What's New in

Temenos Transact

March 2021

Table of Contents

Release Highlights

Application Framework

Infrastructure » Customer Data Access Logging

Temenos Transact and TPH have a structured logging mechanism of all user activities. This mechanism is configurable in a `USER` profile, which ensures that all activities of the user are logged for audit purposes. This information is also available for extracts and reports, if required. The logging is now extended for every `CUSTOMER` involved in a transaction or Enquiry. The feature to log additional (customised) information for each of those `CUSTOMERs` is also available for L3 customisation. This customising feature is currently available only for jBC and not for Java.

Click here to understand the technical impact of this enhancement for customisation and upgrades.

Click here to understand the installation and configuration updates for this enhancement.

The topic related to this feature is given below:

Customer Data Access Logging

Banking Framework

System Tables/Standing Order » Structured Address Fields in BENEFICIARY and STANDING.ORDER

To comply with ISO 20022 standards, payment messages should include detailed address information for customer addresses. The BENEFICIARY and STANDING.ORDER applications in Temenos Transact have been enhanced with additional fields to support detailed address information for beneficiary customer, ultimate creditor and ultimate debtor. The address details captured in the BENEFICIARY application are automatically populated in the STANDING.ORDER application. The address details captured in the STANDING.ORDER application are mapped to the PAYMENT.ORDER application, if the payment is instructed to be executed with PAYMENT.ORDER.

Click here to understand the technical impact of this enhancement for customisation and upgrades.

The topics related to this feature are given below:

Beneficiary and Ultimate Creditor Details

Beneficiary, Ultimate Creditor and Ultimate Debtor Details

Arrangements Architecture » Defining Mandate Requirements for Customer Portfolios

In Temenos Transact, users can set up mandate requirements at the customer or account level, indicating the conditions in which various types of transactions are considered approved.

Temenos Transact now allows banks to define the mandate requirements at customer's portfolio level. This functionality provides customers options to segregate the investment approvals and assign the investment-related approvals to the relevant business functions in their companies.

The topics related to this feature are given below:

Defining Mandate Requirements for Customer Portfolios

Creating a Mandate Application Group

Authorising or Deleting a Mandate Application Group

Viewing, Amending or Reversing a Mandate Application Group

Creating Mandate Requirements

Authorising or Deleting the Mandate Requirements

Amending or Reversing the Mandate Requirements

Search Mandate Requirements

EB.MANDATE.PARAMETER

Delivery » Delivery Preferences for Portfolios and Additional Recipients

Temenos Transact now provides banks the ability to capture the mailing or alert preferences of customers through the `DE.CUSTOMER.PREFERENCES` application. The bank can define the delivery preferences for a customer and/or a specific account of the customer. Outward delivery messages or alerts are processed based on the customer's preference.

The Delivery module has been enhanced with the ability to send an additional copy or redirect a customer delivery message or alert to other recipients, based on the preferences indicated by the primary customer. Also, the Delivery module now provides an option to define the contact preferences for a customer's portfolio, which allows banks to capture these at the account or customer level.

This functionality allows customers to:

- Redirect or send an additional copy of their messages or alerts to other recipients.
- Maintain separate alert preferences for their portfolios individually.

The topics related to this feature are given below:

Delivery Preferences for Portfolios and Additional Recipients

Tasks for Contact Preferences

Enquiries and Reports

Accounts » Capturing Company or Branch-specific Settlement Accounts

The `CUSTOMER.SSI` application in Temenos Transact is used to define the settlement account for different types of transactions for customers. Some customers may want to create and settle Forex (FX) transactions in different locations or branches. In such cases, the bank may define branch-specific default settlement account for the customer.

The *Company* field is added in the `CUSTOMER.SSI` application, which allows the user to capture the branch company for which the settlement account will be defined and used. During the transaction initiation process, the appropriate default settlement account is retrieved based on the branch-specific configuration in `CUSTOMER.SSI`.

The topics related to this feature are given below:

Capturing Company or Branch-specific Settlement Accounts

Defining Settlement Account Specific to Branch Company

Private Wealth

Securities » Allowing Internal or Cash Account in Transaction

There are some fund Asset Management Companies (AMCs) that do not have a nostro or vostro relationship with the distributing bank. In such instances, the funds are kept in a suspense account and at the end of the day a net payment is made to the AMC after receiving a confirmation from the distributing bank. There is a requirement to support such instances and allow internal or cash account.

The Securities module in Temenos Transact is enhanced to accept internal or cash account. The internal or cash account setup in `CUSTOMER.SECURITY` application is defaulted in the transaction. This helps the banks who do not maintain a nostro or vostro relationship with the distributing bank to input transaction using an internal or cash account.

The topics related to this feature are given below:

Allowing Internal or Cash Account in `CUSTOMER.SECURITY`

Defaulting the Broker Account in `SEC.TRADE`

Corporate Actions » US Tax Regulation Compliance

The Section 871(m) of US tax regulation requires the system to calculate and account withholding tax on US instruments and structure notes held in Securities (SC) module.

Temenos Transact can now determine the income generated from US instruments and structure notes with US security underlying and consider the income for taxation under the new regulation. The following are the two types of withholding processing:

- Withholding tax calculation and accounting, when bank is the withholding agent
- Withholding tax calculation and no accounting on the customer's account, when the issuer is the withholding agent and tax is withheld at source.

This regulation ensures that any customer who does not hold the US security but gets the payoff equivalent to the income generated on the US security is now accounted for withholding tax.

> **The topics related to this feature are given below:**
>
> Configuring Notional Dividend
>
> Withholding Tax based on Notional Proceeds

Securities » Market Indicator and Settlement Party Tags

Market indicator refers to Tag 22F in SWIFT messages MT540/MT541/MT542/MT543. This is a mandatory tag in mandatory sequence E (settlement details). SWIFT tags support various formats such as P, Q and R for tags related to settlement parties. Rules configuration is required to define the format to be used for settlement parties in messages MT540/541/542/543.

The Securities module in Temenos Transact now supports the setting up of rules to:

- Default market indicator at the transaction level.
- Define the SWIFT format to be used for settlement related tags in SWIFT messages MT540/MT541/MT542/MT543.

> **The topics related to this feature are given below:**
>
> Defaulting of Market Indicator at Transaction Level

Securities » Enabling Cash SSI setup and Improvements to Securities SSI

In private banks, for the various products (like, Equities or Bonds), the counterparties can play various roles like brokers or custodians. The Standard Settlement Instructions (SSI) for each counterparty may change according to the role they play. Each counterparty have their own life cycle and to ensure faster payments and settlements, the SSI details for each broker and depository needs to be setup along with their validity period. When a trade is executed, these settlement instructions are to be sent to the custodian. Thus, it is necessary the system defaults the correct SSI. SSIs place a crucial rule to reduce the settlement failures and hence maintain a high level of efficiency in the quality of instructions to the custodians.

The following are the new functionality of this enhancement:

- The system is enhanced to record both the securities and cash SSI details.
- Added `SC.CASH.SSI.INSTRUCT` application to record the details of the cash SSI. This application records the beneficiary details, intermediary bank details, broker account and so on, of the broker or depository.
- Additionally, the system now allows:
  - Setting up of duplicate SSI for the same SSI combination where the final settlement account alone differs.
  - Setting up of a validity period to be set per SSI. If trade takes place after the validity period, then the system does not select the expired SSI. The bank can also set SSI based on issuer. Under such cases, the system additionally checks the issuer of the security before defaulting the SSI.
- Grouping of the ISINs to set-up a single SSI, where a single SSI is applied for several ISINs.

- The system now allows the bank to define whether the SSI is applicable for a particular broker or depository at counterparty level. If the SSI status of the broker or depository in `CUSTOMER.SECURITY` is set to Deactivated, the system does not check both the securities and cash SSI for the counterparty.

The topics related to this feature are given below:

Enabling Cash SSI setup and Improvements to Securities SSI

Determining SSI using ISIN and Issuer Group

Creating or Updating Cash SSI and Security SSI

Trades with Manually changed SSI

Regional Banking Solutions

Africa CEMAC Model Bank

Asset Classification and Provisioning

This functionality allows the classification of assets based on the number of payments in default or based on the duration of the debit balance of the account without any significant credit movements.

Storing the asset classification categories, the definition of parameters and methods, the frequency for calculation of provision based on asset classification, provision to store the bill details, details of overdrawn accounts are all existing functionalities in the system that will be reused for this functionality.

The asset classification can be made for syndicated loans, where the overdue will get written to the `SL.OD.PART` and `SL.OD.DUES` applications.

The `AFRACP.ACCOUNT.DAYS.PAST.DUE` application is introduced to store the movement in the classification based on the number of days past due.

The topic related to this feature is given below:

Asset Classification and Provisioning

Argentina Model Bank

Accounts » Embargo and Liberacion APIs

The Embargo Process functionality allows the processing of different trades from COELSA, which manages all the trades received from AFIP (the tax revenue agency of Argentina) on the defaulted customer. The two main steps in the seizure process through COELSA are the online and batch processes.

The request sent by COELSA through the Embargo API will be expressed based on the account currency either Peso and USD by AFIP (principal and interest). In case the account currency is USD, the seizure will consider only the whole number, without decimals and rounding.

The moneda_cuenta and monto_cuenta fields are added to the debito section of the request and will be expressed in the currency of the account.

The ARACCT.EMBARGO.DETAILS.AUTHORISE routine is modified to check the seizure amount provided in the monto_cuenta (account amount) field, the account currency, and released amount, provided in the online seizure request on customer account from COELSA, and apply the seize amount requested complying to the requirements related to the embargos process in Argentina.

The topic related to this feature is given below:

Accounts

Taxes » Automatic Tax Returns

This functionality allows banks to process all the return files before calling the payment orders, with the respective control validations. The system will discard the tax returns for any account whose base currency is not ARS.

The following items are introduced as part of this functionality:

- The ARTAXS.JURISDICTION.PARAMETER application allows banks to debit different internal accounts based on jurisdiction.
- The ARTAXS.JURISDICTION.PARAMETER,INPUT version allows banks to input records in the ARTAXS.JURISDICTION.PARAMETER application.
- The ARTAXS.JURISDICTION.PARAMETER,AUTH version allows banks to authorise the inputted records in the ARTAXS.JURISDICTION.PARAMETER application.
- The ARTAXS.JURISDICTION.PARAMETER.AMEND enquiry allows banks to amend or reverse a record in the ARTAXS.JURISDICTION.PARAMETER application.
- The ARTAXS.JURISDICTION.PARAMETER.AUTH enquiry allows banks to authorise the inputted records in the ARTAXS.JURISDICTION.PARAMETER application.
- New fields are added to the ARTAXS.RETURN.TAX.FILE.DETAILS application to store the details of SIRCREB tax return file and also to include Buenos Aires tax return file.
- New fields are added to the ARTAXS.RETURNS.FILE.PROCESS.STATUS enquiry to incorporate the sequential number of the record.

The topic related to this feature is given below:

Taxes

Belgium Model Bank

TOB Tax » Withholding Tax

This functionality allows banks to calculate the Belgian transaction tax (TOB tax) for persons. In case of co-holding with a customer who is not a Belgian resident the calculation of the tax is split in equal parts among the account holders and only for the part of the account attributable to the Belgian fiscal residents the income is submitted to the TOB tax. This functionality also allows banks to exempt any client from the TOB tax. The exemption is based on the presence or not of the ETOB: ATTEST. NON-SUBMISSION BELGIAN TAXES document in the `CUSTOMER` documents.

The TOB tax rate and the ceiling are determine based on the type of securities. The TOB tax amount is calculated based on the pro rata basis and subject to ceiling. The TOB ceiling is applied on eligible customers and not on transactions.

The following items are introduced as part of this functionality:

- The `BETOBT.TAX.CONFIG` application stores the parameter information.
- The `BETOBT.TAX.PARAMETER` is a parameter application which stores the tax related information.
- The *Eff Tax Res* field is added to the `CUSTOMER` application to capture the effective tax residence of the customer.

> **The topic related to this feature is given below:**
>
> TOB Tax

Finland Model Bank

Subsidy Loans » Interest Subsidy Loan

This functionality allows banks to add, modify, disable or remove the interest subsidy loan related templates into Temenos Transact. It also allows banks to calculate the interest subsidy - to each loan based on the selected interest subsidy - templates.

The interest subsidy loan product offered by the Finnish banks and regulated by the Finnish law in terms of loan period and interest subsidy template include a deductible rate and customer liability.

As a part of this functionality, a new external property class (XSUBSIDYLOANPRODUCT) is provided to the bank. New fields are added to the XSUBSIDYLOANPRODUCT external property class to choose the subsidy interest loan product template and to mention the template effective date so that the subsidy loan can be created with different legislation subsidy interest template values.

The following application and enquiries are introduced as part of this functionality:

- The `NORSIC.SUBSIDY.TEMPLATE` application is used to create and maintain the interest subsidy-related details which can be created based on different currencies and different dated version. It is mapped to the XSUBSIDYLOANPRODUCT property class.
- The `NORSIC.INVOICE.INTEREST.CODE` application is provided to store the loan purpose code for the State Treasury. The bank is allowed to add, delete and modify the purpose codes as part of this application.
- The `NORSIC.INTEREST.CODE.MAPPING` application is provided to map the Temenos Transact periodic index keys to the state treasury interest reference rate codes.

The topic related to this feature is given below:

Subsidy Loans

France Model Bank

PFNL (Prélèvement Forfaitaire Non-Libératoire) Tax

PFNL (Prélèvement Forfaitaire Non-Libératoire) is a tax withheld on income payments (cash interest, dividends, coupon, and issue premium in relation with principal payments) on any undermentioned instrument or product. This is a flat tax and is calculated in addition to the withholding tax (WHT) and it applies to the individuals who are fiscally residents in France. PFNL is voluntary only, therefore a mandate (customer's instruction) allowing the bank to act on his behalf is mandatory.

This functionality allows banks to calculate the PFNL French tax.

The following items are introduced as part of this functionality:

- The *Pfnl Mandate Recd* and *Eff Tax Res* fields are added to the `CUSTOMER` application to indicate whether the mandate is received or not and to capture the effective tax residence of the customer.
- The *Taxable Basis Tax Type*, *Taxable Basis Amt*, and *Tax Credit Amt* fields are added to the `ENTITLEMENT` application to capture the corresponding tax type for the taxable basis amount, amount field in the event currency, and tax credit amount based on Double Tax Treaty (DTT) rate in the event currency.
- The `FRPFNL.TAX.PARAMETER` application is used to define the calculation of taxable basis amount for each event and parameterise the appropriate events and sub asset type for which the PFNL tax is applicable.
- The `CUSTOMER,FRPFNL.INPUT` version is used to create customer records with the effective tax residence and PFNL mandate flags.
- The `CUSTOMER,FRPFNL.NAU` version is used to authorise the customer records with the effective tax residence and PFNL mandate flags.
- The FRPFNL.TAX.SIMULATOR enquiry is used to check if a portfolio Id is subjected to the PFNL tax or not.

The topic related to this feature is given below:

PFNL Tax

India Model Bank

Lending Compliance » EDPMS

This functionality allows banks to manage the export process and transactions. The Reserve Bank of India had launched a comprehensive IT based system called Export Data Processing and Monitoring System (EDPMS) for better monitoring the export of goods and software and facilitating the Authorised Dealer (AD) banks to report various returns through a single platform.

The following applications are introduced as part of the functionality:

- The `INLEND.IEC.STATUS` application stores all important configuration data and other data for the IDPMS and EDPMS functionalities.
- The `INLEND.EDPMS.RECEIPT.DOCUMENT` application stores the data of the shipping bills.
- The `INLEND.EDPMS.TF.PAYMENT.INFO` application stores the data about the received payments.
- The `INLEND.EDPMS.PAYMENT.REALIZATION` application stores the data about the received payments.
- The `INLEND.IDPMS.OTHER.BANK.FIRC` application stores the foreign inward remittance certificate (FIRC) issued by other banks.
- The `INLEND.EDPMS.ERROR.CODES` application stores the error codes as specified by the EDPMS system.

> **The topic related to this feature is given below:**
>
> Lending Compliance

Lending Compliance » Issuance of Electronic Bank Realisation Certificate (eBRC)

This functionality allows banks to create electronic bank realisation certificates

(eBRC). The system validates whether the bills are closed in the export data processing and monitoring system (EDPMS) before any eBRC is issued. The user can correct, cancel and re-issue an eBRC.

The following items are introduced as part of this functionality:

- The `INLEND.EDPMS.EBRC` application is used to capture the details required to generate eBRC for the shipping bill.
- The `INBASE.IFSC.CODE` application is used to store the Indian financial system code (IFSC) details of the branch or bank.
- The UPLOADED.EBRC enquiry is used to list the eBRC's with the error code not equal to 000.
- The FAILED.BRC enquiry is released to display all the BRC's with the *Error Code* not equal to 000 and not equal to Null.
- The CANCELLED.BRC enquiry is used to display the list of eBRC's which have the status as cancelled.
- New fields are added to the `INLEND.IMPORT.EXPORT.ATTRIBUTES` application to provide the required fields for uploading the eBRC to the directorate general of foreign trade (DGFT) server.

> **The topic related to this feature is given below:**
>
> Lending Compliance

Payments Information » FIRC Conditional Advice Generation

This functionality allows banks to issue Foreign Inward Remittance Certificates (FIRC). A FIRC acts as documentation for a foreign money transfer. It tracks the amount of the transfer in both the foreign currency and in rupees, as well as where the transfer comes from and where it ends up. Since the foreign fund transactions might be used for illegal activities like money laundering, a FIRC is a safeguard against such purposes.

For generating the FIRC advise, the *Purpose Code* plays a vital role. The

Purpose Code field defines the purpose of the payment. This purpose code defines if the FIRC advise has to be printed on secured stationery or not.

The following items are introduced as part of this functionality:

- The *Purpose Code* field is introduced in the `PP.ORDER.ENTRY` application to capture the purpose code as mandated by the Reserve Bank of India (RBI). These will be fetched from the `INBASE.PMT.PURPOSE.CODE` application.
- The INBASE.PMT.PURPOSE.CODE.DETAILS enquiry is used to list all the purpose codes, and provide the option to either edit or reverse the record.
- The INBASE.PMT.PURPOSE.CODE.NAU enquiry is used to fetch the INAU records where the versions for editing, authorisation, and deletion are added.
- Versions of the `INBASE.PMT.PURPOSE.CODE` application are released to facilitate the input, amendment, and deletion of a purpose code.
- Fields are introduced in the `INBASE.PMT.PURPOSE.CODE` application to give out further details of the purpose code based on which the Foreign Exchange Transaction Electronic Reporting System (FETERS) reporting is done.

> **The topic related to this feature is given below:**
>
> Payments Information

Israel Model Bank

Matrix Tax Server Interface

This functionality allows banks to send the Ex Event Ratio, Index Rate, Security Trust Fund and the Foreign Currency files to the Matrix Tax Server during the Close of Business (COB). The Matrix Tax Server is used to do the calculations of all Israeli taxes. Once the Matrix Tax Server returns the taxes to be charged, accounting entries are initiated to the account linked in the original transaction.

The following items are as part of this functionality:

- The `ILMATX.TXNS.EXTRACTION` application is used to store the transactions performed during the day.
- The `ILMATX.TXN.MAPPING` application stores the uniquely generated 17-digit transaction reference to be communicated to the Matrix Tax server.
- The `ILMATX.TXN.DETAILS` application holds the status details of every transaction initiated in Temenos Transact.
- The *Branch Code* field of the `ILMATX.PARAMETER` application is introduced to configure the branch code. The value provided here will be used in the extraction file.
- The *Matrix Sub Inv Code* field of the `SUB.ASSET.TYPE` application is introduced to store the Matrix sub-investment code.
- The ILMATX.EXTRACT.INPUT.FILE service is used to generate the Request Input file for security trades or transfers, update the `ILMATX.TXN.DETAILS` application and clear the content of the `ILMATX.TXNS.EXTRACTION` application once the Request Input file has been generated.

> **The topic related to this feature is given below:**
>
> Matrix Tax Server Interface

Italy Model Bank

Withholding Tax on Securities Accrued Interest

This functionality allows banks to post Italian withholding tax on accrued interest (which is calculated outside Temenos Transact) to the customer or internal account.

The following application and enquiry are introduced with this functionality:

- The `WTXSAI.TAX.ACCR.INT.PARAMETER` application is introduced to define the Italian internal tax account number, payment order product type.
- The WTXSAI.TAX.ACCR.INT.POST enquiry is introduced to list the trades, transfers, and entitlements corresponding to the Italian security domicile. It allows crediting or debiting the customer account with the tax amount received from the custodian. It also allows the authorisation of the credit or debit tax payments inputted above.

> **The topic related to this feature is given below:**
>
> Withholding Tax on Securities Accrued Interest

Luxembourg Model Bank

FDR (Fiscalité des Résidents) Tax » Tax Reimbursement on Investment Income

This functionality allows banks to calculate and withhold the Luxembourg FDR (Fiscalité des Résidents) tax. The FDR tax applies on the interest payments, coupons, redemptions, securities sales (sell trades, or transfer out in case of beneficial owner change), the income received by individuals who are fiscally domiciled in Luxembourg and is calculated in addition to the foreign withholding tax.

The tax calculation method used for coupons is based on the coupon revenue method (holding period interest). For the securities sales or transfers out, the tax calculation method is based on the accrued interest revenue method, and for redemptions, it is based on the issue discount revenue method.

The following items are introduced as part of this functionality:

- The `LUFDRT.FDR.PARAMETER` parameter application allows banks to define the criteria values on which the securities, corporate actions, and arrangements are considered and applicable for the FDR taxation.
- The LUFDRT.TAX.SIMULATOR no-file enquiry allows banks to identify if a customer is eligible or not for the FDR tax.
- The *Eff Tax Res* field is added to the `CUSTOMER` application to capture the effective tax residence of the customer.

> **The topic related to this feature is given below:**
>
> FDR (Fiscalité des Résidents) Tax

New Zealand Model Bank

Open Bank Resolution » Term Deposit

This functionality allows the statutory manager (a special type of user) to split the current balances in the deposits into available funds and frozen funds.

The Open Bank Resolution (OBR) is a stage when a statutory manager nominated by a regulator takes control of the bank in case of bank failure. OBR involves a process where the current deposit balances are split into two types of funds, for example, available funds and frozen funds.

The customer can access the available funds but the customer cannot access the frozen funds temporarily. The frozen funds are released in parts until it is fully released.

The `NZOBRS.FUNDS,NZ.FREEZE.INPUT I DEPOSITS` version is introduced to freeze a defined percentage of funds and set up frozen funds within a deposit arrangement which will be later returned to available funds. This unfreezing operation will go on until the release percentage touches 100% for all the deposit products.

The topic related to this feature is given below:

Open Bank Resolution

Saudi Arabia Model Bank

Account Infrastructure » Account Statements

This functionality allows banks to provide account statements that include all the relevant information for the transaction types. Each transaction must contain details related to the delivery channel, debit information in case of debit transactions, and credit information in case of credit transactions.

The following fields are added to the `SABASE.CUSTOMER.PARAM` application to hold the Arabic language code, or the default language code, if the Arabic language code is absent:

- *Arabic Language Code:* This field holds the language code for the Arabic language, which needs to be used for fetching values from a multilanguage field.
- *Default Language Code:* This field holds the default language code which has to be used for fetching the values from a multilanguage field when the Arabic language code is absent.

> **The topic related to this feature is given below:**
>
> Account Infrastructure

Watheeq Services

There can be some legal disputes between the customers of a bank when they will go to the court, and based on that, there will be a legal hearing so that the court will provide instructions to SAMA, being the Central Authority. SAMA will send notifications to all the banks where these customers have defaulted or they are having issues so that banks will be able to block their funds.

This functionality allows banks to ban or deny a customer from dealing with the bank and to lift all restrictions done against a customer and return the customer

info details.

The following items are introduced as part of this functionality:

- The `SAWATQ.BLOCKING.LIST` application is used to serve as a placeholder to store the blocking details of the customer. Also, new versions of this application are created for Ban, Deny and Lift services.
- The SAWATQ.BLOCKING.LIST enquiry is used to view all the blocking list records.
- The SAWATQ.BLOCKING.LIST, EXCEPTION enquiry is used to view the list of exceptions to be handled by the bank.
- The *Block Type* and *Posting Restriction Code* fields are added to the `SAWATQ.PARAMETER` application to store the type of block and the relevant posting restriction for the blocking type.

> **The topic related to this feature is given below:**
>
> Watheeq Services

Spain Model Bank

Cheques and SNCE Clearing » SNCE08 Initiations Additional Outgoing

The SNCE08 miscellaneous payment is a payment mechanism used in Spain for processing payments or collections. These are the payment instructions for all payments, which were not able to meet the standard clearing cut-off time. Every incoming SNCE08 is two-way communication in case of rejects on the recipient side.

This functionality allows banks to send and respond to the SNCE08 messages.

The following items are introduced as part of this functionality:

- The *Returns/Response*, *Transaction Type*, and *No Of Days* fields are added to the `ESCLNG.PARAMETER` application to allow the user to select whether it is a return or response message, to select the transaction type, and to provide the SLA period in days.
- The `ESCLNG.SNCE.DATA.MODEL,INWARD.ACCEPT.MANUAL` version is used to accept the inward messages.
- The `ESCLNG.SNCE.DATA.MODEL,INWARD.REJECT.MANUAL` version is used to reject the inward messages.
- The `ESCLNG.SNCE.DATA.MODEL,INWARD.RETURN.MANUAL` version is used to return the inward messages.

The topic related to this feature is given below:

Cheques and SNCE Clearing

Tunisia Model Bank

Clearing » File Processing

This functionality allows banks to register, amend and cancel Tunisian direct debit mandates through the Tunisia Clearing House (SIBTEL). It also allows banks to generate outward DD mandate files for mandate records created in the system and sending them to SIBTEL.

Mandates are classified as inward, in which case Temenos Transact receives a direct debit claim against the mandate, or outward, in which case a claim is originated from Temenos Transact to the debtor bank.

The following applications and enquiries are introduced as part of this functionality:

- The `DD.DDI,PPTNCL.MANDATE.CREATE` version allows users to create or amend outward mandates.
- The `DD.DDI,PPTNCL.MANDATE.CANCEL` version allows users to cancel outgoing mandates.
- The `DD.DDI,PPTNCL.MANDATE.INWARD` version is used to create inward mandates through OFS and also to view or delete erroneous mandate records.
- The `CMBASE.BATCH.INTRF.EXTRACT.LOGGING` application is used to log the error details captured while processing the direct debit mandates records from the inward file from SIBTEL.
- The PPTNCL.ERROR.LOG enquiry allows users to view the error details for the inward mandate file from SIBTEL.
- The PPTNCL.IHLD.MANDATES enquiry allows users to view or delete failed mandates.
- The AUTH.NEW.MANDATE enquiry displays the records from the DD.DDI$INAU file. When selecting the **Authorise** button, the `DD.DDI,PPTNCL.MANDATE.CREATE` version will be launched.
- The AUTH.CANCEL.MANDATE enquiry displays the records from the `DD.DDI$INAU` file. When selecting the **Authorise** button the `DD.DDI,PPTNCL.MANDATE.NEW.CANCEL` version will be launched.

> **The topic related to this feature is given below:**
>
> Clearing

Foreign Currency Operations » Foreign Currency Limit Management and AVA Issuance

This functionality enables banks to handle the foreign currency operations based on the currency guidelines set by the government and (CBT) Central Bank of Tunisia. The foreign currency operations are processed within the authorized limits granted by CBT and set by banks in addition to managing and tracking their usage. When a customer wishes to get a foreign currency banknote using the AVA contract, the system will check the travel allowance amount in the AVA contract.

The following applications and versions are introduced as part of this functionality:

- The TNFCOP.UNAUTHORIZED.DECLARATIONS enquiry is provided to list the unauthorised declarations and authorise them.
- The `TELLER,AVA.FCY.BUY` version is provided to capture the purchase of foreign currency banknote with an AVA contract. This version is auto-populated with the *Teller Transaction Code*.

> **The topic related to this feature is given below:**
>
> Foreign Currency Operations

Foreign Currency Operations » Incoming Transfer

The Tunisian regulations entail capturing additional regulatory information for the incoming transfers associated with a foreign trade title or business travel allowance.

This functionality allows banks to capture the regulatory information for incoming payments in foreign currency. The incoming payments in foreign currency are either associated with one or multiple titles or business travel allowance.

The following applications and enquiries are introduced as part of this functionality:

- The TNFCOP.INC.TRANSFER.REPAIR enquiry allows the user to add the regulatory information in the incoming transfers where the transaction currency is different from the credit account currency. The `PP.ORDER.ENTRY, TNFCOM.INC.TRANSFER.INP` version is linked to the **Edit** and **View** buttons of the enquiry.
- The `PP.ORDER.ENTRY, TNFCOM.INC.TRANSFER.INP` version is introduced to create incoming payments in foreign currency.

> **The topic related to this feature is given below:**
>
> Foreign Currency Operations

United States Model Bank

Retail » Statements

Periodic statements generated by financial institutions in the U.S. follow certain regulatory standards, which require them to include information regarding overdraft fees charged during the statement cycle and year-to-date. Additionally, statements also show details of checks paid to a customer during the statement cycle.

The *Usretl Sup Susp Narr* field was added to the `AC.ENTRY.PARAM` application. If it holds the value YES then the system will suppress the narrative change in the `AC.STMT.ENTRY` application when exception transactions occur.

The NSF.DETAILS tag is updated in the statement XML file with the NSF OD information, required to be presented on periodic statements in the U.S.

The CHECK.DETAILS tag is updated in the statement XML file with the checks' transaction details. The SUMMARY.INFORAMTION and ACCOUNT.TOTAL tags are also updated as part of this functionality.

The topic related to this feature is given below:

Retail

Regulations » Tax Certification Type

This functionality allows banks to use the certification type of the customer as criteria for Irs Tax reporting.

The topic related to this feature is given below:

Regulations

Retail

Retail Lending» Weighted Average Rate (WAR) for Multiple Disbursement Tranches

In banks for lending products which have disbursement in tranches, each tranche has its own interest rate. Banks prefer an option to enable the calculation of a weighted average interest type for such loans having multiple disbursements.

The Weighted Average Rate (WAR) feature of the existing Arrangement Lending (AL) module ensures that the banks can apply individual loan interest rates for each disbursement initiated within a single loan contract and calculate the WAR for the arrangement that is re-calculated every time a new disbursement is performed and also whenever a rate reset happens.

Each time a new disbursement is initiated, the system checks for its applicable rate and retrieves the Weighted Average Rate (Effective Interest), based on below formula.

$$War_1 = \frac{War_0 \times Disb_0 + Int_1 \times Disb_1}{Disb_0 + Disb_1}$$

Where:

- War_1 = Current calculated weighted average rate.
- War_0 = Previous calculated weighted average rate.
- $Disb_0$ = Previous outstanding principal (before current disbursement/repayment).
- $Disb_1$ = Current amount disbursed.

To enable this calculation, the Rate Type attribute of Interest arrangement condition has to be set to Wght Avg Rate. A periodic interest definition is required in order for the properly setup to enable the calculation of the WAR which determines the actual effective rate of the arrangement. This feature:

- Enables the calculation of a WAR considering the previous rate of the contract for the existing disbursed amount as well the new disbursement

rate for new disbursements.

- Allows the banks to offer individual rate for each disbursed amount, a single effective Weighted Average Rate and single combined schedule of all the underlying disbursements rather than multiple drawings under a Facility.

> **The topics related to this feature are given below:**
>
> Interest
>
> Weighted Average Rate
>
> Creating an Arrangement for Weighted Average Interest Rate Loan
>
> War Loan

Retail Lending » Repayment Order

In an overdue loan scenario, banks prefer to settle the tax component first before collecting the other components of the loan. This enables banks to collect the tax (either fully or partially) when there is a partial payment on the loan. Besides, banks also prefer to sort the bills generated on same date (by the bill amount).

This feature now aids both the preferences.

- Banks can now prioritise the repayment amount to settle the tax component first. To enable this feature, the *Tax Settlement* field is introduced in the Payment Rule Property Class. The allowed values of this field are:
  - **Separated** - Sequences the Tax Property in the *Property* field in Payment Rule product condition.

    > **NOTE:** This feature is not applicable for advance and principal decrease payment rule type.
  - **Prorate** - Indicates the existing behaviour and also the default value where Tax should be calculated and collected on a pro-rated method based on the repaid amount.

- The repayment order for settlement of bills raised on the same date can be based on the total outstanding amount in the bill. To enable this feature, the *Repayment Order* field is introduced in the Payment Rule Property Class. The allowed values are:
  - **Highest or Lowest** - Based on this, the repayment order is set for settlement of bills on the same date based on total bill amount namely the highest or lowest amount respectively. After a partial settlement, when there are two bills with two different outstanding payment amounts, the bills are prioritised based on the highest or lowest outstanding amount in the bill.
  - **Blank (Not set)** - The existing behaviour continues, that is, the repayment order for settlement of bills with the same date is in the order of generation of the bill (random order).

> **The topics related to this feature are given below:**
>
> Payment Rules
>
> Repayment Order

Retail Lending » Percentage Calculation for Principal-only Payments

Banks offer an amortisation plan to customers to repay a specific percentage of the Principal-only repayment over a certain period for the granted loan amount. There is a chance that the recalculation percentage can be any number greater or less than the maximum, that is, without any restriction.

To overcome this, the *Percentage Calc Type* in AA Payment Type framework has now been enhanced to calculate the percentage repayment amount for Principal-only payments and to restrict the payment percentage to the defined maximum percentage in Payment Schedule conditions. The total granted loan amount can be linearly distributed as Principal-only repayment for the percentage defined for the amortisation period as per the repayment frequency.

When a percentage is defined in the newly introduced *Max Percentage* field, the

repayment percentage of original loan principal definition is restricted to the maximum of percentage defined. Amortisation schedule is 100% but the calculation percentage is restricted to the defined percentage.

This feature is advantageous when the loan offered to the customer was an Interest Subsidy loan, where it is easier to collect the Principal-only repayments.

The topics related to this feature are given below:

AA Payment Type

Configuring Scheduling Payments

Percentage Calculation for Principal-only Payments

Product Management » Multi-Currency Account Product Line

The increase in international payments, the need for low-cost transactions denominated in different currencies are some reasons why financial institutions today offer multi-currency accounts. Besides the travelers also benefit from minimizing foreign exchange costs when engaging in local currency transactions. MCY accounts help the customers to get the most out of their money while working with multi-currency transactions.

The Multi-Currency Accounts (MCYAAR) module in AA now allows the user to hold multiple currencies under a single account structure and carry out multi-currency transactions using the same account details (beneficiary name, IBAN, account number). The following are the key features and related benefits that a Multi-Currency Account provides:

- Ability to hold, send and receive money multiple currencies under one financial account providing convenience and ease in managing multiple currencies. This also facilitates mitigating the risk with foreign exchange rate fluctuations.
- Ability to define a base currency for the multi-currency account and link sub-accounts in any currency offered by the bank

- Ability to convert the different currencies into the local (base) currency as required
- Flexibility to route credit transactions to the sub-account of same currency with that of payment currency
- Ability to report currency-wise balances or an aggregated account balance in base currency
- Flexibility to keep funds in the currency of an incoming transaction until it is convenient to convert it
- Reduced transaction costs.

The topics related to this feature are given below:

Multi-Currency Accounts

Tasks for Multi-Currency Account Creation

Tasks for Multi-Currency Account Maintenance

Tasks for Sub-Accounts

Tasks for Eligibility

Technology

Design Framework

Temenos Workbench » Defining Selection Fields

While creating an enquiry or API enquiry, banks can now define the type of selection fields as:

- **Display** – Used for displaying enquiry execution output
- **Input** – Used for user input alone
- **Both** – Used as input before enquiry execution and for displaying query results as well.

> **The topics related to this feature are given below:**
>
> Creating an Enquiry
>
> Creating an API Enquiry

Temenos Workbench » Keycloak Authentication

In addition to the existing authentication mechanism in Fabric, a new authentication service called Keycloak is now available in Temenos Workbench. Keycloak provides single sign-on authentication to the application. You can switch between the existing basic authentication service (Fabric) and Keycloak authentication service.

> **The topic related to this feature is given below:**
>
> Configuring Keycloak Authentication Service

Integration Framework

ESB Adapter - IBM Integration Bus (II) » Enabling TimeToLive Property for MQCONNECT and JMSCONNECT

TimeToLive (TTL) property is a new connection property to specify expiry time, so that, messages get discarded after the specified timeout. It is now allowed as an input at IIB outbound design time. This property can be enabled for both (JMS and MQ) types of connectivity. The expired messages cannot be consumed and are discarded in case of MQCONNECT and sent to expiry queue in case of JMSCONNECT (if expiry queue is specified).

> **The topic related to this feature is given below:**
>
> Working with the Outbound Adapter

Data Event Streaming » Adopting DES CloudEvents Schema

DES delivers all the events to topic in Avro format.

This feature allows DES to deliver events as CloudEvents as well to public stream topics. CloudEvent is a standard event format for Cloud related technologies. Customers can get the CloudEvents to make use of common

structure across different cloud providers.

The topic related to this feature is given below:

Advanced Configuration

Treasury

Money Market » Negative Interest Rate Capitalisation in Money Market

In Money Market (MM) application, interest rate capitalisation was applicable only for positive interest rates. The `MM.MONEY.MARKET` application is enhanced to support capitalisation of negative interest rates on MM Taking or Placement deal. This feature benefits the bank to capitalise interest on negative rates.

The topics related to this feature are given below:

Working with Call or Notice Placement

Working with Fixed Placement

Working with Call or Notice Taking

Working with Fixed Taking

Installation and Configuration Notes

Application Framework

Infrastructure » Customer Data Access Logging

After installing and configuring enquiry and transaction logs, `PROTOCOL` will be written along with `CUSTOMER`'s additional information.

I Technical Notes

Application Framework

Infrastructure » Customer Data Access Logging

There are no changes in local development. However, to record additional information, you need to use,

- *Protocol Info API* for local customisation
- *Customer* to provide the list of additional fields, which provide `CUSTOMER` information

Banking Framework

Structured Address Fields

The entire set of fields from *OT.ID.TYPE* to *SCH.ISSUR* for ULTIMATE.CRED and BENEFICIARY have been associated with multivalue fields in the `BENEFICIARY` application:

- A validation has been added to check that the values are entered only in the first multivalue set for the *DOB*, *BR.PRVNC, BR.CITY* and *BR.COUNTRY* fields.
- Only the first multivalue set has been mapped from `BENEFICIARY` to `STANDING.ORDER`.
- The existing APIs have been modified to include the newly added address fields.

Extensibility APIs

Java Extensibility

Category: New Enhanced Existing Deprecated

Package	Class	Method name	Description	Hook*/ API
payments	PaymentLifecycle	**getCreditAccount**	This hook enables the implementer to return the account for the credit side of the current transaction.	Hook
payments	PaymentLifecycle	**getDebitAccount**	This hook enables the implementer to return the account for the debit side of the current transaction.	Hook
payments	PaymentLifecycle	**validateCreditParty**	This hook enables the implementer to validate the party on the credit side of the current transaction.	Hook
payments	PaymentLifecycle	**validateDebitParty**	This hook enables the implementer to validate the party on the debit side of the current transaction.	Hook
payments	PaymentLifecycle	**getPaymentDate**	This hook enables the implementer to return a PaymentDate object which impacts when the payment is processed.	Hook
payments	PaymentLifecycle	**updateProcessSequence**	This hook enables the implementer to update the processing sequence and decide whether to include the codeword for special processing or not.	Hook
payments	PaymentLifecycle	**skipMessage**	This hook enables the developer to decide whether to skip the generation of the swift output message.	Hook

Package	Class	Method name	Description	Hook*/ API
hook	ComponentApiHook	**getCreditAccount**	Deprecated. Use PaymentLifecycle.getCreditAccount instead	Hook
hook	ComponentApiHook	**getDebitAccount**	Deprecated. Use PaymentLifecycle.getCreditAccount instead	Hook
hook	ComponentApiHook	**validateCreditParty**	Deprecated. Use PaymentLifecycle. validateCreditParty instead	Hook
hook	ComponentApiHook	**validateDebitParty**	Deprecated. Use PaymentLifecycle. validateDebitParty instead	Hook
hook	ComponentApiHook	**getCodewordFlag**	Deprecated. Use PaymentLifecycle.updateProcessSequence instead	Hook
hook	ComponentApiHook	**setCalculatedDate**	Deprecated. Use PaymentLifecycle.getPaymentDate instead	Hook
payments	DataAccess	**getPaymentRecord**	Returns the payment record from the table of the company where the user is currently logged into	API
payments	AccountingEntry	**setAccountId**	Enables the implementer to perform the check digit calculation and also updates modified accountId into the locking table.	Hook
system	ServiceLifecycle	**getSwiftRequests**	This interface enables the implementer to return a number of swift requests for processing by the system.	Hook
system	Session	**Initialize**	This interface enables the implementer to initialise a	Hook

Package	Class	Method name	Description	Hook*/ API
			session by preloading cached data to improve the processing time of subsequent transactions on the session.	
system	Session	loadRecord	This interface enables the implementer to initialise a session by preloading cached data related to the given table and record id to improve the processing time of subsequent transactions on the session.	Hook
system	Session	getCachedRecord	Returns a cached copy of the record from the live table of the company the user is currently logged into.	API
system	Session	getCachedLookupValues	Returns a cached copy of the lookup values for the given lookup key.	API
system	Session	clearLookupCache	Clears all lookup values from the cache.	API
system	Session	setNextVersion	Sets the next VERSION to be launched automatically at the conclusion of the current event. Field values in the next version can be set by providing values in the record structure parameter.	API
system	RecordLifecycle	updateLookupTable	Deprecated use system.RecordLifecycle.getLookupRecordAmendments instead.	Hook
system	RecordLifecycle	getLookupRecordAmendments	This interface enables the implementer to amend multiple lookup table records by adding and removing their related ids.	Hook

Package	Class	Method name	Description	Hook*/ API
*Hooks are placeholders in Transact where routines can be attached to an application. For example, version, enquiry, delivery and so on.				