DHL Invoicing

The invoicing is transactional so requires data extracted from each system.

DHL Invoicing - CTMS

Each server has a crontab entry to run the processing on the 1st of the month. It runs at 0421.

```
#Entry for DHL monthly invoicing

21 4 1 * * /oraapp/util/sql/dhl_invoicing >> /tmp/dhl_invoicing
```

The script calls a database process called DP_DHL_INVOICING.RUN_ME

Using the oratab entries (active systems) the script will run against each live database.

The package code is in CVS and should be maintained through the standard code change procedures.

A system parameter called DHL_INV_EMAIL which contains the list of email addresses to send the output to. Multiple entries should be separated with a semi-colon.

The processing runs slightly different iterations of the queries depending on the system. The queries were split into groups based on the agreed costing models.

There is also a listing of users.

Users:

- List of users who logged in in the last month.

Group 1:

- Trips - count of non-deleted trips in the previous month
- Bookings - count of schedule bookings in the previous month
- Scheduled Orders - count of orders' Load activities in the previous month.

Group 2:

- Scheduled Orders - count of orders' Load activities in the previous month, split down by cost centre and planning group (depot).

Databases and Groups:

Database	Processes Run
aam	Users, Group 1, Group 2
bnl	Users, Group 1, Group 2
con	Users, Group 1, Group 2
dun	Users, Group 1, Capped Orders, Group 2
eur	Users, Group 1, Group 2
hcr	Users, Group 1, Group 2, Scheduled Low Volume Orders, Scheduled Standard Orders
ind	Users, Group 1, Group 2

💡 **Note:**

- dun - includes a list of capped orders (order count is a minimum of 30 orders per trip, or the count of orders per trip, whichever is the larger).
- hcr - includes a split of low-volume orders (volume <= 0.20) and standard orders (any order not in that list above).

Each database will send an email with an attachment which contains tab separated data. It can be pasted directly into Excel.

DHL Invoicing - LOTS

There is a stored procedure in the MySQL database called dhl_invoicing that contains the SQL.

A bat file D:\LOTS\Invoicing\invoicing.bat runs the stored procedure and emails the file created. The file is then deleted (MySQL cannot overwrite a file).

The list of email addresses is hard coded but it is just a text file that can be edited as required.

The bat file runs from the Windows task scheduler on the first of the month at 0421

Contents

1 Process - Creating DevOps for an Opportunity

The intention of this process is to standardise the creation of DevOps jobs associated to an opportunity, for PS and R&D to work in a standardized way.

1.1 Process

1. New case is logged on salesforce and a devops job created - This will have a general heading for the project and will become the Parent case. All requests for estimate, spec, development, release requests(if they come through here), bugs/fixes will be added to this 1 case on here.

2. All cases from Salesforce will be assigned to the swimlane projects.

a. If it is a user story then after analysis a child job will be created and assigned to a member of the R&D team for estimate, spec, development / fix, test, release. The Parent case will always be left on this swimlane.

b. The case assigned to a member of the R&D team for fix to be done.

3. The R&D team will work on the case assigned to them and reassign back to the tester.

Bug fixes will have a comment will be added using **@boomi_integration** so the customer can then be updated or specs/estimates issued.

If estimate / Spec has been produced a comment will be added using **@boomi_integration** so the specs/estimates can be issued to the customer. The case will then be added to the specs or estimates pending swimlane. Once approved a case will be added to the case for this to be scheduled for development.

4. Owner of the salesforce case will update customer and send estimates and quotes on the salesforce case only.

5. The child cases will be closed once software is released to test, the Parent case will not be closed until all the jobs and UAT have been done.

With the above this will give us full visibility of what is happening from both salesforce and devops.

2 Process - DevOps Testing Process

The intention of this process is to confirm the process that all should follow so that the reporting of rework reasons can be followed.

2.1 Finishing Development

R&D

- update Resolution Notes.

- assigns case to Development Manager.
- Set Sub state to *SCR Done*

Dev Manager

- Assign to release manager to be released

Release Manager

- release to 2ndary test environment.
- Assign to Dev Manager

Dev manager

- Assign to Tester.

- Set Sub-state to *On Test*.

2.2 Testing

Testing shall produce a POT/PO2T document in the shared directory \\DGA1FS01OBS\Test_Logs\{System}\{Customer}\{DevOps}.

The document shall be placed in a sub-folder named matching the description of the DevOps case being tested.

The document shall be named POT/PO2T followed by the same matching name of the case being tested.

The format is variable, but the appropriate template should be followed.

Template: File:POT-SFNumber Description v1.1.dotx

Tester:

- Set Sub-state to *Being Tested*

2.3 Passing Testing

- Set Sub state to *Ready for Rel*.
- Update Notes.
- Link in POT document
- Assign to Dev Manager

2.4 Failing Testing

What to do to the original case

- Set Sub state to *Test Failed*.
- Raise a new DevOps **Bug**.
- Update comments.
- Link in POT document
- Assign to Dev Manager

Note: When multiple bugs are found, raise a DevOps case for each bug. Bugs can be consolidated together t=if this makes sense and they can be done together.

2.5 Raising DevOps Bugs

Every error in testing shall be raised as a separate DevOps case:

- The Name shall describe the system, customer/DB and short description of the fault.
- The Description should be described and, if necessary, placed in a document and attached or linked to in the description.
    - ♦ Describe in detail or
    - ♦ Upload a DOCX file or
    - ♦ Add a link to the shared POT document in the shared filesystem.
- Any tags being used by the customer/R&D should be set
- Area - should be set to the system and product.
- Iteration - to the latest iteration for the R&D group.
- The Repro Steps should be noted in detail or linked to the document uploaded/linked.
- The customer must be set
- Priority - the priority of the fix required. Set as per the standards. See table below.
- Severity - the severity of the bug being raised. See table below.
- Bug Type - root cause analysis - see table below.
- Fault - root cause analysis - see table below. Where there are multiple bugs consolidated into one, choose the worst fault.
- Sub State - use the list of values, values defined below
- The parent of this case shall be the User Story being tested.
- Assigned to the development manager, Karthik or Carl.

2.6 Priority, Severity, Bug Type and Fault Cross-reference

Priority and Usage

Priority	Description
1	Must be completed now/next
2	Must be completed before the next release
3	Must be completed for final release
4	Nice to have, not essential to complete

Severity and Usage

Priority	Description
1 - Critical	Showstopper.
2 - High	May have a poor workaround. Must be completed
3 - Medium	Impacting the customer, but with workaround
4 - Low	Nice to have, UI issues.

Bug Type Usage

Bug Type	Description
Configuration	
Customization	
Deployment	
Documentation	
Enhancement	
Environment	Has the bug broken or been caused by an environment issue, such as interfacing, filesystem, etc?
Functional	If there is a function simply not working, choose this option
Invalid	
Legacy	If whilst testing functionality, you note that an existing function not related to the development has been broken, choose this.
Localization	
Missed Requirement	If you note that the specification is explicitly missing the requirement, choose this one.
Performance	Functionality works but is unacceptably poor.

Bug Type	Description
Regression	If whilst testing functionality, you note that an existing function related to the development has been broken, choose this.
Security	
Setup	
Third Party	
Usability	
User Interface	Functionality works but implementation of the UI is poor.

Fault Usage

Fault	Description
Advice Provided to User	
Datafix	
Deploymemt	If a DevOps bug has been raised by R&D SOLELY for release of product changes, then the bug type should be set to this.
Design Constraint	Bug is as designed, but requirement is not fully met
Design Error	Bug is as designed, but not fulfilling requirement (FS Fault)
Development	Development issue
External Interface	Caused by external system
Hardware/Comms error	
Implementation	Caused by Implementation failure
Missing Function	Function completely missing.
No Fault Found	
Program Error	New bug
Program Error (Rework)	Bug directly with new work.
Revised Requirement	FS change
Unable to Replicate	

Sub-states and their usage

Sub-state	Who	Description	Action
For Investigation	Tester/PS	Set initially when a bug is raised.	Assign to Dev Mgr
Investigating	R&D	When investigation begins	Assign to R&D
Being Programmed	R&D	Optional - if the development/fix is multiple days.	
SCR Done	R&D	Work complete. Assign back to development manager	Assign to Dev Mgr
On Test	Dev Mgr	When assigned to tester.	Assign to Tester
Being Tested	Tester	When being tested	
Test Fail	Tester	If the issue has failed.	Assign to Dev Mgr
Ready for Rel	Dev Mgr	When the code release is ready	Assign to Rel Mgr
Installed on TST	Rel Mgr	When the update is released to test	Update Hotfix in Release to patch/release version
Installed on PRD	Rel Mgr	When the update is released to production	Update Hotfix in Release to patch/release version
On-hold	Anyone	When the case is put on hold	**USE WITH CARE. ENSURE ALL MANAGERS ARE NOTIFIED.**

2.7 Developer Updates

R&D

- New Bug
  - ♦ Follow similar process for Finishing Development above re:
    - ◊ Sub State
      - · Resolved Details
    - ◊ Update

        - Bug Type
        - Fault
        - Comments
    - Assign to Development Manager
- Existing DevOps Use Case (if assigned to R&D)
    - Update comments
    - Assign to Development Manager

Note: If multiple bugs are being fixed by the same release (i.e. in the same program/package), then ALL bugs must be assigned back to the Development Manager and on to the tester. Furthermore, one of these bugs should be considered the master - select one. Add a "Related" link to the other bugs showing that they are related to the first DevOps bug. This will help inform the Development Manager, Release Manager and Tester as to what can be tested and fixed together.

Development Manager

- The original use case and updated bug(s) should be returned to the original tester.

2.8 Releasing

Once all associated bugs are fixed, retested and assigned back to the Development Manager, they can all be released to the Release manager for releasing to the next area (TST, PROD, etc) using the standard product-related release procedures.

See also:

- Creating Release Notes

3 Process - Reporting Rework Faults

The intention is to standardize the way that DevOps cases are documented, so that the issues list can identify rework bugs and what caused the issue.

3.1 Using

Devops/Queries.

3.2 Configuration

Create a query with the following columns:

- Title
- Assigned To
- State
- Sub state
- Priority
- Severity
- Created Date
- Tags
- Work Item Type
- Bug Type
- Fault
- Hotfix in Release

Add criteria to select the data, typically:

- Work Item Type - Any
- State - Any
- EITHER/BOTH/NEITHER - depending on requirements.
    - ♦ Tags contains - select your tags
    - ♦ Custom Customer Contains - Enter a part of the customer

If possible, use a shared query or create a shared query.

An example query is here:

- Shared Queries/DevOps Extract for Analysis

Create a DevOps Bug List spreadsheet from template.File:Client BugList Analysis.xltx

3.3 Exporting

Run the Query.

Use the triple-dot menu on top right.

Choose Export to CSV

Open CSV

Copy all data (minus headers)

Open DevOps BugList spreadsheet

Paste data into first tab (DevOps Logs).

Go to second tab (Analysis)

Click Data/Refresh All.

Save.

The Analysis tab will break down the issues in the list and summarize the data.

The analysis provided:

- Total Bugs
    - Filters:
        - By State - you can select the State from the "Row Labels" drop-down filter in the header.
    - Reporting count and percentage per state.
- Bugs by Severity
    - Filters:
        - By State - you can select the State from the "State" drop-down filter in the header.
        - By Severity - you can select the severity from the "Row Labels" drop-down filter in the header.
    - Reporting count and percentage per severity.
- Bugs by Priority
    - Filters:
        - By State - you can select the State from the "State" drop-down filter in the header.
        - By Priority - you can select the severity from the "Row Labels" drop-down filter in the header.
    - Reporting count and percentage per priority.
- Bugs by System
    - Filters:
        - By State - you can select the State from the "State" drop-down filter in the header.
        - By System (Area Path)- you can select the severity from the "Row Labels" drop-down filter in the header.
    - Reporting count and percentage per system (area path).
- Bugs by Bug Type
    - Filters:
        - By Bug Type - you can select the bug type from the "Row Labels" drop-down filter in the header.
    - Reporting count and percentage per bug type, and fault beneath that.
- Bugs by Fault
    - Filters:
        - By Fault - you can select the bug type from the "Row Labels" drop-down filter in the header.
    - Reporting count and percentage per Fault.
- Bugs by Parent
    - Filters:
        - By Parent - you can select the parent case from the "Row Labels" drop-down filter in the header.
    - Reporting count and percentage per Parent case.

If you want more information on each of the bugs that were part of that group in the analysis, double-click on the count to the right of the group - a new tab will open up showing all of the bugs that match the count (usually called "Detail1").