPS could generate anywhere from 3-12 files daily (while adding additional files throughout the day). The utility searches each file for records of type:

* E1 - Exception (packages that indicate any shipping issues),
* D1 -Delivery Short (basic delivery information provided a couple hours after delivery) and
* D2 – Delivery Long (records that provide more complete information).

Based on the provided plain-text tracking number and variations of the purchase order id found in the utility searches the SDI database for matches on the following fields

| Field Name | UPS QuantumView Field(s) | SDI DatabaseTable.Field(s) | Description\Mapping Logic |
| --- | --- | --- | --- |
| Tracking Number | TrackingNumber | PS\_ISA\_ASN\_SHIPPED.ISA\_ASN\_TRACK\_NO | This table is only 1 of 2 PeopleSoft tables that the SDIExchange id can access. M. Randall suggested using the PS\_ISA\_ASN\_Shipped table for potential Tracking Number matches.  Note: 90% of the time the tracking number is not stored in the SDI database when we receive the files. The only match M. Randall found in the DB was on 1Z1639370312224053 in the QVD\_ALT\_SDIINC\_20091119\_150122\_223\_SDIQVD.txt file.  A Purchase Order could have multiple line entries. Each line entry will have its own tracking number. Using the Tracking number as a PK, developer can pull the affiliated line number and PO ID from the database. If Tracking number does not match, users must use the PO\_ID provided by UPS. |
| Purchase Order ID | ShipmentReferenceNumber Values 1&2 and PackageReferenceNumber Values 1&2 | PS\_PO\_LINE\_SHIP.PO\_ID or  PS\_PO\_HDR.PO\_ID | The SDI ID is typically around 10 characters (the older versions were purely numbers, the newer versions start with 1-2 alphabetic characters and end with digits.  Initially, the code searched PS\_PO\_HDR which takes less time to search (initially recommended by M. Randall) but shifted to PS\_PO\_LINE\_SHIP which takes more time (upon noticing the InsiteOnline3.1 codebase (CommentDetails.aspx.vb or repostatus.aspx.vb ) use that table in their queries  UPS stores variations of the PO ID in 4 different fields. Sometimes just in ShipmentReferenceNumberValue2 or PackageReferenceNumberValue1, sometimes both ShipmentReferenceNumber1 and PackageReferenceNUmberValue2 (see QVD\_ALT\_sdiinc\_20191125\_170154\_826\_SDIQVD.txt), sometimes all 4 and sometimes only 1. As a result, the query searches each.  Additionally, UPS stores PO ID in the UPS Reference Fields sometimes with spaces, special characters (‘PO.R010115686’), or extra characters in front (‘PO#: ‘ , ‘SDI PO GR02857913‘), but most times as the standard SDI POID format (PM00068987). See QVD\_ALT\_SDIINC\_20191119\_120127\_313 for example variations.  The utility code runs through, strips the known special characters and attempts to match the possible variations to the PO\_ID in the SDI DB. Dev Team will need to update the QuantumUtility.StripChars() function if encountering new variations.  Most times a match is made on PO ID. Unfortunately, UPS typically does not provide line number, meaning if the tracking number does not exist in the SDI DB, there is no way to know which purchase order item the entry belongs to. |

Once the codebase finds a match on PO ID (and/or Tracking Number), it adds a new line to the PS\_AS\_XPD\_COMMENT Notes\_1000 field. Then users can view the results via the UI by navigating to the Expedite PO Update Page, searching for that specific PO ID and clicking the comments history icon. Note: Because the Comments table uses PO ID and Line Number as a combination key, sometimes users might not be able to see the data (if the code cannot find a line number match).

If the utility does or does not find a match on the tracking number or purchase order id, an alert is written to the SDIX\_UPS\_QUANTUMVIEW\_LOG table.

Code errors are written to the SDIX\_UPS\_QUANTUMVIEW\_ERROR table. Also, if a file does not contain D1, E1 or D2 entries, it is skipped entirely. The message is written to the Quantum View Error table.

#### code installation

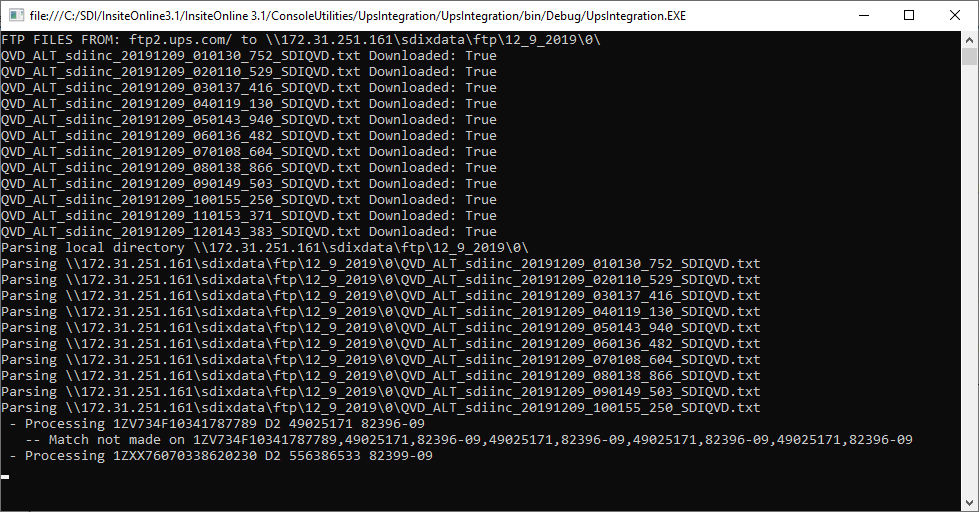
Users should install the entire UPSIntegration package on the server (which contains the necessary WInSCP FTP and Quantum View support utility files).

At the command line, users can either type UpsIntegration.exe (or UpsIntegration) to run that day’s download or UpsIntegration.exe dd//mm//yyyy to pull down for a specific day. Note: users must use that specific date format. Otherwise, the application defaults to current day.

The utility quickly parses each file row until it finds an E1, D1 or D2 match and then verifies if the POID/TrackNum exists in the DB. All search results are output to the console and SDIX Quantum LOG table.

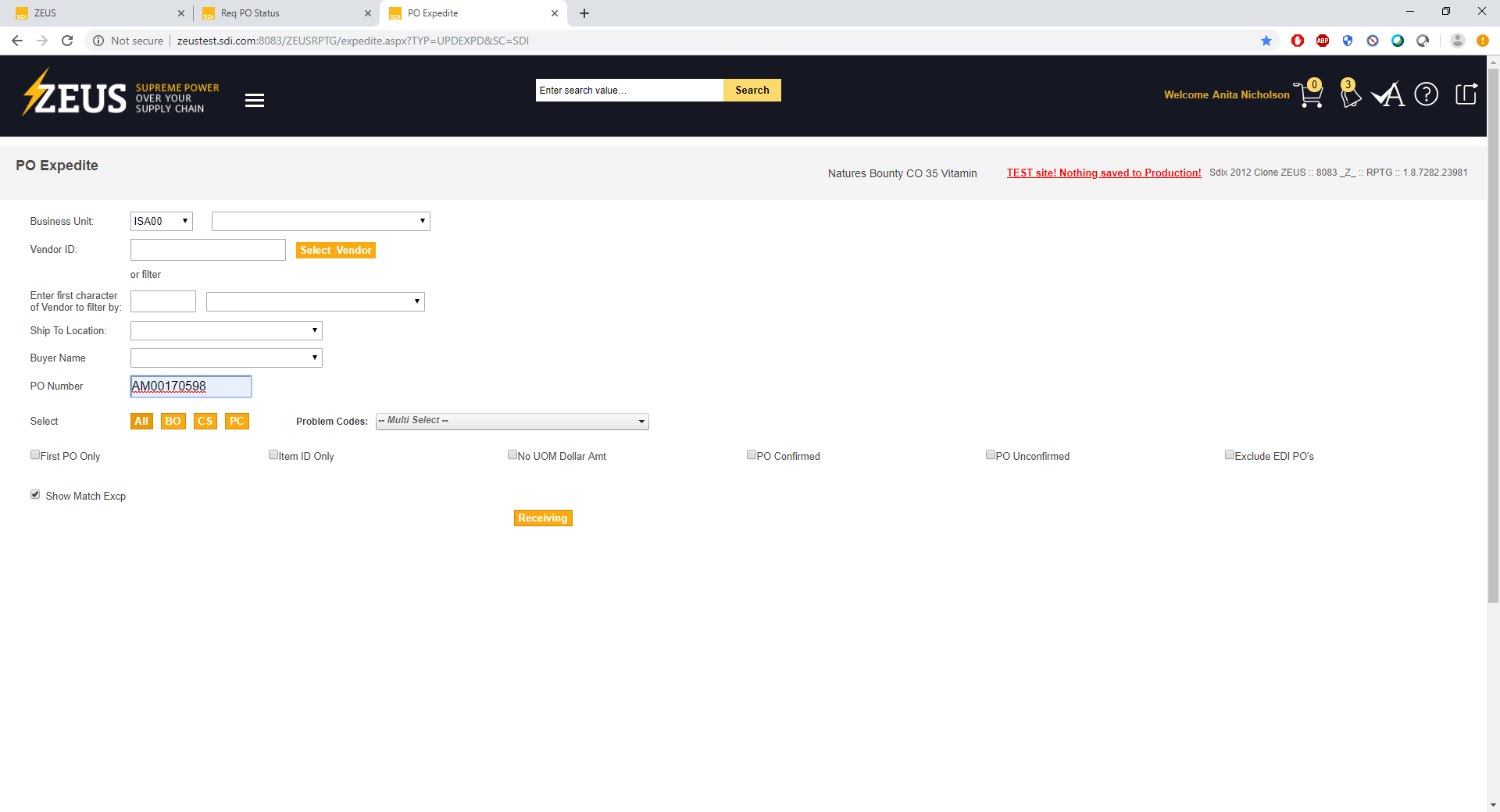
If a file does not contain the E1, D1, or D2 type, the application skips it entirely and pushes an alert to the SDIX Quantum Error table.

Back up files are saved to AWS (\\172.31.251.161\sdixdata\ftp\). by the date the application is run and stored in a subfolder containing the files pulled down for the inputted date i.e. \\172.31.251.161\sdixdata\ftp\12\_12\_2019\12\_5\_2019\



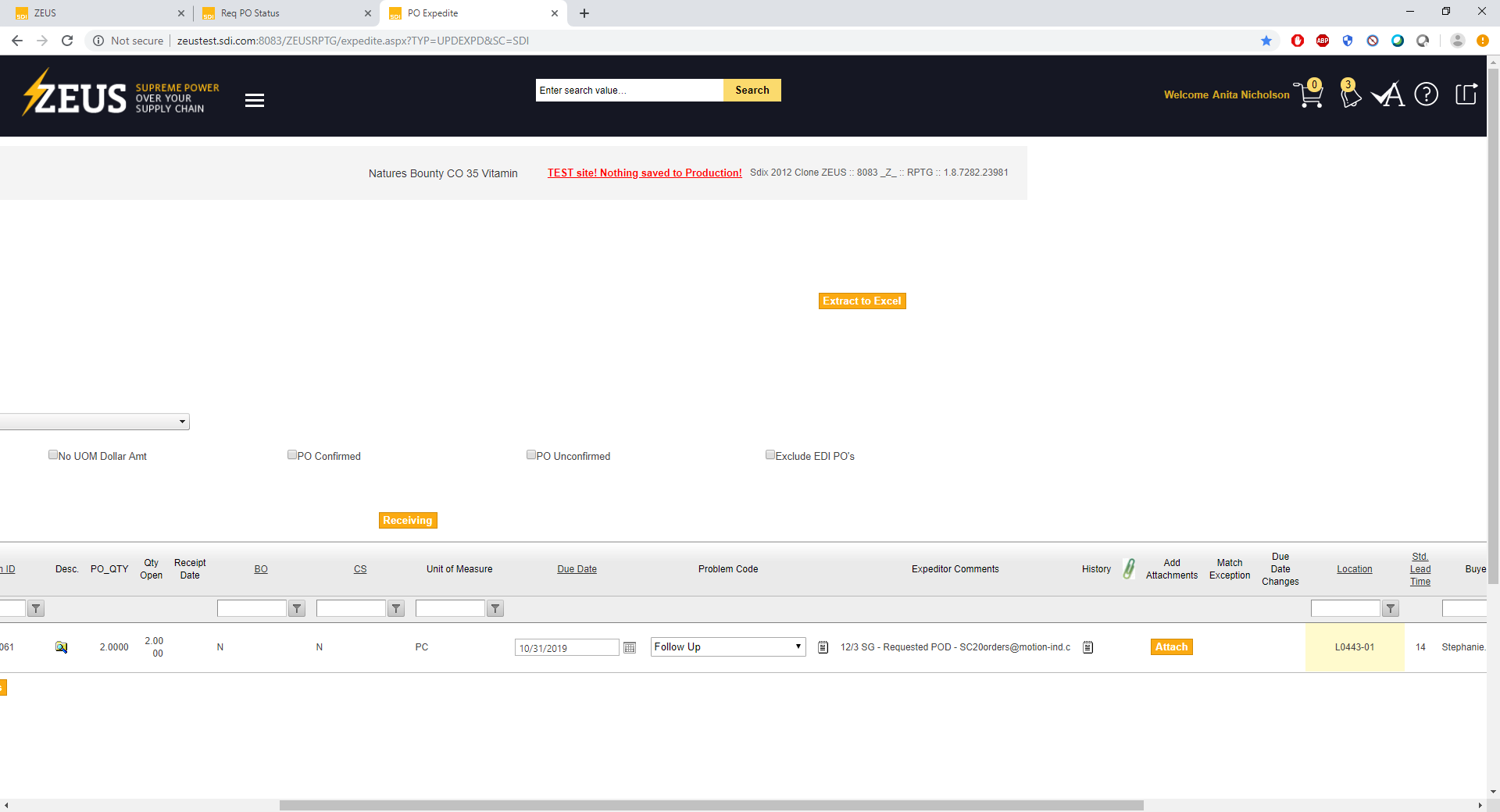
#### viewing RESULTS

Users can navigate to SDI Utilities->PO Expedite Update or to [sitedomain]/ expedite.aspx.



Enter POID and business unit. Set all other fields to empty. Click All.

Scroll to the far right in the resulting page. Click the notepad icon under history beside expeditor comments.



Note: The application might generate numerous entries in the communications table for PO IDs like R090140305 (which has over 10-15 unique tracking number entries in the UPS table). But without verifiable line numbers in the PS PO line shipment table, it will not appear in the PO Expedite Update UI.

#### When to run

UPS updates their FTP files throughout the day. Scheduled job should run at 11:45PM EST daily.

#### System Debugging

The application displays the application progress on the command line. However, it also writes the results of the matches (and parameters) to the SDIX\_UPS\_QUANTUMVIEW\_LOG. If there are any system or C# errors, the raw errors are written to the SDIX\_UPS\_QUANTUMVIEW\_ERROR table.

## Main Features

Below find key application features:

* Searches both SHIPPING and PURCHASE ORDER tables respectively: Because the tracking number is not always found in the shipping table, it also searches the relevant Purchase order tables in addition to shipping tables.
* Processes pipe-delimited and Tab-delimited files: Most of the text files on the UPS server use the pipe (|) character to separate data. However, the QVD\_ALT\_sdiinc\_20191115\_110106\_627\_SDIQVD.txt contained tab-delimited characters. It is possible a user accidentally re-saved it with tabs. But, just in case, the application handles tab-delimited files. Tab-delimited files handle white space differently when read into a C# string buffer as opposed to white space in the plain text file. As a result, for CSV files the code searches for the tracking number and purchase order id using regular expression patterns.
* Handles Multiple field Arrangements: The file fields might not appear in the same order. Additionally, not all the fields will always appear in each file. Additionally, M. Randall and other key SDI users can request UPS adjust the data included in the outputted text files. As a result, the utility does not assume each file will contain the same fields in the same order. It runs through the header of each file to identify its position. On a bonus, the field header titles never vary and always stick to camel case i.e. RecordType/ShipperNumber – Hence the reason a ToUpper search is not used.
* Pulls from 4 different fields to identify purchase order: Purchase order does not always appear in the files, in the same format or in the same fields. Because the SDI PO ID typically contains 10 characters. Older PO IDs were numeric. Newer ones typically start with 1-2 letters, followed by numbers. The PO Ids in the QuantumView tracker text files could appear in the ShipmentReferenceNumberValue fields (1&2) and/or the PackageReferenceNumberValue fields (1&2). Additionally the PO ID appearing in the UPS file might contain a dash(-), colon(😊, pound sign(#), extra spaces or extraneous letters.   
  Sample Values contained in actual files include: TX57-00075945, PM00069193, 4507699387, PO# 8044-1114, PO 0180214017, OH41-00289460, PO L04E094949, L01E094238, N22913361-00-001 PICK  
  The utility tries the values found in all 4 fields with 1-2 different variations per field (removing special characters) to find a match.
* Communication table entries under SDIX – All Com entries entered under operator id (OPRID) SDIX for troubleshooting ease.
* Backs up files to AWS Server: [\\172.31.251.161\sdixdata\ftp](file:///\\172.31.251.161\sdixdata\ftp\12_12_2019\12_5_2019\)
* if run on the same day with the same parameters, the utility deletes pre-existing files - If user FTPs files on 12092019 for 12092019, the utility will create a 12092019/12092019 folder. If the utility is re-run again on 12092019 for 12092019, the utility will pull down fresh files (and delete the earlier ones from the local). However, if users process files on 12/09/2019 from 5 days ago, the utility will not delete the earlier 12092019 and just create a folder 12092019/12042019

## Other Options

The below items are options coded into the system but would require tweaking to activate -

* pull files for a specific date. Currently, app defaults to current date. However, can activate feature to accept date at command line option
* warns users that files might not exist on server if date entered over 15-19 days old
* email users (QuantumEmailUtility – created to handle email capabilities)
* FTP files between specific date ranges
* write log files to local directory
* Option to handle differing database connections

## Suggested Future Modifications

The following items are suggested for the future to ideally speed up queries and make data responses more immediate –

* Use Stored procedures instead of database selects
* Use Quantum API instead of FTP
* Alert if files not FTPd to local (base functionality already exists, just need to activate)

## Issues

The below items exist as potential issues in the data

* None of the tracking #s in the earlier files were found in the relevant RPTG or STAR database tables (can forward on those tracking IDs for validation) \_ For instance, in [\\172.31.251.161\sdixdata\ftp\12\_10\_2019\12\_5\_2019\QVD\_ALT\_sdiinc\_20191205\_010147\_189\_SDIQVD.txt](file:///\\172.31.251.161\sdixdata\ftp\12_10_2019\12_5_2019\QVD_ALT_sdiinc_20191205_010147_189_SDIQVD.txt) - none of the tracking numbers were matched. It was primarily the POIDs.
* THere is no way to differentiate which poid entry belongs to which line number without a tracking number in the DB. Otherwise there redundant entries
* The POIDs matched in PS\_PO-HDR don’t appear in SHIPPING either
* Typically find matches more along the Poids in the PS\_PO\_HDR table as opposed to the tracking numbers
* UPS alternates the PO ID format. I currently update the code everytime I encounter a new variation, but that will require close watching of the LOG table and updating the code accordingly if a new pattern appears
* Redundant entries in COM table, currently the same POID might have multiple tracking numbers in the same file. That means that POID might have multiple entries in the comments table
* When there’s a match on POID, but there’s no corresponding line\_nbr, Sched\_nbr or oprid from the ship table, for insert into the COM table, use 0s for those files
* What happens when a POID exists and has multiple line\_nbrs in both the COM and SHIPPING TABLES? Currently I default to using the last row, but any other suggestions?
* For inserting into the ps\_isa\_xpd\_comment table, utility uses the default ‘SH’ as the problem code.
* Other Data Discrepancies – Sometimes Tracking number appears twice, both in tracking number field and the value fields but with spaces i.e. 1Z9161990348652126,1Z 916 199 03 4850 8710,,1Z 916 199 03 4850 8710,,1Z 916 199 03 4850 8710 as seen in \\172.31.251.161\sdixdata\ftp\12\_16\_2019\12\_3\_2019\QVD\_ALT\_sdiinc\_20191203\_010145\_585\_SDIQVD.txt

## Questions

* Are there any other fields you might want to pull from the files?
* Sometimes POIds have 2 values in one – initially the utility did not parse them out–

0290214800/00315055042 OR 335520065171741, 8593793254 OR AS04041816 & AS04040762,S-14341 & S-14342 – Should these be handled separately? Note: They probably are not individual Pos considering they are all numeric and the ones I searched on did not appear in PS\_PO\_HDR in the RPTG DB, but let us know.

* Testing Multiple Database Options: Will require DBAdmin to synch the PS\_PO\_HDR, PS\_ISA\_ASN\_SHIPPED and PS\_AS\_XPD\_COMMENT tables to STAR so that the QA tester, Mindy and key users can verify functionality. *Note: Talked to Rashmi about this. She suggested not synching to STAR but running on RPTG.*
* Redundant Entries*:* As the same POID might have multiple tracking numbers in the same file, POID might have multiple entries in the comments table. Should we just combine these into one row (as opposed to multiple) . For instance, POID C01G591513 had 8 unique tracking numbers in the [\\172.31.251.161\sdixdata\ftp\12\_10\_2019\12\_5\_2019\QVD\_ALT\_sdiinc\_20191205\_010147\_189\_SDIQVD.txt](file:///\\172.31.251.161\sdixdata\ftp\12_10_2019\12_5_2019\QVD_ALT_sdiinc_20191205_010147_189_SDIQVD.txt) *Talked to end user about lack of line number to create composite key with PO ID. Currently on pause*
* For emailing, do you want that batched or an email sent to client immediately? Sometimes Vendor\_Id contains vendor email directly, but what else can we use to identify client contact information ? Also, what are the SMTP server/smtp port and email credentials (logon/password) and what default sender email should utility use *End user would prefer sending instantly. As these are small batches and if run in the evening, this should be an issue. But should consider batches in the future for better system performance*
* When there’s a match on POID, but there’s no corresponding line\_nbr, Sched\_nbr or oprid from the ship table should we use something other than 0s for entry into the com table for those corresponding fields. *OPRID is operator ID, so created test id called “SDIX.’ Reviewed excel spreadsheet and DB to identify potential other way to uniquely identify entries in the UPS exports without matching tracking number in SDI DB and without provided line nbr. Talked to Mindy, IT Project Manager, M. Randall and end user. Currently on pause.*
* What happens when a POID exists and has multiple line\_nbrs in both the COM and SHIPPING TABLES? Currently I default to using the last row, but any other suggestions? See above. *The Line NBR and PO ID are composite keys, without them, there is no way to prevent redundant entries or link them to their proper invoice line item.*
* WHAT DO I USE FOR ISA\_PROBLEM\_CODE IF IT IS AN INSERT? Currently, I use record type*. Note: After reviewing vb code and db entries, noticed the initial PO\_ID entries use AK i.e. Acknowledgement. But, since this information relates to Shipping should use the ISA\_PROBLEM\_CODE ‘SH.’ Need to use the pre-existing ISA\_Problem\_codes or data will not appear in the UI comment history as they query checks the table that holds the ISA\_PROBLEM\_CODE affiliations.’*
* New: what to do when can’t distinguish between various line nbrs and sched nbrs as only have poid – i’m pulling the last See above. RIght now, don’t create dummy ids as the LINE\_NBRS correspond to actual invoice lines on the specific PO.

## Referenced Documents

* Code Documentation – See attached excel. This application contains 3 utility files for FTP, Database and string manipulations, in addition to the code
* CSV\_File\_guide\_v8 – Jan2016.pdf For more info on the formatting of the plain text Quantum View files, please see (provided by M. Randall)
* UPS Integration 1.2 and 2.2 – For a high level, first pass requirements document please see, (provided by M. Randall V. Karthikeyan, whose team created the first pass program.cs)

## Code change record log

Below find key changes to the code in the past three weeks upon discovering new data requirements or new tables for the main query.

|  |
| --- |
| New query on 11/26 based on new comments from M. Randall  Switched to using isa\_asn\_shipped for tracking number and searching ps\_po\_hdr for po\_id  12 switching to ps\_po\_line which has more data than ps\_po\_hdr |
| New Query change made on 12/16  Added join on PS\_PO\_HDR as sometimes what appears in PS\_PO\_HDR does not appear in PS\_PO\_LINE\_SHIP. While PS\_PO\_HDR is preferred table recommended by M.Randall, PS\_PO\_LINE\_SHIP contains more PO related data i.e. line nbr, etc. |
| New query on week of 12/11 based on review of NeedApprove.aspx.vb and VendorProfile.Aspx.vb  Switched to using isa\_asn\_shipped for tracking number and searching ps\_po\_hdr for po\_id  12 switching to ps\_po\_line which has more data than ps\_po\_hdr |
| Added ability to parse on tabs and pipes week of 12/5  Parsed file with tabs that generated errors. Added check to see if separator = ‘|’ appears in CSV file, if yes, use standard processing, if not, use tab delimited processing which searches for the POID/Tracking number pattern |
| New Query Week of 12/5  Checked PS\_PO\_LINE\_SHIP temporarily from PS\_PO\_HDR as PS\_PO\_LINE\_SHIPPED contained more information |
| New Query Week on 12/10  Temporarily Used PS\_APXVCHRLINE tbl until noticed the reportstatus.aspx.vb used PS\_PO\_LINE\_SHIP |
| Query Update Week of 12/11  Updated the comFromSql line – Previously it looked like the below. Updated the LINE NBR, SCHED NBR restrictions as those are typically empty on the shipping table side (as shipping table typically does not have a data match on tracking number)  //String comFromSql = “ LEFT JOIN PS\_ISA\_XPD\_COMMENT COM ON PO.PO\_ID = COM.PO\_ID AND COM.BUSINESS\_UNIT = SH.BUSINESS\_UNIT AND COM.LINE\_NBR=SH.LINE\_NBR AND COM.SCHED\_NBR=SH.SCHED\_NBR AND COM.OPRID=SH.OPRID “; |
| Week of 12/9 and 12/5 – Added then removed handling possibility of nested POID logic  Initially added nesting logic as some PackageReferenceNumberValue fields contain numerous digits separated by slashes or commas. Wanted to check to see if they were POIds. Checking them manually, none of them were. Also, running this code caused stack overflow, if recursively calling parseRow 2-3 times while it’s still parsing a current row. Might require closer debugging to tighten code logic and eliminate stack overflow possibility.  /\* // if package value ref 1 contains a field separated by ?, & or , - then split up into separate poids  if (qf.PackageReferenceNumberValue1.Contains(“,”) )  {  String[] poid = qf.PackageReferenceNumberValue1.Split(‘,’);  foreach(var po in poid)  parseRow(currentRow, qf, row, filename , po );  }  if (qf.PackageReferenceNumberValue1.Contains(“&”))  {  String[] poid = qf.PackageReferenceNumberValue1.Split(‘,’);  foreach (var po in poid)  parseRow(currentRow, qf, row, filename, po );  }  if (qf.PackageReferenceNumberValue1.Contains(“/”))  {  String[] poid = qf.PackageReferenceNumberValue1.Split(‘/’);  foreach (var po in poid)  parseRow(currentRow, qf, row, filename, po);  }\*/ |
| Updated query logic – week of 12/11 and 12/5   * Initially, data only did query on one default database connection. But, for testing purposes realized STAR was not up to date, so did query on first star then RPTG and if matches existed in STAR, then updated RPTG. Learned later on that RPTG is another dev DB just cleared out daily so could use that. Updated logic back to using only one default DB. * Also, initially SQL first checked PS\_PO\_HDR on PO\_ID and then ISA\_ASN\_SHIPPED table on tracking number to identify amount of matches made on PO ID vs Tracking number. But, after no matches found on tracking number, just reverted to using PO ID (but kept the Shipping table left join)   /\* No longer doing separate check of SH table as main table – since not finding on Track num, but will leave in left join This existed just to see which table was matched and 100% it was PS\_PO\_HDR on PO\_ID and not track # \*/  if (!dbReader.HasRows || dbReader == null)  {  dbReader = QuantumDbUtility.executeDbReader(dbConn, asnSelectSql + shFromSql + comFromSql + whereSql, dbParams);  sdix\_ups\_quantumview\_log\_params[5] = “TRUE: ISA\_ASN\_SHIPPED”;  } |
| Updated logic for inserting into COM table on 12/10  After further discussion with m. Randall on 12/10 since there is a log history, will no longer update, just insert  \* also will just use isa problem code for SH \*/ |
| Updated Insert Query to COM table to include current timestamp  Note: Query also pulled the previous OPERATOR, but in the end for any new entries will use OPRID, SDIX  ps\_isa\_xpd\_comment\_params = new String[8] { qf.business\_unit, qf.ps\_po\_id, qf.ps\_line\_nbr, qf.ps\_sched\_nbr,  qf.isa\_problem\_code , qf.ps\_notes\_1000\_new, “SDIX” /\* QuantumUtility.returnNull(dbReader[“SH\_OPRID”].ToString()) \*/,  /\*”31-DEC-1999” \*/ DateTime.Today.ToString(“dd-MMM-yy”).ToUpper() }; |
| /\*  if (!String.IsNullOrEmpty(qf.ps\_notes\_1000)) // && updateStar  {  QuantumDbUtility.executeDbUpdate(dbConn, “Update PS\_ISA\_XPD\_COMMENT SET NOTES\_1000= ‘@5’ WHERE PO\_ID=’@1’ and “ + “ BUSINESS\_UNIT=’@0’ and LINE\_NBR=@2 and SCHED\_NBR=@3 and OPRID=’@6’ “, ps\_isa\_xpd\_comment\_params);  QuantumUtility.logError(“ -- Updated PS\_ISA\_XPD\_COMMENT.NOTES\_1000, PO: “ + qf.ps\_po\_id); //indicate num of rows updated sdix\_ups\_quantumview\_log\_params[3] = “Updated PS\_ISA\_XPD\_COMMENT.NOTES\_1000-PARAMS: “ + String.Join(“,”, ps\_isa\_xpd\_comment\_params);  QuantumDbUtility.executeDbUpdate(dbConn, sdix\_ups\_quantumview\_log\_sql, sdix\_ups\_quantumview\_log\_params);  }  else \*/ |
| Defaulted to using ‘SH’ which stands for Shipping for ISA\_PROBLEM\_CODE in the COM table  Especially since a) difficult to match on line number b) this should be a new entry that will appear in comments history. Currently test data does not appear in UI without line number match, we could do a comments history on the overarching PO page if internal client and web services mgr amenable – Internal, client will verify if they want to see tracking data if can’t match to line number  //ps\_isa\_xpd\_comment\_params[4] = qf.RecordType;  //\*\*\* WHAT DO I USE FOR ISA\_PROBLEM\_CODE IF IT IS AN INSERT? What happens when there’s a match on POID=’AM00173902’ in ps\_po\_hdr,but there’s no corresponding line\_nbr, Sched\_nbr or oprid from the ship table? Use 0s? verify before inserting into notes row doesn’t already exist verify the test data appears in the UI |
| Updated log table  Create table SDIX\_UPS\_QUANTUMVIEW\_LOG (  /\* use if ORA 11 – ups\_quantumview\_id number(11,0) NOT NULL constraint pk\_ups\_quantumview\_id primary key, \*/  /\* Use if ORA12C or higher – ups\_quantumview\_id number GENERATED ALWAYS AS IDENTITY, \*/  ups\_filename varchar2(500) null,  po\_id NVARCHAR2(250) null,  isa\_asn\_track\_no VARCHAR2(30) null,  ups\_sdi\_match varchar2(45) null,  utility\_action varchar2(250) null,  ups\_file\_location varchar2(500) null,  dttm\_added TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  );  /\* Use triggers for auto increment, if Oracle version is Oracle11  CREATE SEQUENCE sdix\_seq\_ups\_quantumview\_id START WITH 0;  CREATE OR REPLACE TRIGGER sdix\_trgr\_ups\_quantumview\_id  AFTER INSERT ON SDIX\_UPS\_QUANTUMVIEW\_LOG FOR EACH ROW  BEGIN  SELECT sdix\_seq\_ups\_quantumview\_id.nextval INTO :new.ups\_quantumview\_id FROM dual;  END;  \*/  Create table SDIX\_UPS\_QUANTUMVIEW\_ERROR (  /\* use if ORA 11 – ups\_quantumview\_error\_id number(11,0) NOT NULL constraint pk\_ups\_quantumview\_id primary key, \*/  /\* Use if ORA12C or higher – ups\_quantumview\_error\_id number GENERATED ALWAYS AS IDENTITY, \*/  error varchar2(1000) null,  dttm\_added TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  ); |
| Based on the provided plain-text tracking number (typically 18-characters) and variations of the purchase order id found in the ShipmentReferenceNumber (Values 1&2) and PackageReferenceNumber (values 1 &2) fields, the utility searches the SDI database for matches. |
| Testing DB  Per conversation with Rashmi and M. Randall, suggested not to create test entries in the PS tables on STAR (or back up some of the RPTG tables to START) but to test against Reporting directly . Also RPTG has the more recent POIDs sent over from UPS |