Orbit

Orbit Interfacing to SAP
(An Example)

Purpose

- This presentation documents the implementation of an interface between Orbit and SAP
- Slides will describe
  - How the interface works
  - How Orbit works to support the interface
  - How to setup/use Orbit to drive the interface

How the Interface Works

The interface between Orbit and SAP is file based. This means communication of data between Orbit and SAP occurs through the exchange of files between the 2 systems.

On the Orbit side, the Orbit Bridge application is responsible for creating files that are sent to SAP and reading/handling files received from SAP.

Orbit Bridge runs on a dedicated server in the Pinedale offices of Walking Man Software. That same server is also running an instance of a 3rd party application called "JaSFtp10" (HiTek Software - http://www.hiteksoftware.com). JaSFtp10 is used to automate SFTP file transfers between the Orbit Bridge Server and Junipers SAP environment.

The steps in the communication process are:

Sending Files TO SAP

- Every 10 minutes an Orbit Bridge task runs which may create files for Payment Orders, Receipt Advices and GL Entries

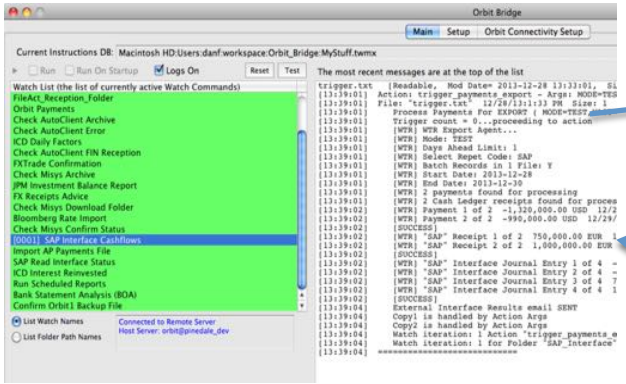
- Every 30 minutes, a JaSFtp10 schedule runs which uses SFTP to upload files from Orbit Bridge folders to designated directories that are visible to on Juniper's SAP environment

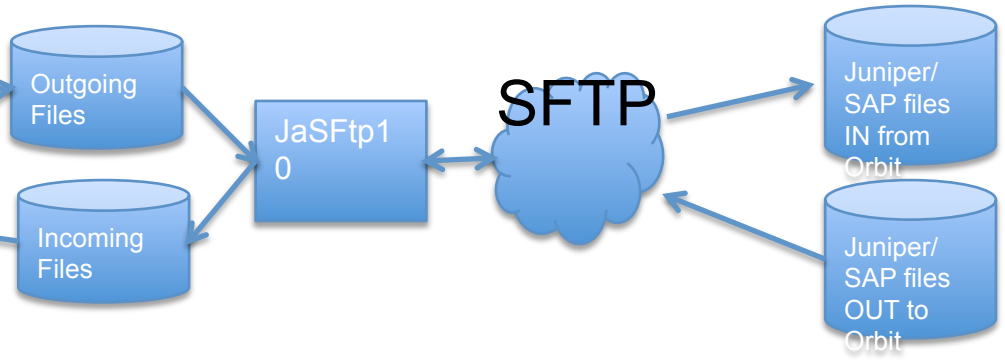
Receiving Files FROM SAP

- Every 15 minutes, a JaSFtp10 schedule runs which uses SFTP to download "status" files from directories provided by Juniper's SAP environment. Files are downloaded into folders that are visible to Orbit Bridge

- Every 10 minutes an Orbit Bridge task, processes any SAP status files that appear (placed there by JaSFtp10/SFTP)

Orbit Bridge





- Orbit Bridge is an application which runs unattended. Every 10 minutes it executes a list of tasks
- A task in Orbit Bridge is a function that either creates files in designated directories or reads files that appear in designated directories
- For the Orbit/SAP interface, there are 2 tasks that are included in Orbit Bridge's task list:
  - "SAP Interface Cashflows"
  - "SAP Read Interface Status"
- Each of these tasks is described on the next several slides

Orbit Bridge – “SAP Interface Cashflows” Task

- Task Name = “SAP Interface Cashflows”
- Orbit Bridge Action Name = “trigger_payments_export”

When the trigger_payments_export action runs

- Orbit Bridge Instantiates an instance of class WTRPaymentsBatch
- Bridge calls method WTRPaymentsBatch.checkForPayments
  - This method queries the Orbit TCash_WTR_Que table for payments where status = “APPROVED”
- Bridge calls method WTRPaymentsBatch.checkForCashLedgerPayments
  - This method queries the TCash_Ledger table for **RECEIPTS** where TxnCode.repetCode = “SAP”

- For Payments, Bridge creates a file containing one or more SAP **Payment Orders**. A single file may contain multiple payment orders. All of the payments in a single file will belong to a common batchID. BatchID is saved in the TCash_Transfers.xfrBatchID field for each payment in the batch
- For Receipts, Bridge creates a file containing one or more SAP **Receipt Advices**. A single file may contain multiple Receipt Advices. All of the advices in a single file will belong to a batch (TCash_Transfers.xfrBatchID)
- For both PAYMENTS and RECEIPTS, Orbit Bridge will create General Ledger Journal Entries in a separate file. This file will contain debit/credit entries for all of the payments and receipts processed in the current execution of the Orbit Bridge task

Orbit Bridge – Payments

- Payments
- Orbit Bridge creates a single file containing one or more SAP **Payment Orders**. A single file may contain multiple payment orders
- File Name: orbit_payments_SAP131228_14252020.txt
  - File Name Prefix: orbit_payments_SAP
  - File Creation Date: YYMMDD = 131228
  - BatchID: 14252020
- BatchID's will always be unique
- File Format
  - A payment file is formatted as an XML document
  - The document is composed of one or more <Row> elements
  - Each <Row> element specifies a single payment order. You can think of the "<Row>" element as a record
  - A <Row> element is composed of a number of different fields as shown on the next slide

Orbit Bridge – Payments

- Payments (continued)

<Row>
<Company_Code>323</Company_Code>
<House_Bank>BofA London</House_Bank>
<Account_ID>600853787019</Account_ID>
<Beneficiary></Beneficiary>
<Template_ID>123456</Template_ID>
<Value_Date>29/12/2013</Value_Date>
<Payment_Amount>-1320000.00</Payment_Amoun>
<Currency>USD</Currency>
<Single_payment>14251929</Single_payment>
<Reference_text>WIRE OUT to BARC for FX settlement</Reference_text>
<Orbit_Txn_ID>14251929</Orbit_Txn_ID>
</Row>

- <Single_payment>
  - Orbit provides SAP with Orbit Cash Ledger Items. A cash flow may be composed of one or more Cash Ledger items. In most cases there is a one-to-one relationship between a Cash Ledger item and a Cash Flow. On occasion a Cash Flow may be composed of more than one Cash Ledger entry. That is the Cash Flow might equal $10,000 and be composed of 2 Cash separate Ledger entries of $6,000 and $4,000. In this case there will be 2 <Row> elements with unique <Orbit_Txn_ID> values, but both <Row> elements will share a common <Single_payment> value
- <Orbit_Txn_ID>
  - Each <Row> element will be identified by a unique <Orbit_Txn_ID> value. This value is the TCash_Ledger.rid value and is the primary key of the item in the TCash_Ledger table

Orbit Bridge – Receipts

- Receipts

- Orbit Bridge creates a file containing one or more SAP **Receipt Advices**. A single file may contain multiple receipt advices. All of the receipts in a single file will belong to a common batch (TCash_Transfers.xfrBatchID)
- File Name: orbit_receipts_SAP131228_14252021.txt
  - File Name Prefix: orbit_receipts_SAP
  - File Creation Date: YYMMDD = 131228
  - BatchID: 14252021
- BatchID's will always be unique
- File Format
  - A receipt file is formatted as an XML document
  - The document is composed of one or more <Row> elements
  - Each <Row> element specifies a single Receipt Advice. You can think of the "<Row>" element as a record
  - A <Row> element (record) is composed of a number of different fields as shown on the next slide

Orbit Bridge – Receipts

- Receipts (continued)

<Row>
<Company_Code>320</Company_Code>
<House_Bank>BofA Amsterdam</House_Bank>
<Account_ID>600416587012</Account_ID>
<Value_Date>29/12/2013</Value_Date>
<Payment_Amount>750000.00</Payment_Amount>
<Currency>EUR</Currency>
<Single_payment>14252002</Single_payment>
<Reference_text>WIRE IN for FX settlement from BARC</Reference_tex>
<Orbit_Txn_ID>14252002</Orbit_Txn_ID>
</Row>

- <Single_payment>
  - Orbit provides SAP with Orbit Cash Ledger Items. A cash flow may be composed of one or more Cash Ledger items. In most cases there is a one-to-one relationship between a Cash Ledger item and a Cash Flow. On occasion a Cash Flow may be composed of more than one Cash Ledger entry. That is the Cash Flow might equal $10,000 and be composed of 2 Cash separate Ledger entries of $6,000 and $4,000. In this case there will be 2 <Row> elements with unique <Orbit_Txn_ID> values, but both <Row> elements will share a common <Single_payment> value
- <Orbit_Txn_ID>
  - Each <Row> element will be identified by a unique <Orbit_Txn_ID> value. This value is the TCash_Ledger.rid value and is the primary key of the item in the TCash_Ledger table

Orbit Bridge – Journal Entries

- General Ledger Journal Entries (continued)
- In the example below, tabs have been translated into <t>
- The rows painted red are Header records

```
28/12/2013<t>SA<t>100<t>29/12/2013<t>USD<t>FXR<t>FX_SETTLEMENT_DELIVERY<t>H
40<t>9308<t>990000.00<t>29/12/2013<t><t><t><t><t><t><t><t><t><t><t>L
50<t>1001<t>990000.00<t>29/12/2013<t><t><t><t><t><t><t><t><t><t><t>L
40<t>9308<t>1320000.00<t>29/12/2013<t><t><t><t><t><t><t><t><t><t><t>L
50<t>1001<t>1320000.00<t>29/12/2013<t><t><t><t><t><t><t><t><t><t><t>L
28/12/2013<t>SA<t>320<t>29/12/2013<t>EUR<t>FXR<t>FX_SETTLEMENT_RECEIPT<t>H
40<t>1001<t>750000.00<t>29/12/2013<t><t><t><t><t><t><t><t><t><t><t>L
50<t>1301<t>750000.00<t>29/12/2013<t><t><t><t><t><t><t><t><t><t><t>L
40<t>1011<t>1000000.00<t>29/12/2013<t><t><t><t><t><t><t><t><t><t><t>L
50<t>9308<t>1000000.00<t>29/12/2013<t><t><t><t><t><t><t><t><t><t><t>L
```

Orbit Bridge – Journal Entries

- General Ledger Journal Entries (continued)
- Header Record Layout

[TODAY_DMY][TAB]
SA[TAB]
[COMPANY_NUMBER][TAB]
[VALUEDATE_DMY][TAB]
[CURRENCY][TAB]
[MRKTTYPE] (FX=FXR,MMF=MMF, Investment Maturity=IMT)[TAB]
[GLTYPE] (description provided by Orbit) [TAB]
H

Orbit Bridge – Journal Entries

- General Ledger Journal Entries (continued)
- Debit Record

40[TAB]
[DEBIT_CODE][TAB]
[AMOUNT][TAB]
[VALUEDATE_DMY][TAB]
[BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB][
BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB]
L

- Credit Record

50[TAB]
[CREDIT_CODE][TAB]
[AMOUNT][TAB]
[VALUEDATE_DMY][TAB]
[BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB][
BLANK][TAB]
[BLANK][TAB]
[BLANK][TAB]
L

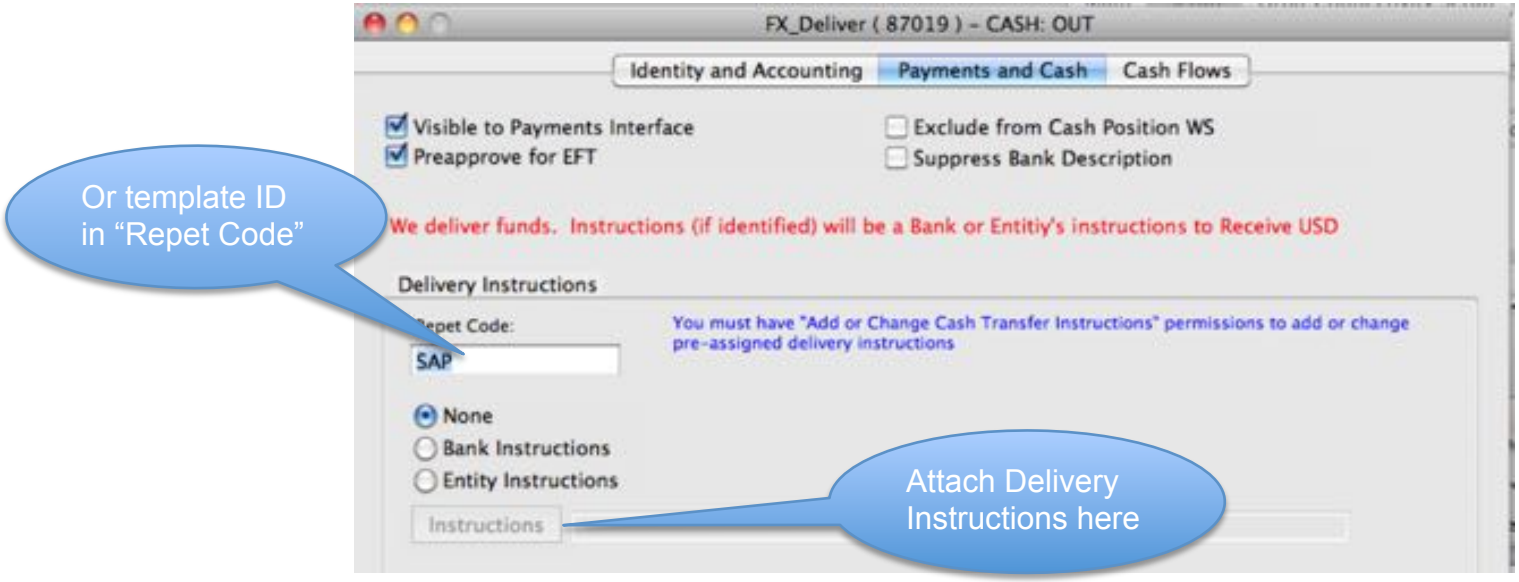
Orbit Bridge – Journal Entries

- General Ledger Journal Entries

- Orbit Bridge creates a file containing SAP **Journal Entry** lines for all of the Payments and Receipts processed in the current execution of the “SAP Interface Cashflows” task .
- File Name: orbit_gl_SAP131228_14252021.txt
  - File Name Prefix: orbit_gl_SAP
  - File Creation Date: YYMMDD = 131228
  - BatchID: 14252022
- BatchID’s will always be unique
- File Format
  - A GL file is formatted as tab delimited text file that will open in Excel
  - The document is composed of one or more lines
  - Each line defines a record. There are 2 record types: Header records and Detail records
  - Header records identify a category of entries. Header records are followed by multiple Detail records
  - Each Detail record defines is a DEBIT or a CREDIT

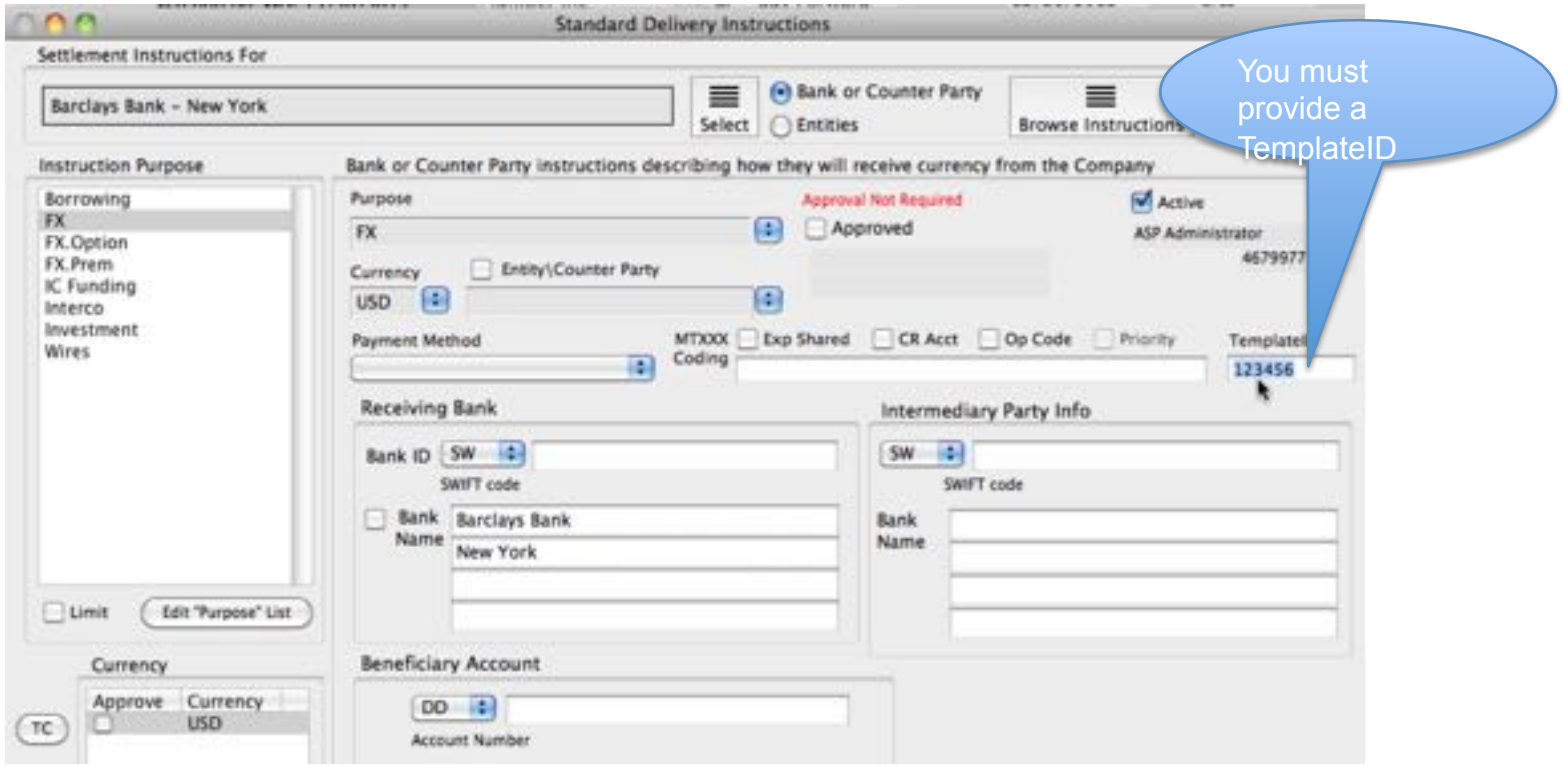
How are Payment Orders Created in Orbit ?

- A Payment Order is an Orbit “CASH OUT”, Cash Ledger entry that is visible to the payments interface
- A Cash Ledger entry is visible to the payments interface if it’s Transaction Code is setup like this:
  - The “Visible to Payments Interface” and “Preapproved for EFT” check-boxes are checked
  - For PAYMENTS, a TEMPLATEID may be coded in the “Repet Code” field
  - If a TEMPLATEID is not found in the repet code field of the Transaction Code, Orbit will look for delivery instructions attached to the Transaction Code
  - In this example, the “Repet Code” is set to SAP. This is not required for PAYMENTS, but is required for RECEIPTS. Note that RECEIPTS will never be visible to the payments interface
  - For FX payments, Orbit will normally assign delivery instructions dynamically based on the trade counter-party during the “FX Cash Settlement” process. Thus, for Transaction Codes used by FX Trade settlement cash flows, the TEMPLATEID should be entered in a delivery instruction and not hard coded in the Transaction Code



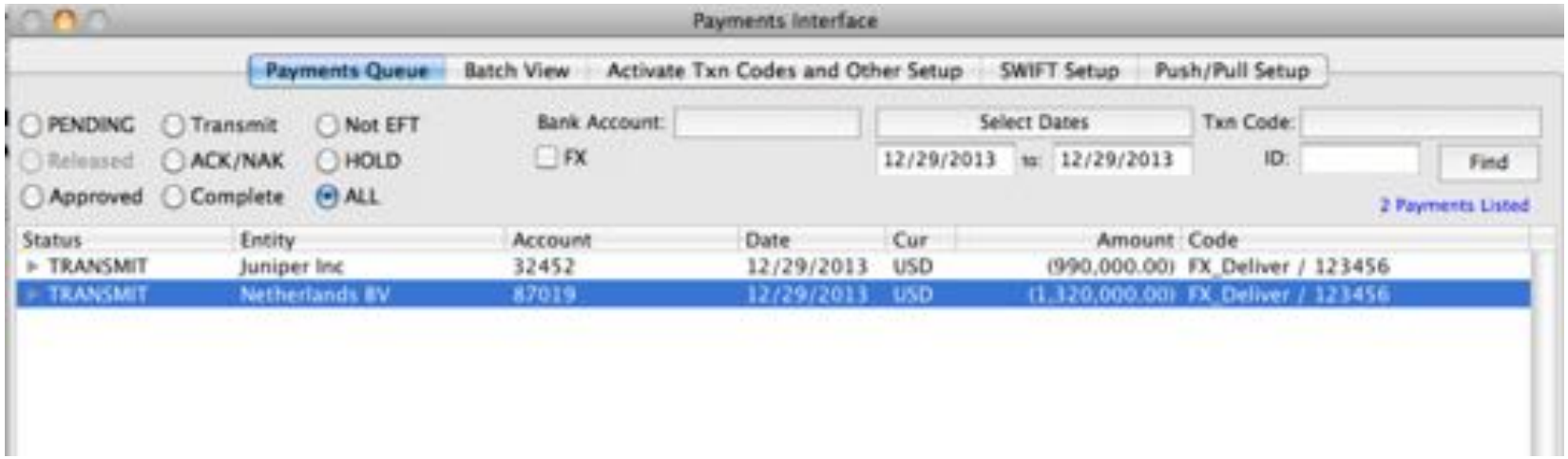
Payment Orders in Orbit – TemplateID's

- If a Cash Ledger entry Transaction Code does not have a TEMPLATEID coded in it's "Repet Code" field, the Transaction Code is must be attached to delivery instructions
- If delivery instructions are used, the instructions must include a value in the TemplateID field
- This value will be loaded into the <Template_ID>123456</Template_ID> field of a row element when the payment order is exported to SAP
- For FX settlement Payment Orders, the Transaction Codes are not hard-coded to specific instructions. This is because, the settlement cash-flow can occur with any one of our trade counter-parties. Orbit will dynamically attach delivery instructions to FX Cash Ledger entries during the FX Settlement process



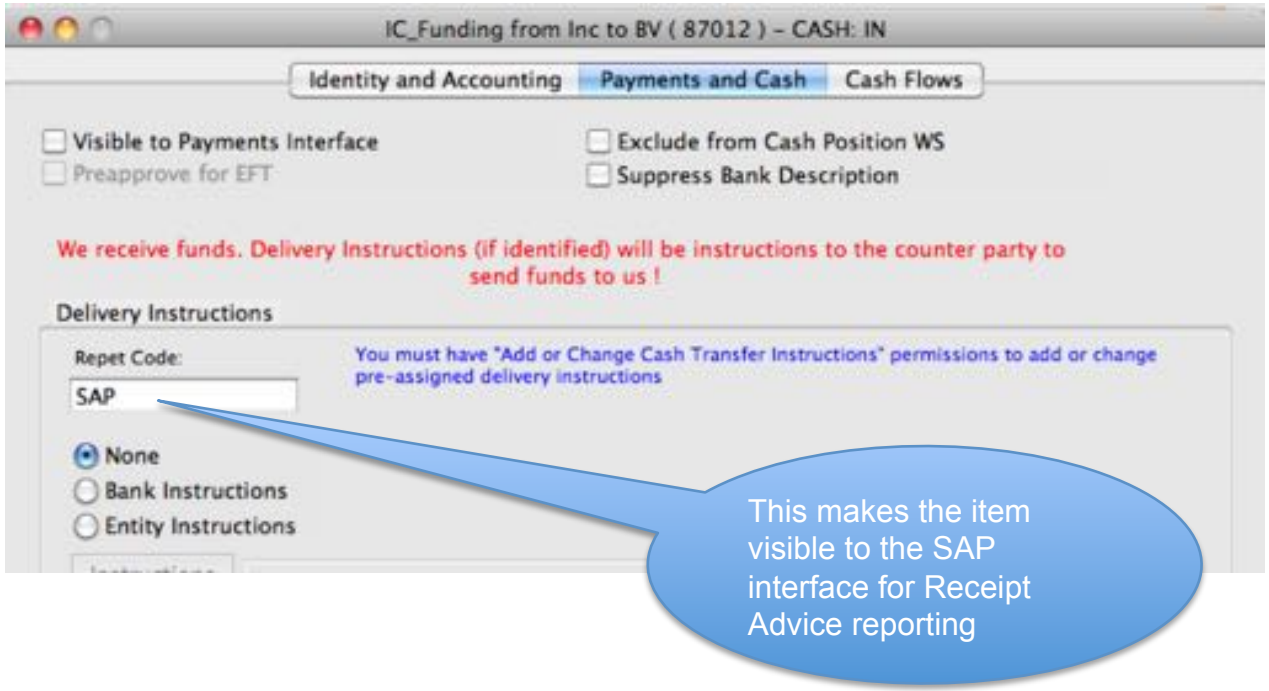
Payment Orders – Life Cycle

- When payment orders are created in Orbit, they appear in the Payments Queue
- Click on the “Payments Queue” button on the “Cash” tab of the “Main Menu Window”
- Enter a starting and ending date. Select the “ALL” radio button and click the “Find” button. Orbit will list all of the payments in the queue for the specified date range
- Status
  - The initial status will be APPROVED
  - TRANSMIT – when the payment order is picked up by Orbit Bridge and written to an SAP interface file, it’s status is changed to TRANSMIT. The Cash Ledger item can no longer be changed in Orbit, once the status changes from APPROVED
  - When SAP picks up the file and returns a status file to us, the status of the item will change in the following Order
    - SENT (received by SAP), ERROR (SAP reports an error in the payment order – most likely a bad templateID), BANK (SAP has uploaded the item to the bank), COMPLETE (bank says the payment was made)



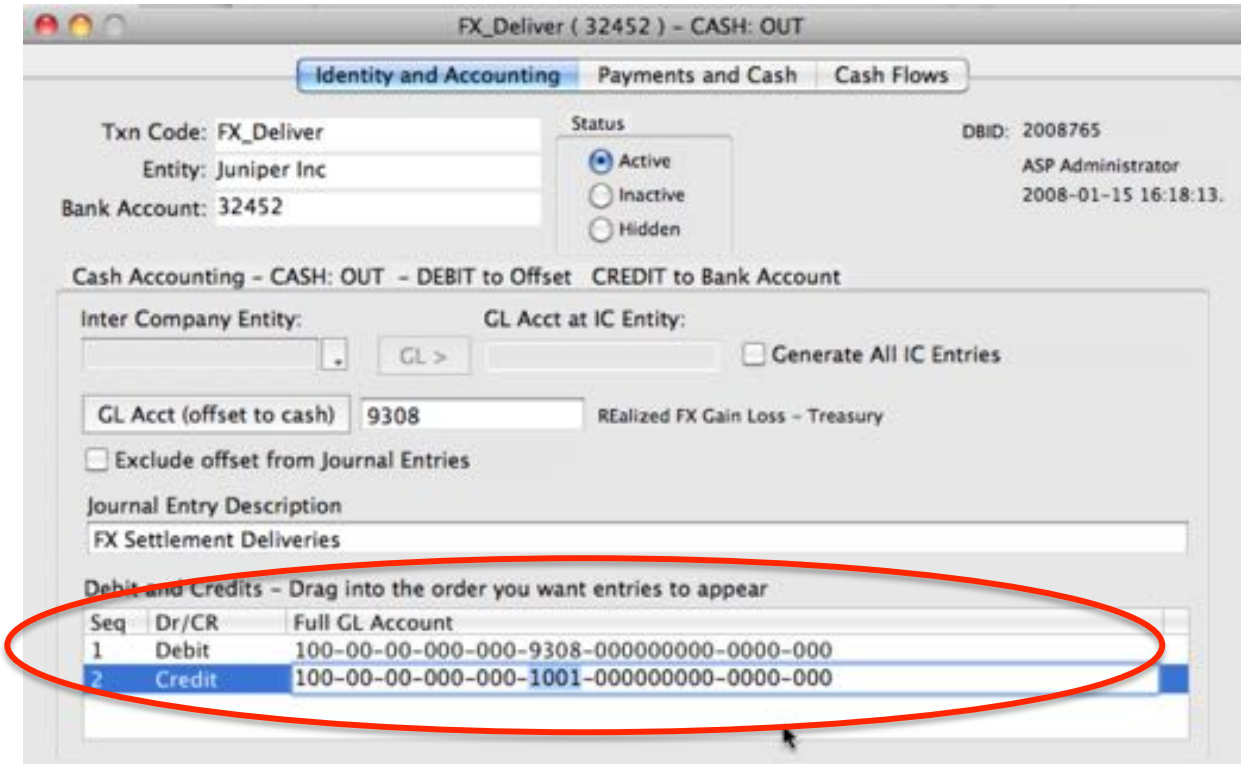
How are Receipts Created in Orbit ?

- A Receipt is an Orbit “CASH IN”, Cash Ledger entry whose Transaction code includes “SAP” in the “Repet Code” field
- C**ash Receipts**, that are to be visible to the SAP interface, **must** have “SAP” coded in the Transaction Code “Repet Code” field. CASH OUT transaction codes MAY have “SAP” coded in “Repet Code” but selection by the SAP interface for CASH OUT (payment Orders) is not dependent on this.
- “CASH IN”, Cash Ledger entries with Transaction Codes whose “Repet Code” field does not equal “SAP” will NOT BE included in Receipt Advice reporting to SAP



How Are Journal Entries Coded in Orbit ?

- The [DEBIT_CODE] and [CREDIT_CODE] values, included in General Ledger Journal Entry detail lines, come from the Cash Ledger Transaction Code
- Journal Entries are reported to SAP for all Orbit Payments and Receipts that are exported to SAP
- Each Payment and Receipt includes a Debit line and a Credit line
- Recall that all Payments and Receipts are Cash Ledger entries in Orbit
- Every Cash Ledger entry includes a Transaction Code
- Transaction Codes define the DEBIT and CREDIT GL coding that will go with the Cash Flow

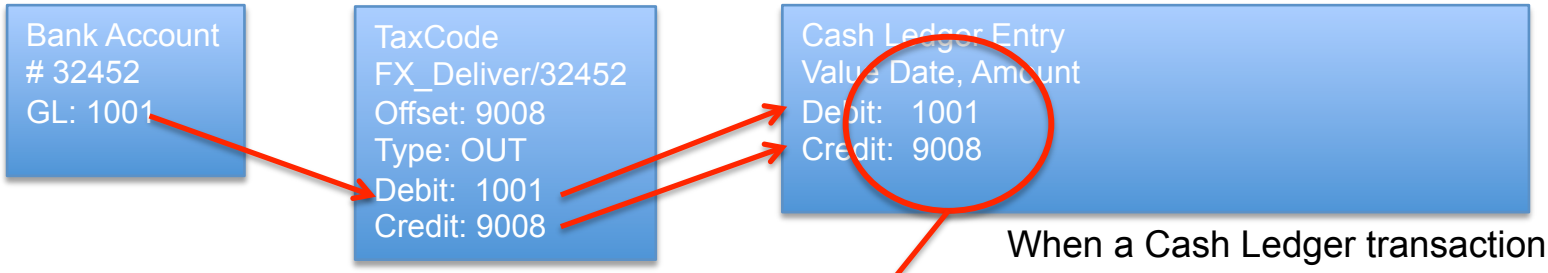


How Are Journal Entries Coded in Orbit ?

- To change the default GL Coding that goes with a Cash Flow, you change the coding in the Transaction Code that goes with the Cash Ledger entry (the Cash Ledger entry is the representation of the cash flow in Orbit)
- Transaction Codes are identified by a Code (FX_Deliver) and a Bank Account (32452)
- The Code: FX_Deliver can be combined with many different Bank Accounts. This means that the GL Coding for FX_Deliver can be different for each Bank Account
- Each Transaction Code/Bank Account combination included default coding for a DEBIT and CREDIT
- For Cash OUT (payment) Transaction Codes, the CREDIT is always to CASH and the DEBIT is to an offsetting account
- The Transaction Code knows what Bank Account it belongs to and asks the Bank Account what it's identity is in the GL. The Bank Account GL Code is then used as the CREDIT account in the Cash Flow Journal Entry
- The Offset GL Account (DEBIT) is set in the Transaction Code itself
- For Cash IN (receipt) Transaction Codes, the DEBIT is always to CASH and the CREDIT is to the offset account.

- When a Cash Ledger entry is created, Orbit automatically codes the DEBIT/CREDIT based on what is set in the Transaction Code. However, the operator may manually override that coding for individual Cash Ledger entries
- Therefore, by default, the GL Coding that goes to SAP comes from the Cash Ledger Transaction Code, but ultimately, the GL coding resides with the Cash Ledger entry itself
- The next slide is a diagram of the different parts that work together in Orbit

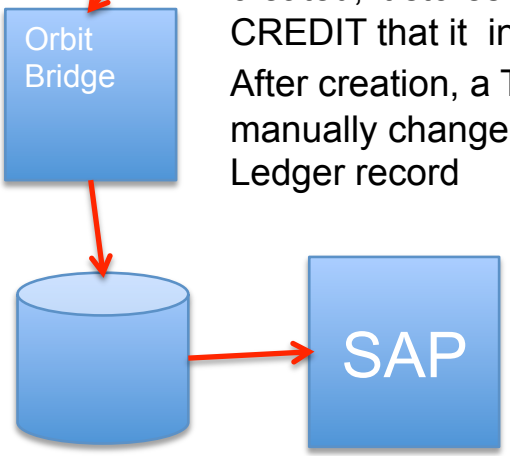
Cash Ledger Components



A TxnCode is a union between the code and a bank account. The TxnCode inherits the DEBIT from the Bank Account

When a Cash Ledger transaction is created, it stores a copy of the DEBIT/ CREDIT that it inherits from the TxnCode.

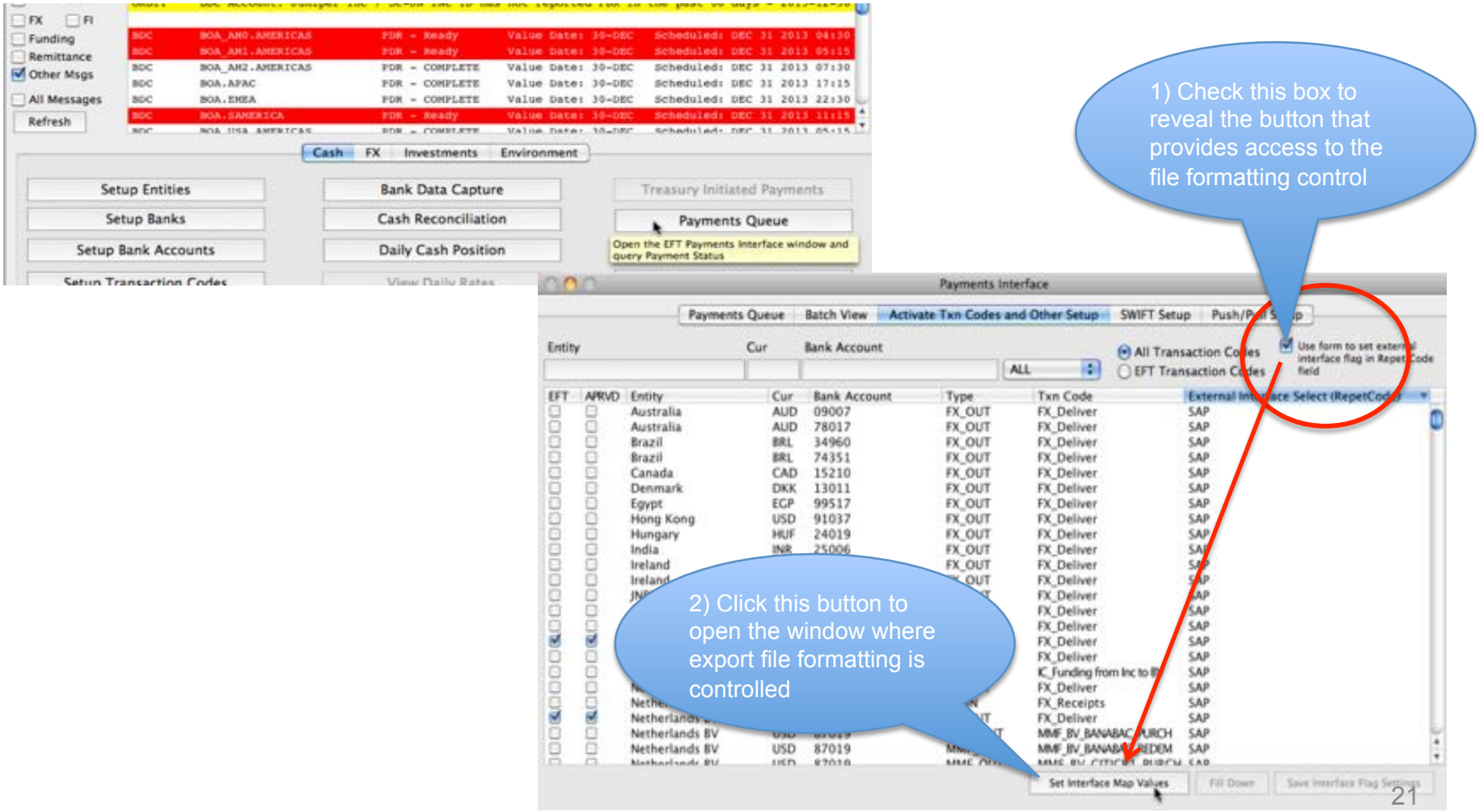
After creation, a Treasury operator can manually change these values in the Cash Ledger record



Orbit Bridge gets the DEBIT/CREDIT that it passes onto SAP from the Cash Ledger record

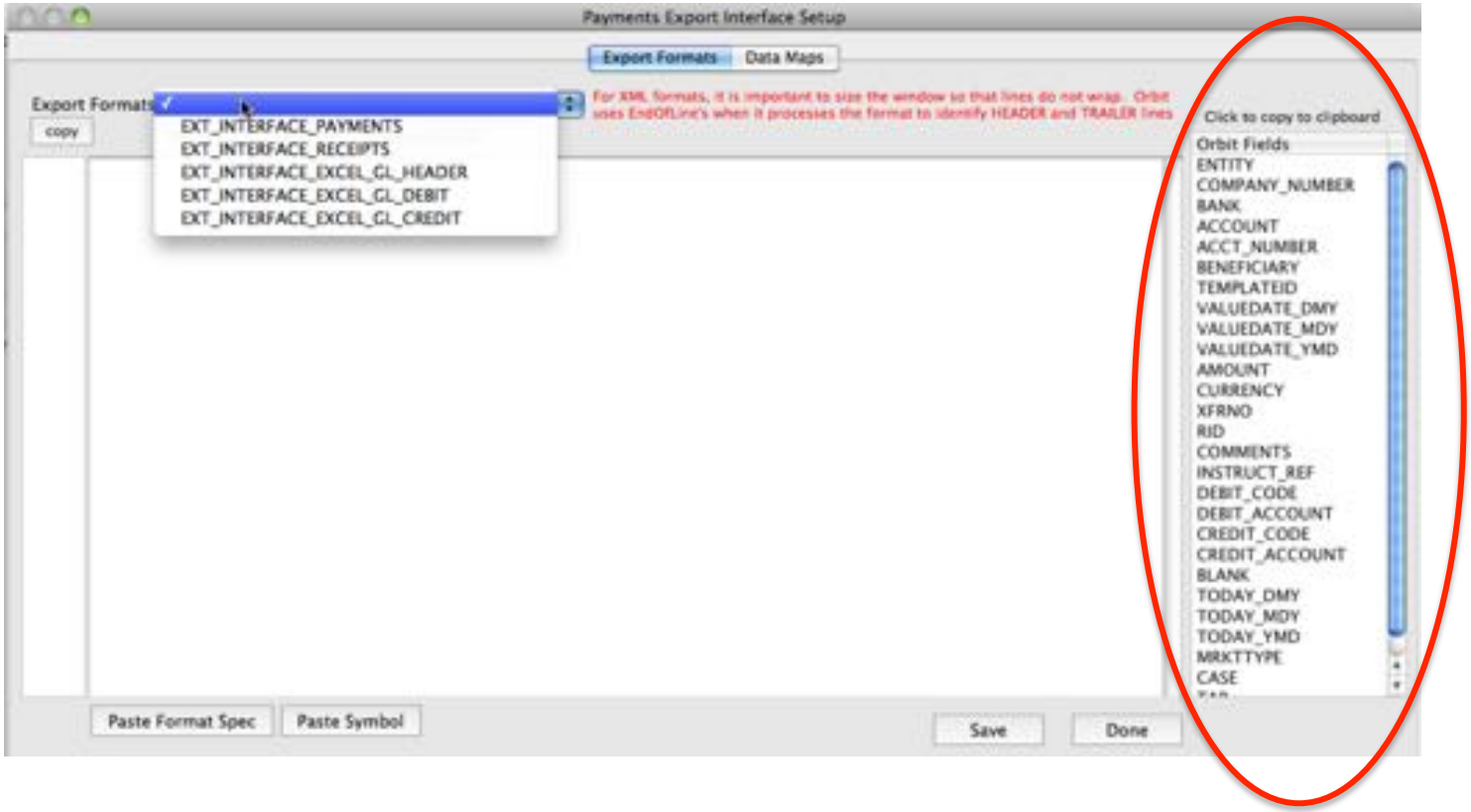
How Are Export File Formats Defined ?

- The formatting of data in orbit_payments, orbit_receipts and orbit_gl export files is controlled by screens hidden away in a corner of the “Payments Interface” window
- To get there from the “Main Menu Window”: go to the “Cash” tab. Click the “Payments Queue” button. The “Payments Interface” window opens. Click on the “Activate Txn Codes and Other Setup” tab. Follow the instructions shown in the blue bubbles



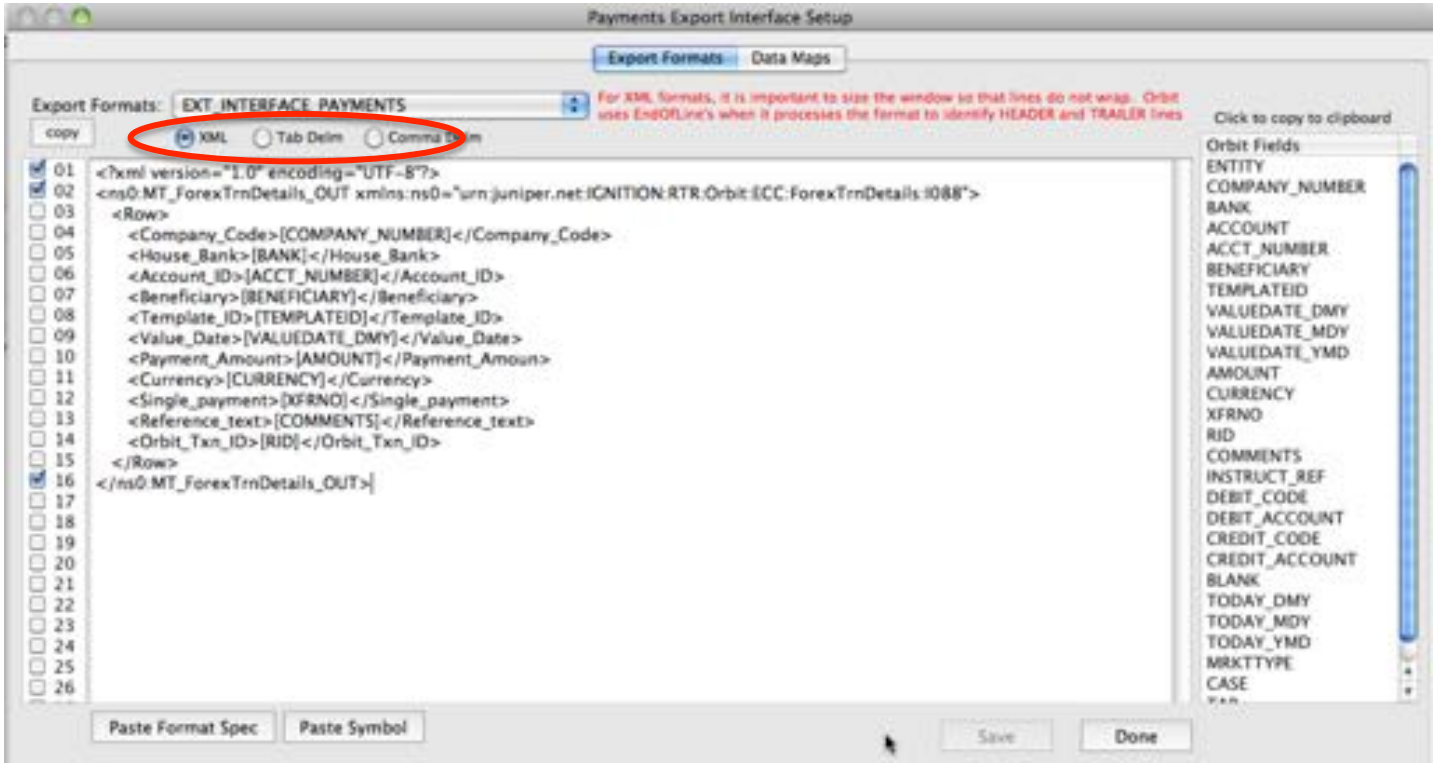
How Are Export File Formats Defined ?

- This window provides controls for defining the formatting of export files
- The drop down menu is a list of the formats
- The values on the right (circled in red) are names that Orbit understands. These names will be embedded as symbols in the format spec. When Orbit Bridge creates export data, it replaces the symbols with values for the names it understands
- This approach allows us to make modest changes to export formats without having to reprogram Orbit Bridge



How Are Export File Formats Defined ?

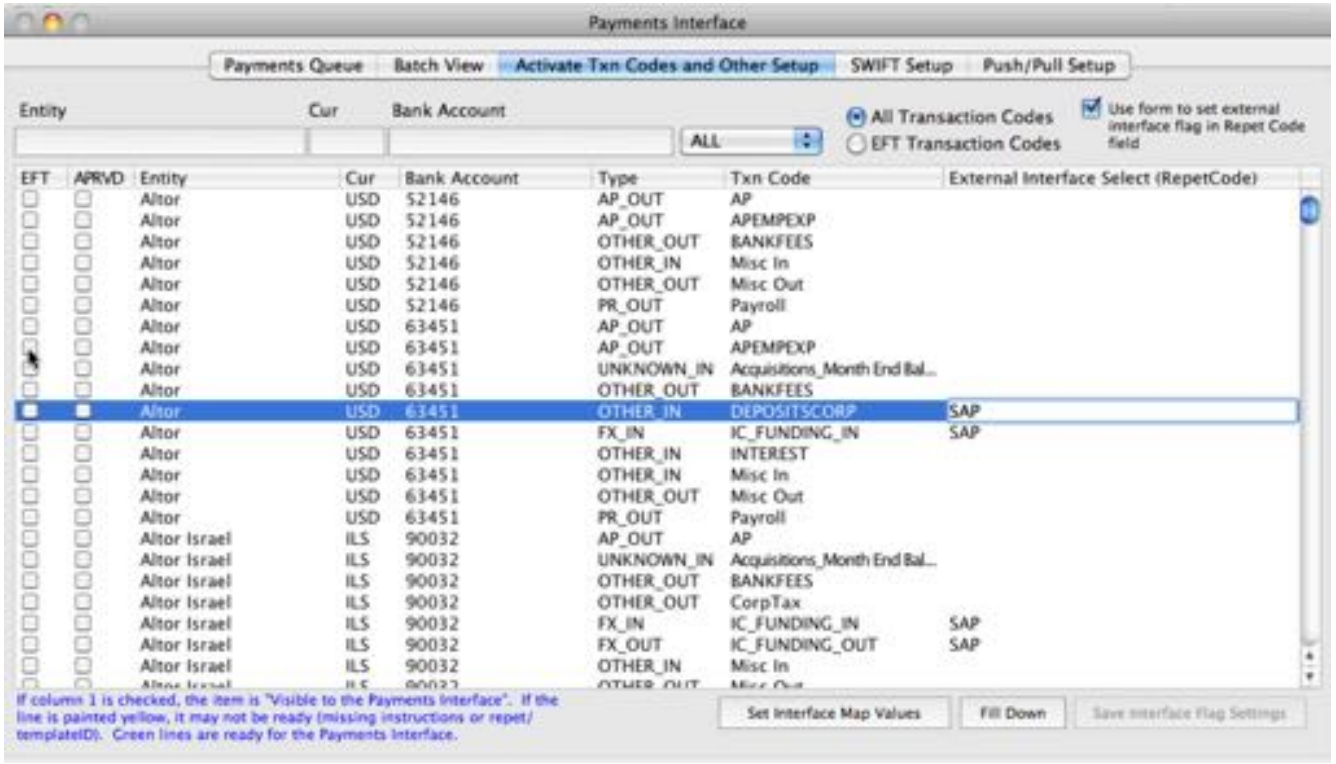
- Here is an example of the format spec for the Payments file. You can see that it defines XML with symbols embedded ([BANK], [ACCT_NUMBER] etc.), that Bridge will convert into values at runtime
- For XML, we also tell Orbit where the XML header/trailer lines are located
- It's also important to properly set the radio buttons that tell Orbit what the basic output is (XML, Tab, Comma)



Getting Orbit Setup for the SAP Interface

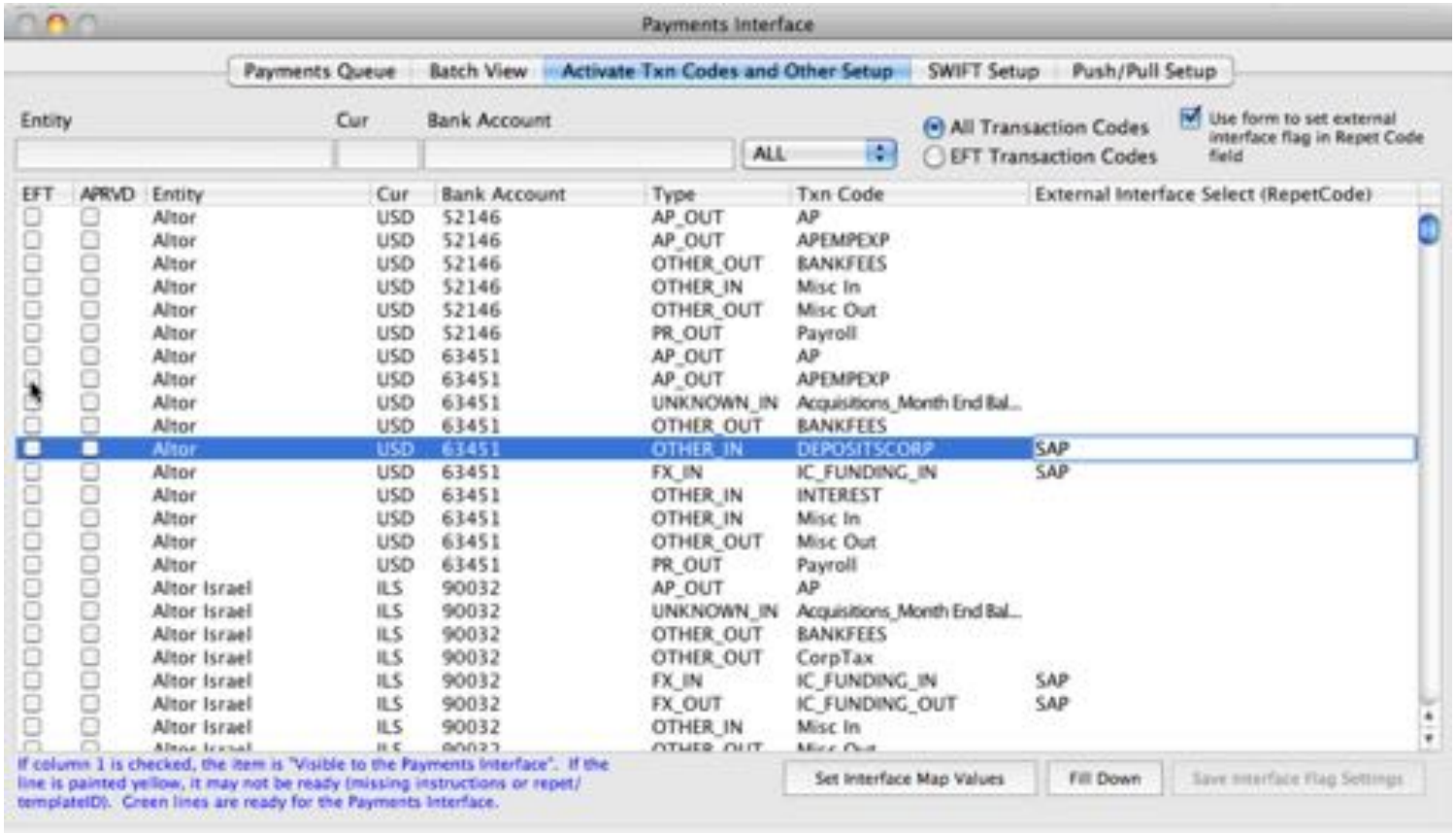
- To get Orbit ready for the SAP interface, complete the following tasks:
  - Identify All Transaction Codes for Payments and Receipts that you want to include in SAP reporting
  - For Payments – FX
    - Visit existing FX delivery instructions. Set a TemplateID value in each instruction and save it
    - Visit each FX, CASH OUT, transaction code and make those transaction codes “Visible to Payments Interface” and “Pre-Approved for EFT”
  - For Payments – NON-FX
    - Decide where you want to code TemplateID’s: in the Transaction Code or in separate delivery instructions. It’s easier to use the repet code field in the Transaction Code. It’s more flexible to code TemplateID’s in separate delivery instructions
    - Visit each NON-FX, CASH OUT, transaction code and make those transaction codes “Visible to Payments Interface” and “Pre-Approved for EFT”. Enter the TemplateID in the repet code field or attach delivery instructions to the Transaction Code
  - For Receipts
    - Visit each CASH IN, Transaction code that you want to include in SAP reporting
    - Set the value of the repet code field to “SAP”
    - We can use the “Activate Txn Codes and Other Setup” tab on the “Payments Interface” window to help with this

Getting Orbit Setup for the SAP Interface



- Set the “Use Form to Set External Interface flag…” checkbox to put the UI into the proper “mode”
- use the “EFT” and “APRVD” checkbox columns to make Transaction Codes visible to payments interface and pre-approved.
    - Note: the state is set for Transaction Codes as soon as you check/uncheck the checkbox. There is no need to “Save” the transaction code
- Double-click on a Transaction Code line to bring up the Transaction Code Detail editor and set delivery instructions or repet code values.
    - Note: You will need to save the Transaction Code in the editor when you make changes in the editor
- When you check the “EFT” column, the line will be painted yellow if the Transaction code cannot reach a TemplateID through either the repet code field or delivery instructions. If the Transaction Code is ready for the payments interface, the line will be painted green

Getting Orbit Setup for the SAP Interface



- The far right column can be used to set the value of the Transaction Code “repet code” field
- You can enter “SAP” here (for receipts) or a valid TemplateID (for payments)
- When you make changes in this column the “Save Interface Flag Settings” button will become enabled. Press it to save repet-code values for all displayed transaction codes