

Enhancements and fixes are listed below, with their associated identifiers in brackets, as applicable.

Enhancements

API

- Watchdog is now implemented for the **Monitoring MSMQ OutputQueue** when thread is dead [IN_20241113_0007].

Interoperability

- Amendment number 176 has been implemented for the ISO 4217 Currency Code Service. This amendment introduces a new currency code "XCG" and the name "532" to the UPU-IP database, BAMS, PPSC, and other related systems [RFS_20250317_0029].

Fixes

API

- The order retrieval process has been enhanced for better reliability with external partners when calling the Get() service [PB_20241204_0001].

Interoperability

- The issues related to the generated EDI file with a default value have been fixed [PB_20241011_0003] [PB_20241206_0002].

Enhancements and fixes are listed below, with their associated identifiers in brackets, as applicable.

Enhancements

API

- The new API operation, **RequestEdiCancellation**, allows UPU-IP partners to request cancellation of postal payments to be paid by the EDI partner. EDI partners do not need to modify their existing processes to handle cancellation requests from UPU-IP partners in addition to EDI partners.
- When Monord (CC) and Monord (DS) EDI messages were previously prevented because of a postal payment lock, the **Unlock** function has been improved to guarantee that these messages are generated correctly. This modification affects how postal payment cancellations within UPU-IP partner exchanges and EDI cancellation requests between UPU-IP and EDI partners are processed.

Interoperability

- The EDI converter now generates EDI Monord messages with a cancellation request (DS) event.
- The EDI converter now validates and processes EDI Resord messages from EDI partners with cancellation requests **accepted (ED)** or **rejected (DR)** events. The UPU-IP now accepts state transitions to cancellation request accepted (ED) and rejected (DR) events.

Fixes

API

- The UPU-IP API **Issue operations** now correctly prevents the emission of postal payment with an incorrect **ValidUntil** UTC timestamp set in the past [PB_20240905_0004].

Interoperability

- The EDI converter correctly logs the event state code that is linked to the Resord file rejection [PB_20231109_0001].
- To avoid the EDI partner rejecting the EDI message, the EDI converter ensures that the constraining data lists in M38 are mapped and respected [PB_20240905_0003].
- When not supplied by API, the EDI converter fills in the default data value, **Unspecified**, for the M38 mandatory element, **Office**, preventing the EDI partner from rejecting the EDI message [PB_20240827_0002].

Enhancements and fixes are listed below, with their associated identifiers in brackets, as applicable.

Enhancements

Database

- Added a new ISO currency code for Zimbabwe: ZWG - Zimbabwe Gold [RFS_20240702_0014]

Fixes

API

- Fixed the error encountered during the transition of money order payment state from **EP** to **FP** due to overlapping jobs. The issue has been averted, and the money order event is now stored correctly [PB_20240429_0002]
- Improved the handling of modified money order information. Before a service can store data, UPU-IP now verifies the state code of the money order. If the state code is unchanged, the service retrieves the latest changes, adds a new event, and stores the data. If the state code has changed, the service throws an exception.
- Fixed the error encountered while calling the procedure, SP_SET_MONEYORDER [PB_20240425_0001]
- When calling external services for a reimbursement of a money order, UPU-IP checks first if the **EP** or **ER** event has been stored in the database and only stores the event if it does not exist yet.

Interfaces

- UPU-IP now generates correctly the encryption envelope for Monord files exchanged between two UPU-IP partners for PPS* Clearing purposes [PB_20240607_0001]
- Improved the handling of money order state transitions so they no longer result in collisions; the event is stored properly and the correct details are recorded in the log.

This release contains the fix listed below, along with its associated identifier.

Fix

PPS*Clearing / QCS-related area

- The export to QCS Finance and PPS*Clearing now generates Monord and MonordAck files when the issuing partner is an external partner and the paying partner is calling the UPU-IP API (PB_20231030_0003).

Enhancements and fixes with their associated identifiers in brackets, as applicable, are listed below.

Enhancements

- Ability to regenerate multiple EDI files to feed QCS Finance and PPS*Clearing to manage support issues [PB_20231030_0003]
- UPU-IP now integrates the latest version of the PTC.Transport library, supporting the latest Rebex third-party library and the latest SFTP algorithms [PB_20231030_0006]

Fixes

API

- Calls and respective responses to the UPU-IP API operation ReimburseEx() are now logged in the log table

Database

- Reference data for currencies has been updated to match the latest ISO list [PB_20231030_0002]

Interfaces

- UPU-IP now implements mapping of CAPTURED & REGISTERED states of FORSAGE system [PB_20230921_0001]
- Data mapping between the UPU-IP and FORSAGE systems now includes the message to the recipient (free text information from the sender) in the money order details [PB_20231030_0001]

PPS*Clearing / QCS-related

- The export to QCS Finance and PPS*Clearing now generates a Monord file when the issuing partner is an external partner and the paying partner is calling the UPU-IP API [PB_20231030_0003]

A fix with its associated identifier in brackets, as applicable, is listed below.

Fixes

Interoperability

- UPU-IP now acknowledges Resord messages with **EI** and **EW** states received from EDI partners

Fixes with their associated identifiers in brackets, as applicable, are listed below.

Fixes

API

- The UPU-IP API operation **GetByCriteria** now handles requests without purchase date parameters, and the error is reported correctly without modifying the API contract [TFS 6606]
- The UPU-IP API operation **Issue now** reports the error details without spelling mistakes if the transferred amount falls outside of or exceeds the accepted range [PB_20230203_0002]
- The UPU-IP API operation **Pay now** returns a harmonized message if the calling partner is different from the paying and authorized partner [PB_20230203_0001]

Interoperability

- The EDI converter now generates an **IFSError** message with the relevant error code for each event. This is when the EDI message being rejected has multiple events and only one causes the rejection [TFS 6555]
- The EDI converter now generates an **IFSError** message with the relevant error code X when no agreement exists [TFS 6554]
- The EDI converter now ensures that the elements, EvtUTCDDT in Event element of Monord and Resord files, UTCDDT in the Purchase element of Monord files, and UTCDDT in the Payment element of Resord files always have a value in the generated EDI files [TFS 7204]

Fixes with their associated identifiers in brackets, as applicable, are listed below.

Fixes

API

- The UPU-IP API Reimburse operation for an external Forsage order now triggers an additional status check in Forsage. If the status has evolved, the status gets automatically updated in UPU-IP [IN_20220718_0004]

Service

- The expiration service now considers 31 days after emission before Forsage orders expire in order to exclude the emission day from the validity duration as in the Forsage system [IN_20220718_0004]

Interoperability

- For each emission, payment, or cancellation operation with an EDI partner, the EDI converter now generates Monord/Resord, as well as an IFSAck file, even if a new event occurs between the time of registration or reception and the time of generating the EDI file by the EDI converter [IN_20210824_0018]
- The EDI converter now logs the correct state code of new events added to orders in the EDI exchange audit trail [IN_20220204_0006]
- UPU-IP no longer feeds QCS Finance twice the exchanges between IFS and external partner [PB_20210929_0003]
- The EDI converter now supports EDI files with version 02 EDI standard [IN_20220916_0004]

Enhancements and fixes with their associated identifiers in brackets, as applicable, are listed below.

Enhancements

API

- Enhanced the **GetEx** API to check the **UseScAsLocalId** flag in BAMS. If the flag is set to "Y" (true), the search engine retrieves postal payment orders where the secret code is equal to the local ID input parameter.

Database

- It is no longer allowed to issue postal payment orders with the space character in the postal payment identifier.

Interface

- It is now possible to search for postal payments that are available for payment using a short identifier (a secret code) provided by the customer during the postal payment order creation. The short identifier is used instead of the local ID to filter postal payments, if the bilateral agreement between the two organizations has 'Yes' for the property **Use secret code as local id**.

Enhancements and fixes included in this build are listed with their associated identifiers in brackets, as applicable.

Enhancements

API

- **Addition of Electronic Advance Postransfer (EAP) API component to the UPU-IP:** The EAP component serves external systems (including the PosTransfer app) that want to exchange (register and collect) information about electronic advance data of postal payments or PosTransfer. This new component implements the following features:
 - New EAP Register service to be called by the PosTransfer app without requiring a client certificate (when the app is published for end users), but must have a token for every operation to restrict access to the published PosTransfer app. It includes a new API operation for submitting an EAP and returning an EAP code when successful.
 - New API operation in EAP Collect service (available only to enrolled clients) to allow retrieving EAP details related to the organization based on the EAP code provided.
 - UPU-IP EAP repository to store EAP information for any designated operator (DO) with a published PosTransfer app.
 - New EAP Windows service to delete expired EAPs after a certain period.

Enhancements and fixes included in this build are listed with their associated identifiers in brackets, as applicable.

Fixes

API documentation

- The UPU-IP API documentation now only includes description of objects in use in UPU-IP API; objects in use in BAMS and EAP API are now documented in their respective documentation [ITSM ticket IN_20220602_0016].

Database

- Resolved the issue of bilateral agreement cache loads when BAMS API returns an agreement with CurrentState parameter null [ITSM ticket IN_20220614_0013].
- Changed the Country property on TAddress class from "ISO Country code" to "2 digit ISO Country code" to ensure that the UPU-IP API issues postal payments with a country code that is valid and accepted by IFS partners who expect a 2-digit ISO country code.

Enhancements and fixes included in this build are listed with their associated identifiers in brackets, as applicable.

Enhancements

API

- Implemented a secured **Transport Level Security (TLS)** API that encapsulates security credentials and claims with every message, providing end-to-end security independent of the transport protocol. Web service requests and responses to and from UPU-IP are encrypted and authenticated with x.509 certificates. This API was introduced in release 1.04.02.
- Integrated the enhanced **LoadAgreements** API of BAMS to now allow retrieval of bilateral agreement requests corresponding to the search criteria such as current state of the agreement, issuing/paying organization, and date.

Database

- Each UPU-IP postal payment order is now associated with a unique bilateral agreement identifier.

Interoperability

- UPU-IP now integrates the new bilateral agreement through **BAMSPROXY** class. This new bilateral agreement came about after BAMS introduced a new interface to define bilateral agreement and retrieve a new BA schema, improving the ability to define each property dynamically to match the needs of customers.
- Upgraded the UPU-IP platform in order to support changes done to BAMS as per BAMS 2021 project, which now supports customer services and fees, and agreement properties.

Enhancements and fixes included in this build are listed with their associated identifiers in brackets, as applicable.

Enhancements

API

Postal payment retrieval

- The UPU-IP API operations **Get** and **GetByCriteria** have been updated to include new call parameters that filter the retrieved postal payments. The update allows the integrating systems to retrieve only updates that have been made since the last call

Fixes

Interoperability

State transitions and EDI exchanges

Several corrections have been made to the EDI converter, which takes care of postal payment events between UPU-IP partners and EDI partners (IFS and STEFI).

- The EDI converter has been enhanced to store a copy of the exchanged EDI files in a central repository
- When failing to collect a file from the FTP folder, it now moves a copy of the files to an error folder
- Backup EDI files are stored in a structured hierarchy
- EDI message files are tracked with information for support

Reference data has been revised to

- prevent cancellation of GT postal payments
- authorize reception from EDI partners of expired postal payment from IN and GT products
- support reimbursement of postal payments expired or declared as impossible to pay by EDI partners
- support expiration of postal payments for the different products and partners of exchanges (EDI or UPU-IP)

The EDI converter now correctly

- acknowledges or rejects EDI messages even if a new event occurs between registration of event, reception of event and the moment the EDI file is generated by the converter job [IN_20210824_0018 - PB_20210826_0014]
- generates IFSErr with the reason code to the incorrect event, in particular reason X when a valid bilateral agreement is missing

PPS*Clearing / QCS-related

- The EDI converter which feeds other systems (QCS Finance and PPS*Clearing) with UPU-IP exchanges has been corrected to apply the same ListID when feeding both QCS Finance and PPS*Clearing with the same EDI message

Enhancements and fixes included in this build are listed with their associated PRCR or CRM numbers in brackets, as applicable.

Enhancements

API

- New API operation IssueWithLock() has a locking mechanism enabled by default. Issue postal payments that require additional process before they become available to the paying partner organization, for example during analysis for a specific authority, set of control operations
- New API operation Lock() enables a lock on postal payment preventing other partners' operations. Reserve postal payments available under offline process and prevent any evolution (state change: expiration, cancellation). Automated service requires locking to payment partners during process in external account systems; while payment is restricted to specific offline office (e.g. point of sale); or analysis for a specific authority, set of control operations
- New API operation Unlock() disables the lock on postal payments. Release a locked postal payment, and at the same time (optionally) record an operation from the current partner, even if the postal payment has passed its end validity date
- Update of the existing Cancel(), Reimburse(), Pay() and ConfirmPayToAccount() API operations to return ErrorCode 14 (operation is not allowed along with ErrorDetails) when called on locked postal payments by a partner organization that is not the owner of the lock.
- Update of the Get() and ReceiveForAccount() API operations to retrieve only postal payments available for payment which are not locked.
- Update of the GetByCriteria() and CheckStatus() API operations to include lock status details in response.

Interoperability

- The converter to former M38 EDI messages guarantees exchanges between new and old technologies. It delays while locking the conversion and transports new postal payment details to the former postal payments platform based on FTP EDI files. Once unlocked, it transmits new postal payments details.
- The converter to former M38 EDI messages automatically rejects with IFSErr message (and reason code T - Rejected because of a wrong event sequence T") cancellation requests received from EDI partners that request cancellation of postal payments locked by the paying organization.

Service

- The Expiration service delays expirations of lock postal payments until they are unlocked.

Configuration

- The UPU-IP now includes new parameters to support definition of the maximum size of API operation response (maxReceivedMessageSize) and maximum execution delay (exe-

cutionTimeout) and three additional parameters for the RIA SSL configuration (RIA_SSL_ENABLED, RIA_SSL_FRIENDLY_NAME, RIA_SSL_URL)

Interface

The UPU-IP proxy to RIA system now supports SSL connection with client certificate.

Fixes

API

- Issue() API operation now validates that the provided moID is correct (that it does not exceed the maximum length).
- ReceiveForAccount() API operation now retrieves payable postal payments from GT product.
- ConfirmPayToAccount() API operation now correctly allows activities only on GT and IN products.

Interoperability

- The converter to former M38 EDI messages now correctly includes a payment element in Resord with EI event for receiving IN/GT postal payment from the EDI partner.

Interfaces

- Update of RIA proxy to support SSL connection with client certificate

PPS*Clearing / QCS-related

- The export to PPS*Clearing now generates Resord files with EE events.
- The export to PPS*Clearing now includes EE and FP events even for postal payments for which the bilateral agreement does not exist (postal payment with external system) or for which the bilateral agreement has expired.

API documentation

- The UPU-IP API documentation has been revised to include new API operations and is now available online: https://upu.api.post/upu-ip_api/api/WSDistributor.IExternalService.html

Enhancements and fixes included in this build are listed with their associated PRCR or CRM numbers in brackets, as applicable.

Fixes

Configuration

- Renamed the configuration parameter, **EXPIRY_PERIOD** to **EXPIRY_PERIOD_IN_MIN** for clarity

PPS*Clearing / QCS-related

- Fixed the issue of Resord events from 2 UPU-IP partners not imported into the PPS*Clearing system [CRM106920]
- Fixed the issue of Monord acknowledgements containing events from 2 UPU-IP partners not imported into the PPS*Clearing system [CRM106919]
- If an error occurs while exporting to QCS, UPU-IP no longer creates duplicate messages in the PPS*Clearing System [CRM106885]
- Orders pushed to the Forsage system now have the product type code [CRM106889]
- Orders exchanged between UPU-IP partners which were under non-Clearing agreements are now exported to PPS*Clearing and QCS Finance systems [CRM106886]
- The export to PPS*Clearing now includes files exchanged between UPU-IP client and external system; Tracing is now available in QCS, and orders can now be settled in PPS*Clearing [CRM106887]
- For orders exchanged between UPU-IP partners without EDI, events exported to the PPS*Clearing system now include cancellation (ED) events [CRM106888]
- Orders exchanged between IFS and UPU-IP partner or between UPU-IP partners, now expire according to the expiry setting defined in the bilateral agreement [CRM106891; CRM106890]
- Inpayment (cash-to-account) orders with the recorded outbound EI (Payment Impossible) event, no longer produces recurrent warning logs every minute; The inbound FI (Payment Impossible) event is now created successfully [CRM106894]
- UPU-IP now assigns the expiry status to Inpayment (cash-to-account) orders properly [CRM106892]
- UPU-IP now sends an IFSErr message with the correct test element value in the pre-production environment [CRM107982]
- When the bilateral agreement expires, UPU-IP now generates an IFSErr message successfully [CRM108029]
- UPU-IP no longer fails to generate an IFSErr message even if the new bilateral agreement is not yet found in the UPU-IP cache [CRM106163]

API

- The GetFXRates service logic has been updated; UPU-IP now provides the correct exchange rate to the calling operator [CRM106141]
- Updated the UPU-IP API documentation to indicate PPS*Clearing-specific properties which are not used in both IFS and UPU-IP [CRM 106142]

- When the exchange rate file is not available, UPU-IP now returns a clear error message [CRM106143]
- When exchanging with an IFS 4 partner, the UPU-IP API now checks the bilateral agreement expiry date [CRM108028]
- Updated the description for **TMODetails.Remuneration** in the UPU-IP API documentation [CRM106140]

Interfaces

- Added a new required field for compliance reason when paying a RIA money order [CRM105635]

Versions prior to 1.02

These versions comprise the initial releases of UPU-IP, which include the following features:

Centralized system

- Switch for real-time message routing
- Repository of all payments; unique reference in case of discrepancy between connected Posts and partners
- Central management of bilateral agreements in BAMS

Urgent (instant) service with Web services solution

- Real-time communication
- Synchronous protocol for urgent international postal payment services Payment anywhere facility
- Client decides at Post, at a sub-agent of the Post, or at another agent

One-time connection

- One-time integration of further external partners or MTOs to UPU-IP for every Post

Existing add-on services remain available

- **PPS*Clearing**: exchanges over the UPU-IP feed PPS*Clearing to be included in settlement and netting process

Interoperability

- **Converter to former M38 EDI**: bridge between the new UPU-IP and the EDI-FTP platform to convert and transport M38 EDI messages to the previous postal payments platform based on FTP EDI files, and to guarantee exchanges between new and old technologies
- **External systems proxy** (e.g. Forsage): to relay postal payment events to external systems if the external system is the reference system (push mode)

First release

The **UPU Interconnection Platform (UPU-IP)** is a platform that enables near real-time exchange of postal payment-related messages between business partners.

Designed to address the limitations of transmitting EDI postal payment messages via the File Transfer Protocol (FTP), the UPU-IP exposes Web Services to business partners to interface with their system.