

**Billing Operation Knowledge**

**ICP Billing**

***Large Account XML Export (XML Spooler)***

|  |  |
| --- | --- |
| **Prepared By :** | **Hafizul Fadhli Majid** |
| **Date:** | **10 August 2015** |
| **Version** | **0.1** |

**Document History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Title | Version | State | Created by | Date |
| Large Account XML Export (XML Spooler) | 0.1 | Final | Hafizul | 10 August 2015 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

1 Introduction 4

1.1 Reference document 4

2 Overview 5

3 Bill Run Processes 6

3.1 Creating and Uploading Flist 6

3.2 Run Large Account Tagging 6

3.3 pin\_deferred\_accts 6

3.4 Billing 7

4 RUN\_BILL.SH 8

4.1 RUN\_BILL.SH 8

5 Monitoring 9

5.1 Throughput Monitoring 9

5.2 Billing Log 9

# Introduction

This document is to describe how to export Large Account XML using XML Spooler.

## Reference document

References are made based on the following documentation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Document** | **File Name** | **Ref Type** | **Version No.** |
| 1 | BRM 7.3.1 Documentation |  |  |  |

# Overview

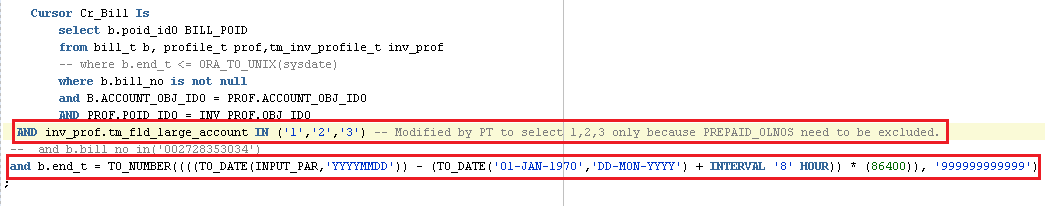
After billing/invoicing complete, Large Account XML can be exported using XML Spooler. XML Spooler are used because the number of service and usage for the invoice are big in numbers and cannot be cater by using normal export module.

# Export For Full BP

## Pre-Requisite Setting

### Stored Procedure : TM\_CUSTOM\_LA\_FLAG\_PFT

#### For default run which is exporting XMLs for whole BP batch Eg. BP01 August following setting in stored procedure must be use. Query must be set to read only the large account tag and bill date that we want to export. The procedure are located in NBRMPRD DB.



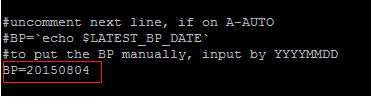
### Script : CUSTOM\_INVOICE\_EXT.sh

Next, the date to be exported need to be specified in the script to make sure that procedure above can recognized the date to be exported. The script are located in the following path:

Node 1 (10.41.66.10)

Path : /app/brm/tmbrm\_batch/InvXmlSpooler

Script : CUSTOM\_INVOICE\_EXT.sh



The format of the input date is YYYYMMDD

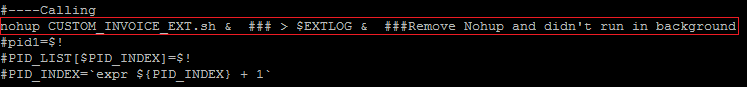
### Script : brm\_custom\_invoice\_extract.sh

This script is the main script to trigger XML Spooler. Before triggering one setting in this script need to be uncommented.

Node 1 (10.41.66.10)

Path : /app/brm/tmbrm\_batch/InvXmlSpooler

Script : brm\_custom\_invoice\_extract.sh



This part need to be left uncomment so the DB part of the script can be triggered.

### Triggering the script

This script is the main script to trigger XML Spooler. The script can be triggered

Node 1 (10.41.66.10)

Path : /app/brm/tmbrm\_batch/InvXmlSpooler

Script : brm\_custom\_invoice\_extract.sh

Command : nohup brm\_custom\_invoice\_extract.sh &

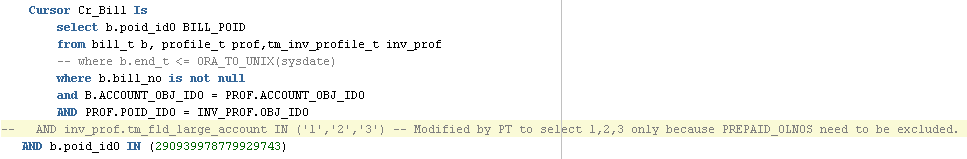
# Partial Export

## Pre-Requisite Setting

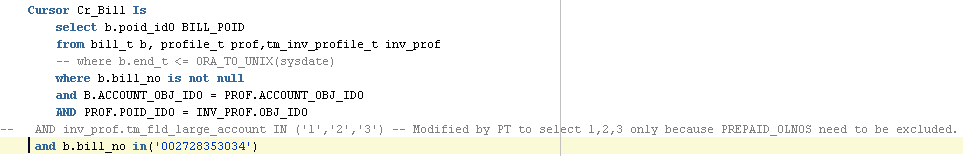
### Stored Procedure : TM\_CUSTOM\_LA\_FLAG\_PFT

#### Sometime for recon or re-export existing bill, we will be using bill number or bill poid to export a specific bill. This can be set in the stored procedure itself so only wanted bill will be produce.

By Using Bill Poid



By Using Bill Number



### Triggering the script

This script is the main script to trigger XML Spooler. The script can be triggered

Node 1 (10.41.66.10)

Path : /app/brm/tmbrm\_batch/InvXmlSpooler

Script : brm\_custom\_invoice\_extract.sh

Command : nohup brm\_custom\_invoice\_extract.sh &

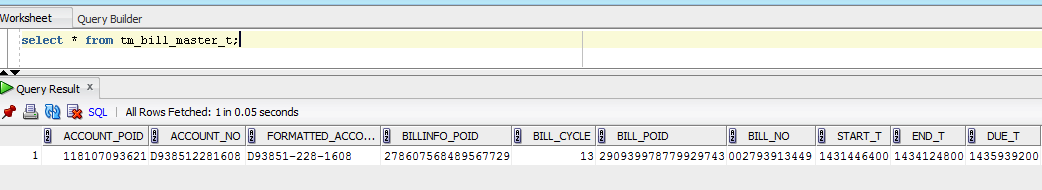
No changes needed for the script. Just triggered the script after compile and save the stored procedure.

# Troubleshooting

## Process Flow

### Master Table

When XML Spooler script triggered, all expected bill will be inserted into master table which is tm\_bill\_master\_t in NBRMPRD. Before date being inserted, the existing table content will be cleared, which mean the table will be truncated each time spooler script triggered.



This table will contain all the records for bill that will be exported including the amount total, amount due, tax amount etc. There one column need to be monitored which is STATUS. After the database part completed the STATUS will change to ‘EC’ so that exporting part will continue.

### XML Spooler

After DB part completed, the exporting part will triggered which will begin to spool the data into XML. All these data will pull out from DB by using java program. XMLs produce will be in the following path :

SLA bill : /app/shared/tmbrm\_batch/InvXmlSpooler/output/input/SLA

LLA bill : /app/shared/tmbrm\_batch/InvXmlSpooler/output/input/LLA

### CMC Checker

After export complete, CMC checker will run to check the CMC issue with the bill. The process is the same with NORMAL account CMC checker but the XML input path will be taken from the above output path.

## Rerunning The Script

In case of the script need to be killed halfway for some issue, before retriggering, all temp table created from the previous run need to be dropped to make way for next run to be successfully executed. All the temporary table can be dropped by using the following SQL command.

drop table TM\_NON\_USG\_CHARGES\_T\_1;

drop table TM\_NON\_USG\_CHARGES\_T\_2;

drop table TM\_NON\_USG\_CHARGES\_T\_3;

drop table TM\_NON\_USG\_CHARGES\_T\_4;

drop table TM\_NON\_USG\_CHARGES\_T\_5;

drop table TM\_NON\_USG\_CHARGES\_T\_6;

drop table TM\_NON\_USG\_CHARGES\_T\_7;

drop table TM\_NON\_USG\_CHARGES\_T\_8;

drop table TM\_NON\_USG\_CHARGES\_T\_9;

drop table TM\_NON\_USG\_CHARGES\_T\_10;

drop table TM\_USG\_CHARGES\_T\_1;

drop table TM\_USG\_CHARGES\_T\_10;

drop table TM\_USG\_CHARGES\_T\_2;

drop table TM\_USG\_CHARGES\_T\_3;

drop table TM\_USG\_CHARGES\_T\_4;

drop table TM\_USG\_CHARGES\_T\_5;

drop table TM\_USG\_CHARGES\_T\_6;

drop table TM\_USG\_CHARGES\_T\_7;

drop table TM\_USG\_CHARGES\_T\_8;

drop table TM\_USG\_CHARGES\_T\_9;

drop table TM\_NON\_USG\_INV\_DATA\_T\_1;

drop table TM\_NON\_USG\_INV\_DATA\_T\_10;

drop table TM\_NON\_USG\_INV\_DATA\_T\_2;

drop table TM\_NON\_USG\_INV\_DATA\_T\_3;

drop table TM\_NON\_USG\_INV\_DATA\_T\_4;

drop table TM\_NON\_USG\_INV\_DATA\_T\_5;

drop table TM\_NON\_USG\_INV\_DATA\_T\_6;

drop table TM\_NON\_USG\_INV\_DATA\_T\_7;

drop table TM\_NON\_USG\_INV\_DATA\_T\_8;

drop table TM\_NON\_USG\_INV\_DATA\_T\_9;

drop table TM\_USG\_INV\_DATA\_T\_1;

drop table TM\_USG\_INV\_DATA\_T\_10;

drop table TM\_USG\_INV\_DATA\_T\_2;

drop table TM\_USG\_INV\_DATA\_T\_3;

drop table TM\_USG\_INV\_DATA\_T\_4;

drop table TM\_USG\_INV\_DATA\_T\_5;

drop table TM\_USG\_INV\_DATA\_T\_6;

drop table TM\_USG\_INV\_DATA\_T\_7;

drop table TM\_USG\_INV\_DATA\_T\_8;

drop table TM\_USG\_INV\_DATA\_T\_9;

drop table TM\_USG\_CHARGES\_T\_TMP\_1;

drop table TM\_USG\_CHARGES\_T\_TMP\_2;

drop table TM\_USG\_CHARGES\_T\_TMP\_3;

drop table TM\_USG\_CHARGES\_T\_TMP\_4;

drop table TM\_USG\_CHARGES\_T\_TMP\_5;

drop table TM\_USG\_CHARGES\_T\_TMP\_6;

drop table TM\_USG\_CHARGES\_T\_TMP\_7;

drop table TM\_USG\_CHARGES\_T\_TMP\_8;

drop table TM\_USG\_CHARGES\_T\_TMP\_9;

drop table TM\_USG\_CHARGES\_T\_TMP\_10;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_1;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_2;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_3;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_4;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_5;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_6;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_7;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_8;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_9;

drop table TM\_NON\_USG\_CHARGES\_T\_TMP\_10;