You will need access to these ftp folders in dev/test:

compal

dsh\_apj\_bulk\_expense

subao

Testing DSH GHUB processes

The first step is an inbound Pull Request. These come in two ways 1)GUC and 2) Trading Networks.

Compal places an ftp file on their server, and we have a process that goes and picks it up. They are set up in this WWT\_LOOKUP: WWT\_EDI\_FTP\_INBOUND. This is the process that gets the files. EDI\_Common.preProcessEDI.flowServices:getFTPFiles.

However, Compal does not have a dev or test site. So we cannot do this from dev/test. For dev and test get a production file from TN and drop that file to our local ftp server. You will save the file with this naming convention WWT\_20141110214117.txt (note this corresponds to a date time but it does not have to be the current date), and drop it in this ftp folder. compal/outbound

The EDI\_Common.preProcessEDI.flowServices:getFTPFiles service will pick it up and send it to TN. Once it hits TN it will fire off the service as defined in this processing rule: [DGH Process 940 Pull Request Inbound](http://deveai7:8585/webm.apps.tasks.admin.integration.b2b.procrules.details.editor?wmp_tc=15389&wmp_rt=render&selectedTab=criteriaTab&procRuleID=50e4h1006tc9l17o00000018&returnURL=%2fmeta%2fdefault%2fwm_xt_fabricfolder%2f0000015359%3fwmp_tc%3d15421%26wmp_rt%3daction%26wmp_ta%3d%2523%257bWm_tn_procrules_searchresultsDefaultviewView.refreshSearchResults%257d%26sh.wmp_ks%3dtrue). The service is DGH\_940\_PullRequest\_Inbound.flowServices:process940.

The service will load the WWT\_DGH\_PULLREQ\_STG table. Then it will also call this pl/sql which will load the WWT\_DGH\_LINE table: apps.wwt\_dgh\_line\_processor.insertlinesfromstage

The 2nd way a pull request can be received is a Caper/GUC file. There are 2 sources set up in the caper lookup WWT\_FLAT\_FILE\_CLEANSING. DGH Manual Pull Request and DGH Wistron Pull Request. This file will get “cleaned” and then the GUC source, DGH 940 PULLREQUEST, will pick up the cleaned file. And it will call this pl/sql package, wwt\_dgh\_pull\_request\_upload.process\_file, which basically does the same thing as the webmethods service.

When the analysts do their testing they use the DGH Manual Pull Request file layout so they drop the file here: <FTP\_ROOT>dsh\_apj\_bulk\_expense/chengdu\_pull\_request

The filename for the manual pull request really doesn’t matter. But it needs to be a .csv file.

You should use that folder and use the layout that I am sending along with these instructions.

Someone from the analyst team can do a commit in the GHUB app (I am not sure how to do this since I have never actually been in the GHUB app). You could also manually reset a commit record – there is a process\_status column on the commit outbound table. The process will send out 1 or more commits (depending on the partner). Sirtron always gets a commit. Compal gets a Commit for their product, but Wistron does not get one. The outbound commit process is scheduled every minute in prod. In dev/test it can be scheduled at whatever interval you’d like or it can be run manually. DGH\_Commit\_Outbound.flowServices:transmitOutboundCommits

This will pick up any commits that are waiting to go out. WWT\_DGH\_COMMIT\_OUTBOUND is the driving table. So basically a commit is WWT saying, we commit to giving you these parts. Sirtron (which used to be known as Subao) is the partner in China that actually takes the parts over to the factories – Wistron and Compal’s sites. The commit outbound uses the broker. It sends docs to the broker for Sirtron and Compal. The DGH\_Commit\_Outbound.flowServices:compalMapping is called if a Compal doc is published to the broker. The DGH\_Commit\_Outbound.flowServices:sirtronMapping is called if a Sirtron doc is published to the broker. These services map and send the files to TN. TN will ftp the files to the appropriate directories. They are put onto our local ftp server in DEV/TEST for Compal but are actually sent to Compal in prod. Sirtron picks up the files from our local ftp server so they go to our server in dev/test/prod. These are the folders you can check for testing. Compal/inbound/ SN\_WWT\_20130313A3\_01.txt and subao/outbound/ WWT\_940\_20140708161623407.csv I have included sample file names so you know the naming convention.

Once Sirtron gets our commit they will send us an ASN. Of course in dev/test this does not really happen so you have to manually drop an ASN as if Sirtron sent one.

The ASN from Sirtron will be dropped in the subao/inbound folder. The naming is important. You can run a select statement courtesy of Chris Brooks to create the asn format (of course you may have to tweak it when you are testing). Then you just do a save as and boom you are done! Filename should match this format: SuBao\_945\_WWT120626001.csv Note that the WWTxxxx part is the asn number from the data.

The ASN will get picked up by the FTP EVENT HANDLER. This is the service. WWT\_FTP\_Common.flowServices:handleUploadCompleteEvent

That service will pick up the file and call this webmethods service DGH\_ASN\_Processing.flowServices:receiveSirtronASN. This service creates a record in the WWT\_DGH\_ASN\_LINE table. This service will also create a Sales Order record in the Orig tables.

There are some outbounds you can run to create test files:

DGH\_852\_Outbound.flowServices:processCompal852 – This goes to TN – so in dev/test it will ftp to our internal ftp server and will go in the compal/inbound folder. This is backwards because when we put the real files on Compal’s server – they are “inbounds” to Compal. So we mimicked that naming convention on our server.

DGH\_OpenSupplyOutbound.flowServices:processSirtronOpenSupply - this goes to Sirtron so it also goes onto our ftp server. It goes to TN as well and gets routed to the ftp folder. It goes to the subao/open\_supply folder.

This completes the GHUB servies.

Dell Non-GHUB services:

DSH\_846\_Outbound.flowServices:process\_846 – this creates an EDI outbound to Foxconn containing onhand quantities. The 846 is an inventory file. The process currently creates 3 different files. So if you run this flow service it will populate the outbound table and call the EDI service to generate the files and send them to TN.

The Brazil onhand quantity outbound uses sftp so these should be tested as well.

DSH\_BrazilOnHandQty.flowServices:processAPJOnhandQty

DSH\_BrazilOnHandQty.flowServices:processBrazilOnhandQty

These basically call the same flow service – passing in the correct region name. This goes to their sftp server – when I run this I try and log into their test server and see the file out in their directory. All of the data is in the package constants – server, user, password. Sometimes though the file disappears so fast you can’t even tell that it hit the server. But it is also archived, so I also look there to see if there is a file archived.

DSH\_Onhand\_Quantity.flowServices:processOnhandQty – this is an outbound to Dell – the Onhand to SCP is how the analysts refer to it. It was not written by me, but I have tested it a lot. It also uses sftp. The server, user, password is the same as the onhand to Brazil. In fact the files are even named the same way… so do not run these 2 at exactly the same time or one will overwrite the other – since the date timestamp is in the filename.

DSH\_GCFI\_Onhand\_Quantity – this process is a little strange to test. In production it is called by the Allocation process that Steve Karst wrote. That concurrent program is WWT DSH GCFI Item Allocation Process. The pl/sql is wwt\_dsh\_gcfi\_item\_alloc\_pkg.main. But this is a complex way to test this process because there is a lookup that this process looks at to see if it is time to send an onhand file or not. So for this purpose you can just run the webmethods services to create the onhand file for each region. These file will be ftp’d to a Dell server. There is also a copy of each ODM’s data sent to them.

DSH\_GCFI\_Onhand\_Quantity.flowServices:processOnhandQtyAPJ

DSH\_GCFI\_Onhand\_Quantity.flowServices:processOnhandQtyDAO

DSH\_GCFI\_PlanOrder\_Outbound – I do not know this process at all.

DSH\_ASN – I don’t really know this process at all. It is the trailer asn process. It goes hand in hand with the MRN process. DSH\_MaterialReplenishmentNotification. Talk to the analysts about testing this. They are the best resource… Ketan or Chris Brooks. One of them is an Inbound and the other is an Outbound. They are either emailed or sent via XML – and some partners use the webUpload.

DSH\_OnHandQty\_Inbound.flowServices:processOnHandQty – this is the inbound onhand file from IPSoft. It is in an xcbl format. I usually get files from prod and drop them into dev/test. But the files are huge so you can’t drop a full file in mywebmethods or it gives you an error. I will attach a sample file that you can drop into mywebmethods. It will go to TN and will create an xcbl response document to IPSoft.

DSH\_SAF\_Inbound – I don’t know this process at all, but the files come in through TN.

DSH\_Vendor\_Schedule – Brenda knows these. They are 2 outbound files that go to Austin Foam. They go to TN and then get ftp’d. Run this flow service. DSH\_Vendor\_Schedule.flowServices:getVendorSchedule

These go to our ftp server and Austin foam logs into their account and picks them up. Here is their info in dev/test/prod.

User id: austinfoam

Password: sa7waWRU

---------------------------------------------------------------------------------------------------------------------------------------

DSH\_ODM\_Invoice – This process is complicated. There are 2 options that the ODMs use for sending us these invoices. Webupload and EDIINT through TN. So you will need to test both paths. For webupload - you will need to log into the webupload. If you do not have access – you need to get added to some group in the user approval tool – I am not sure what that group is – the analysts have access to the tool and can help you get access. Once you have access to log into the web upload (and you will need this in dev/test separately I think) then you also need to be added to the lookups.

Here is a link for the dev web upload: <http://www-dev.wwt.com/uploadCenter/begin.do>

For test just replace the –dev with a –test.

There are lookups behind the webupload: just add yourself to the lookup and you will have the option to pick ODM invoice partners and the DSH ODM Invoice documents.

The trading partner you select does not have to match what you are dropping in the file – in reality the partners have links that we have shared with them, so they don’t get an option to choose a partner. The code does no validation on the partner versus the filename or data. So just pick any DSH ODM Invoice partner. Then the document type is DSH\_ODM\_INVOICES. Then you go find your file and select it and click the upload button. The screen will render the contents of the file you dropped with any errors, successes.

Here are the lookups you need to know about:

WWT\_TP\_DOC\_LINK – this links the partner to the document type. You should not need to modify this.

WWT\_UPLOAD\_USERS – add yourself to this one. Look at my id CROSSENS as an example. You need to add your name along with the partner ids. As long as you pick some that are ODM Invoice partners you are good to go.

WWT\_WEB\_UPLOAD\_DOCUMENTS – this contains the document types that are used by the web upload.

WWT\_INTEGRATION\_PARTNERS – this lookup has the partners and their ids and their profile type. These ids correspond to the ids found in the WWT\_TP\_DOC\_LINK lookup from above.

The other way ODM invoices come in is via TN. In order to test those, just grab a document from prod TN and modify the partner ids in the data and drop it in mywebmethods. The document type is Dell\_XML\_810

There are 2 tags in the data: <SENDID>PEGATRON</SENDID> <RECEID>614948396</RECEID>

You will need to add a T on the end of WWT’s receiver id. The sender id should be the same for the partners in dev/test/prod…. But I am not 100% sure. .. so if you get an unknown sender it may need to be changed as well.

These invoices go into our common inbound invoice table. WWT\_INV\_HEADER and WWT\_INV\_LINE

---------------------------------------------------------------------------

Here is an upload that will test the GUC process. You can go to UNIX or you can just map this folder on your windows machine and install the free program 7 zip. Be careful where you download this from so you don’t get spyware. I think I got it here: http://www.7-zip.org/

Map this server link to a free drive letter [\\wwtntapvif1b.wwt.com\wwt\_data](file:///\\wwtntapvif1b.wwt.com\wwt_data)... Then go to that drive letter and you can maneuver the paths and get to any of the upload files.

Y:\catrepos\prod\archive\upload\gcfi\_odm\_bklg

These are zipped. You can unzip one or more using the7 zip file and drop them to this location.

A good caper source to test: DGH DELL BACKLOG APJ

Here is where the files are archived: wwt\_data\caper\prod\DGH\_DELL\_BACKLOG\_APJ

Again, you can map that wwt\_data fileshare to your local computer and you don’t even have to go to the linux server to get the files. Otherwise you can log into the prod server to get recent files and drop them into this folder: dsh\_apj\_bulk\_expense/backlog

Caper will clean the file and write it to the dsh\_apj\_bulk\_expense/backlog/cleaned folder. Then GUC will pick it up from there. This is the GUC source DGH DELL BACKLOG. Source id 149. GUC will call this pl/sql APPS.wwt\_dgh\_be\_backlog\_pkg.process\_be\_backlog. It will load this table. wwt\_dgh\_be\_backlog