BUREAU OF THE
Fiscal Service
U.S. DEPARTMENT OF THE TREASURY

G-Invoicing

Reporting the Performance Amount

(Planned Future Enhancement)

Version 1.0 – January 10, 2022

Performance Amount

- Objectives
  - Address any rounding issues
  - Support refunds with zero Quantity
- Proposed Solutions
  - Allow agencies to report the Performance Amount
    - Currently, Amount is calculated as Quantity x Price [rounded]
  - Rounding
    - Allow agencies to submit Performance Quantity, Amount, or both
      - Quantity will be optional, if Amount is submitted
    - G-Invoicing to calculate missing Amount or Quantity, when needed
    - G-Invoicing to return error if user or client submits values for Amount and Quantity which are incompatible
  - Refunds
    - Allow agencies to report a new Performance Type (212 Refund) having an Amount but no Quantity

Zero Quantity Refunds

- Today, refunds are always based on Quantity (times Price)
- Adding a *Performance Amount* data element allows G-Invoicing to address another change request: Zero Quantity Refunds
  - Pass along a refund from Servicing to Requesting Agency without affecting quantity advanced or delivered/performed
  - Servicing Agency may submit a new Performance Type for a specific Amount, always with a Quantity of zero (required)
    - 212 – Refund (aka, Returned to Customer)
  - Refund must reference an active Order, Line and Schedule
  - Refund settles immediately, and may be adjusted by the Servicing Agency
- Zero Quantity Refunds will necessitate the de-obligation of remaining funds when Order is closed (even if quantity balance is zero)
- Does not require Partner 2 to be capable of handling Refunds
  - Until capable, the Requesting Agency will see the Refund in IPAC and in the G-Invoicing user interface, but not in their financial system

Rounding Issues

- When do rounding issues occur?
- How to avoid rounding issues?
- What can be done about them?

- *Definitions (from a G-Invoicing perspective):*
  - *Value-Based Orders have these attributes:*
    - *UOM = 'DO' (US Dollars)*
    - *Unit Price = $1.00*
  - *Quantity-Based Orders have these attributes:*
    - *UOM other than 'DO' (US Dollars)*
    - *Unit Price can be other than $1.00*

Quantity-Based Purchase Order

- A rounding issue of +/- 1 penny may occur when Unit Price is an odd decimal and Performance Quantity is fractional
- Example 1 (overpay):
  - Order/Schedule: Price = $9.99, Quantity = 1.00
  - Delivered/Performed 0.50 units for $5.00
  - Delivered/Performed 0.50 units for $5.00

    1.00 ✓ $10.00 > $9.99 X
- Example 2 (underpay):
  - Order/Schedule: Price = $4.01, Quantity = 1.00
  - Delivered/Performed 0.33 units for $1.32
  - Delivered/Performed 0.33 units for $1.32
  - Delivered/Performed 0.34 units for $1.36

    1.00 ✓ $4.00 < $4.01 X

Quantity-Based Sales Order

- G-Invoicing requires Unit Price and Quantity, then calculates Amount ($)
- Backing into a Quantity doesn't always work perfectly
  - Quantity = Amount / Unit Price (on Schedule)
- Example 3 (overpay):
  - Sales Order in ERP System: Amount = $200.00, Unit Price = $2.87
  - Quantity = $200.00 / $2.87 = 69.69 … sent to G-Invoicing as Order Quantity
  - Order in G-Invoicing: 69.69 x $2.87 = $200.01 > $200.00 X
    - In this scenario the Sales Order in the ERP system is out of sync with the Order in G-Invoicing before Performance is even reported
  - Advance Payment of 34.84 units x $2.87 = $ 99.99
  - Advance Payment of 34.85 units x $2.87 = $100.02

    69.69 ✓ $200.01 X
  - $200.01 is consistent with G-Invoicing but not with Sales Order

Purchase/Sales Order Mismatch

- Example 4
  - **Seller's system is value-based**, and project cost is set at $12,345.67
  - **Buyer's system is quantity-based**, and an hourly rate is set at $123.45
  - Order Originating Partner Indicator is 'R' so Unit Price = $123.45
  - Quantity = Amount / Unit Price = 100.005427 which rounds to 100.01
  - Option 1 (underpay)
    - G-Invoicing told that Quantity = 100.00 and Unit Price is $123.45
    - 100.00 x $123.45 = $12,345.00 which is 67 cents under project cost X
  - Option 2 (overpay)
    - G-Invoicing told that Quantity = 100.01 and Unit Price is $123.45
    - 100.01 x $123.45 = $12,346.23 which is 56 cents over project cost X
- Is this a rounding problem?
- Is it a precision problem?
- Is it due to the partners not agreeing to UOM, Unit Price and Quantity?

Proposed Enhancement (post-Mandate)

- Add a *Performance Amount* to FIDS, User Interface and APIs
  - Today the Servicing Agency supplies only the Performance Quantity
    - G-Invoicing calculates Amount (using the Schedule Unit Price)
    - Amount = Quantity x Unit Price [rounded]
  - In the future, the Servicing Agency will submit Quantity and/or Amount
    - G-Invoicing will calculate Quantity or Amount, as needed
- G-Invoicing will detect and correct rounding issues
  - Example 1 (overpay):
    - Order/Schedule: Price = $9.99, Quantity = 1.00
    - Delivered/Performed 0.50 units for $5.00
    - Delivered/Performed 0.50 units for $4.99 (corrected)

      1.00 ✓ $9.99 ✓
- Trading partners may continue to experience 1 cent differences until both agencies are calling the new API version

How to Avoid Rounding Issues?

- Enhancements will not be available until after the FY '23 mandate
- In the meantime:
  - Choose Units of Measurement that avoid fractional quantities, whenever possible, for example:
    - Deliver/Perform (6) EA (each) versus (0.5) DZ (dozen)
  - Value-based Orders offer the most flexibility if fractional quantities are unavoidable
    - Example: Purchase fuel by the DO (dollar), not by the GA (gallon)
      - This also accommodates fluctuations in price
      - Value-based Order is interpreted as "not to exceed" $ quantity .
  - Partners must agree on the following data elements on the Order:
    - Unit of Measurement
    - Unit Price
    - Quantity

Calculating Performance Amount

- Rounding issues may occur under the following conditions:
  1. A fractional Performance Quantity has been submitted
  2. Schedule Unit Price contains a decimal that ends in an odd number
     - e.g., 1.1, 1.03, 1.005, 1.0007, 1.00009
- Calculate the Performance Amount or Quantity with each new detail
  - Example: Schedule Quantity = 5, Unit Price = $2.99

1.	(existing)	Qty 0.5	Amt = 1.50	
2.	(existing)	Qty 0.5	Amt = 1.49	
3.	(existing)	Qty 0.5	Amt = 1.50	
4.	**(new)**	**Qty 0.5**	**Amt =**	**???**

  - Performance Amount Calculation:
    - Calculate the Total Amt = sum (Qty) x Price = 2.0 x 2.99 = 5.98
    - Calculate the sum of Existing Amts = 1.50 + 1.49 + 1.50 = - 4.49
    - Calculated **Amount** = Total Amt – sum of Existing Amts … = **1.49**

Applying Performance Amount

- G-Invoicing calculates Performance Amount when not submitted
- G-Invoicing will verify the submitted Amount when Performance Quantity is submitted too
  - When Amount values differ, examine the remaining Schedule Quantity
    - When remaining quantity = 0, then 400 error: *Performance Amount must be <calculatedAmount> or null*
    - When remaining quantity > 0, check the difference
      - > If difference is no more than a dollar then use the submitted Amount
      - > If difference > $1.00 then 400 error: *Submitted Performance Amount must be within a dollar of <calculatedAmount>*
  - Example: Schedule Quantity = 5, Unit Price = $299.99
    1. (existing) Qty 0.5, Amt = 150.00
    2. **(new) Qty 0.5, (new) Amt = 150.00**
       - > Calculated Amt (149.99) <> submitted Amt (150.00)
       - > Remaining Qty (4.00) > 0, so check the difference …
       - > Difference of $0.01 is less than or equal to $1.00, so **(new) Amt = 150.00**

Calculating Performance Quantity

- Allow agencies to submit Performance Amount without a Quantity
- G-Invoicing will then calculate Quantity for each new Performance detail
  - Example: Schedule Quantity = 5, Unit Price = $29.995

1.	(existing)	Amt = 9.00	Qty = 0.30
2.	(existing)	Amt = 15.00	Qty = 0.50
3.	(existing)	Amt = 20.99	Qty = 0.70
4.	**(new)**	**Amt = 27.00**	**Qty =** **???**

  - PQ Calculation:
    - Total Qty = sum Amt / Price = 71.99 / 29.99 = 2.40
    - Sum of Existing Qty = 0.30 + 0.50 + 0.70 = - 1.50
    - Calculated **Quantity** = Total Qty – sum of Existing Qty = 2.40 - 1.50 … = **0.90**

Wrap Up

- Performance Amount will be added to Performance FIDS
  - Rounding issues may then be resolved
  - Zero-quantity refunds may then be supported
- No plans to resolve Purchase/Sales Order mismatches (on slide 7)
- Agency Controls
  - Agencies will need to upgrade to new API version before Performance Amount can be submitted to address rounding issues
    - By that point, agency users and agency systems should be able to accommodate errors due to miscalculated amounts
    - A partner that cannot yet pull down Performance Amount may still be a dollar off
    - One agency may benefit from the upgrade while their partner still experiences rounding issues
  - The use of Refunds need not be controlled through Feature Management
    - Servicing Agency will need to upgrade in order to submit Refunds
    - Until then, Servicing Agency continues using IPAC for zero-quantity Refunds
    - Requesting Agency will continue to reconcile Refunds through IPAC until both partners have upgraded

Implementation Details

- The following slides document what needs to be changed in published G-Invoicing specifications in order to implement this new feature
- Implementation details are provided so that:
  - Product Owner(s) can confirm detailed business requirements
  - Development team(s) can visualize changes needed to support new features
- At this time, we cannot predict when development of this feature will begin, when it will be released into QA or Production, or in which release it will be included
- When the time comes to begin work on these features, the details will be added to the specifications, a release will be targeted, and API specifications will be republished
- One of the implementation details described below is a move from XML to JSON payloads for APIs involving GT&Cs, Orders and Performance
  - Future (post-mandate) APIs will only support JSON payloads
    - *Note: Newer APIs (e.g., Pull Org, Remittance) are already in JSON format*

Changes to Performance FIDS

- Amount (for rounding rules and refunds)
  - New data element at the Performance Detail level: performanceAmount (JSON)
  - Definition: The (dollar) value of a Performance transaction (e.g., advance, delivery) recorded against an Order/Line/Schedule, equal to Unit Price times Quantity.
  - When Performance Status is 'STL' (settled), the Performance Amount is equal to the Transfer Amount (sent to IPAC)
    - *Note: G-Invoicing team may decide to repurpose Transfer Amount (in FIDS) as simply (Performance) Amount, because we don't currently include Transfer Amount in the XML payload. Status, Transfer Date, DO Symbol and Doc Ref Num all indicate if the Amount was settled or not, so Transfer Amount is redundant once (Performance) Amount is added.*
  - Validation Rules
    - Amount and Quantity are conditional – one, the other, or both are required

Changes to APIs for Rounding

- Performance Amount
  - Add performanceAmount to JSON payload for single Performance push and pull APIs
- Rounding Algorithm
  - Push API Performance Rules
    - Apply new algorithm when calculating Performance Amount from Quantity and Price
    - Apply new algorithm when calculating Performance Quantity from Quantity and Amount
    - Check that the difference in Amount supplied by client against the calculated Amount is no more than $1.00

Note: Assume API payloads have already been converted to JSON

Changes for Refunds

- Performance FIDS
  - Add new data type value and description
    - Available to Servicing Agency for advanced or non-advanced Schedules
  - Allow UOM to be null value for performance type 212 only
  - Performance Date can be in the past but not in the future
  - Refund can be adjusted by the Servicing Agency
  - Refund amount may not exceed the Transfer Amount for that Schedule
- APIs
  - Pull Performance
    - Return a null value for UOM in payload for Refund
  - Push Performance
    - Accommodate new Performance Type (212) as described on previous (FIDS) slide
    - Allow zero Quantity record with non-zero Amount for Refund
    - See rules on previous slide (FIDS for Refunds)

Note: Assume new performanceAmount already added to JSON

Other Changes (not related to APIs)

- Performance User Interface
  - Performance Summary page
    - Sum of Amount(s) for an entire Performance transaction could be shown
    - Add filter value for new Performance Type (212)
  - Performance Transaction Details page
    - Amount already is calculated and displayed, but the source will be directly from the database
    - Display a 212 type (Refund)
  - Order Schedule Balance (OSB) page
    - Amount already is calculated and displayed, but the source will be directly from the database
    - Remaining Amount may need to be recalculated on the OSB page
    - Display 212 type, related to a Schedule, and allow user to create a new Refund
  - Create Performance modal
    - Amount must be added, allowing user to enter Amount, Quantity or both
    - Rounding algorithm must be applied, as described in this slide deck
    - Allow users to create a Refund (212), for which Quantity is zero and UOM is null
  - Export Summary and Export Details (CSV)
    - Add new column for Amount or change the source of Amount
  - Performance Report (from Reporting Database)
    - Change data source for Amount to be directly from the database (not calculated)

Version History

Version	Date	Release	Description
v0.1	12/1/2021	TBD	Original version, reviewed by Treasury, CGI and Oracle
v0.2	12/3/2021	TBD	API payload will be JSON. Added technical details.
v0.3	12/7/2021	TBD	Minor clarifications and updates following internal team review
v0.4	12/22/2021	TBD	Changed acceptable rounding varience from $0.01 to $1.00
v1.0	1/10/2022	TBD	First published version on G-Invoicing website