What's New in

Temenos Transact

October 2021

Information in this document is subject to change without notice.

No part of this document may be reproduced or transmitted in any form or by any means, for any purpose, without the express written permission of TEMENOS HEADQUARTERS SA.

© 2021 Temenos Headquarters SA - all rights reserved.

Table of Contents

|Release Highlights

Application Framework

System Core » Standards and Restrictions in SS2021

Restrictions in Creating and Amending Records

With the emerging SaaS/Cloud based solutions certain restrictions are introduced to restrict the SaaS implementation clients from attaching any new local customized routines in the various user exit fields that permit it. Transact is now enhanced to restrict user exits governed by IN2HOOK and IN2HOOKOTHERS field types.

Restrictions in Adding and Modifying Routines

Applications can now individually impose restrictions for creating new records and amending existing records, providing a greater control on the customization for the SaaS clients.

> **The topic related to this feature is given below:**
>
> Standards and Restriction in SS2021

Banking Framework

Direct Debit » Capturing Bank Address for Direct Debit Initiation Messages

Temenos Transact now allows banks to set up the address for the pain.008 messages in the main `DD.PARAMETER` application. While handing off the DD collection requests, the Direct Debit (DD) module uses this address to send the pain.008 messages. If the address is not set in the main `DD.PARAMETER` application, the Delivery module sends the pain.008 messages to the ISOMX delivery address of the customer. If the customer does not have an ISOMX delivery address, the Delivery module puts the message in repair status. This functionality reduces the operational overhead to create an ISOMX delivery address for every customer.

> **The topics related to this feature are given below:**
>
> Capturing Bank Address for Direct Debit Initiation Messages
>
> Sending DD Collection to an External Payment System

Delivery » SWIFT Multi-format MT MX Messages

The FINPlus service used by CBRP+ allows the banks to exchange payment and payment-related MX messages. Banks have the option to choose between MT and MX formats until November 2025. For CBPR+, SWIFT provides a translation from MT to MX and vice versa, either at the Transaction Management Platform or as part of the FINPlus service (from CAMT to MT). SWIFT allows banks to decide if they want to receive MT or MX messages for each of their BIC8 or BIC11. The bank can refine this decision based on the message type and/or currency.

Until November 2025, the SWIFT MX messages sent by financial institutions that have already adopted ISO20022 standards will be translated by SWIFT to MT before being sent to receiving institutions that still use MT formats. In such cases, SWIFT sends a multi-format message to the receiving institutions representing the original ISO20022 message, which embeds the MT message as a comment. The receiving institutions can configure their SWIFT interface to send either the original message received from SWIFT or just the translated MT to their business application.

The Delivery Transformation Layer can receive SWIFT multi-format messages and, depending on the parameterisation set up by the bank, route either the MX message (the default option) or the MT message to Temenos Payments Hub (TPH) or the Delivery module. In both cases, the original MX message is stored in the Delivery module.

This functionality allows the implementations to gradually adopt new standards.

> **The topics related to this feature are given below:**
>
> SWIFT Multi-format MT MX Messages
>
> Configuring SWIFT Multi-format MT MX Messages
>
> Working with SWIFT Multi-format MT MX Messages

Collateral » Margin Valuation through Credit Policy Structure

Temenos Transact provides banks an option to define the credit policy system. The credit policy includes rules for defining the lending ratio (margin rate) based on various factors like tenure of the asset, currency of the asset and asset size. This functionality offers the following benefits:

- The credit policy structure provides flexibility to the banks to mitigate risk driven by various market factors.
- The credit policy structure allows definition of rules based on any attribute like asset class, sub asset class, currency, issuer, market, sector, risk

country, index, bond type, credit ratings, remaining tenor, fund type, fund size and issue size.

- The credit policy structure supports multiple set of rules, which can be linked to one or more credit policy systems.
- The bank can apply certain caps that act as a cushion to prevent the asset value from depreciating further in case of volatility and prevents risk in case of cross-currency.
- The concentration pool framework allows banks to benefit from the advantage of the liquid assets in the portfolio.

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

The topic related to this feature is given below:

Margin Valuation through Credit Policy Structure

Corporate

Facility » Pricing Grid

Pricing grid property class has been enhanced and the Usage Percentage field (of Interest property class) is added to `AA.TARGET.ELEMENTS` as one of the target elements.

The bank wants to charge fees on the facility commitment balances to be a fraction of the interest margins on the loans, that is commitment fee for facility is equal to the percentage of current interest margins on drawings under the facility.

If a bank charges margin to the customers based on their credit rating, then the commitment fee charged for the loans should be 40% of the interest margin applied on the loans. If the ratings of the customer changes in the future, the respective margin and the commitment fees is updated accordingly.

The enhancement allows the bank to charge commitment fees as a fraction of the margin rate with the help of the Pricing Grid property class.

> **The topic related to this feature is given below:**
>
> Commitment Fee as Percentage Margin

Retail Lending » Risk Participation for Bilateral Loans

Corporate Loans generally come with high risks associated with it as the amount involved is usually big and also due to the borrower’s dependency on cash flows for repayment of the loan. Corporate Loan such as bilateral loans involves high risk. A loan is associated with a credit risk as there is a chance for the borrower to default on the repayments. The Banks need to ensure that the Loans are paid back by the borrower without fail.

The banks mitigate the risk exposure in a loan by selling the risk exposure to other Banks which helps them to diversify the Risk Portfolio. The Banks which share this Risk Exposure from the owning bank are called Risk Participants.

In this enhancement, Participant Property Class is used to define the Risk Participants while creating the contract. The enhancement allows the Banks to add Risk Participants while creating Bilateral Contracts. Whenever the borrower commitment changes due to an Activity, based on the activity and the Risk participant type (Fixed/Variable) the Risk Participants share also changes.

The Corporate Lending functionality supports the bank creating Bilateral contracts with Risk Participants so that the funding Bank can share the Risk exposure with other banks (Risk participants) thereby reducing the burden of entire risk exposure on one bank.

> **The topic related to this feature is given below:**
>
> Risk Participation for Bilateral Loans
>
> Risk Participation for Bilateral Loans-Facility

Private Wealth

Corporate Actions » Omnibus Level Generation of MT565 for Short Events

For some corporate events, customers elect several times before the cut-off date. The customers may also elect for a specific quantity and leave the balance unelected. This process is handled through the multi-election setup. However, in this setup, the corporate action instruction (MT565) was generated for each customer election.

The Corporate Actions (CA) module already supports multi-election, now the SWIFT message for CA instruction (MT565) is enhanced to support generation at the omnibus level. This means that for short events, MT565 is not required to be sent for each entitlement, but at an aggregate level.

The topics related to this feature are given below:

Configuring MT565 Generation in Multi-Election

Generating MT565 Manually

Tasks - Generating MT565 Manually

Lombard Lending » Margin Lending

The OV(Online Valuation) module is now enhanced to support Margin Lending or part pledging of portfolio holdings, where a chosen quantity of securities or percentage positions or portfolio can be pledged to avail Margin Loans. The following are the new features:

- A new front-end table (`SC.FACILITY.APPLN`) is available to capture the loan request from a customer. It updates the type of loan requested, responses to eligibility questionnaire, details of documents received and the details of collateral being pledged.

- Once the loan is sanctioned, the system creates a collateral automatically. Based on the details input in `SC.FACILITY.APPLN` , a manual workflow is launched to create a limit, link the limit to the collateral and to create a loan. Then, the loan can be used to purchase shares through the `SEC.OPEN.ORDER` application.
- Also, the system does an automatic valuation of the collateral and checks the utilisation of limit during COB and online.

The complete workflow of margin lending starting with an application from the customer to utilisation of the loan allows the customers to avail multiple loans.

> **The topics related to this feature are given below:**
>
> Margin Lending
>
> Creating Credit Facility
>
> Amending Facility
>
> Approving Facility
>
> Listing Sanctioned Facility and Creating Margin Loan

Regional Banking Solutions

Australia Model Bank

Withholding Tax and TFN Number » Dual Listed Securities

Dual listed securities are securities that are listed on Australia as well as another country. Such securities have a different type of tax treatment and so any income arising out of events such as cash, dividend, bonuses and stock split is taxed differently. The dual listed instruments with respect to this development are those that are listed and traded on ASX, but also have a listing elsewhere.

This functionality allows banks to calculate the Withholding Tax for the dual listed security and for stapled securities.

The following fields have been introduced as part of this functionality:

- The *Doc Country* and *Treaty Rate* (to indicate if the *Treaty Rate* is applicable) fields have been added to the `SEC.ACC.MASTER` application which allow capturing the documents provided by the customer that determine the Withholding Tax status.
- The *Fito Applicable* and *Fito Component* fields for Foreign Income Tax Offset (FITO) have been added to the `AUWHTX.IDC.YEAR.END.TAX.PROFILE` and the `AUWHTX.YEAR.END.COMPONENT.VALUES` applications. The existing framework for other year end components is entirely dependent on the same percentage being applied across entitlements for an event.

> **The topic related to this feature is given below:**
>
> Withholding Tax and Trust Income Distributions

Finland Model Bank

Incoming Payment Processing

This functionality allows financial institutions to process incoming payments where the company cash account is maintained in another bank. The financial institution will receive incoming camt.54 and camt.53 payment messages and will process these (for fees, repayment, prepayment, payoff) based on the Finnish reference number. The INCMMS module is required to receive the camt.54 and camt.53 payment messages, and it runs as a microservice in its own stack.

The following items have been introduced as part of this functionality:

- The `FIIPMT.INCOMING.PAYMENT.PARAM` application allows users to parse the creditor reference number and map the payment order initiation type which needs to be used to initiate the settlement.
- The `FIIMPT.INCOMING.PAYMENT.FILE` staging application has been introduced to act as the master log file which will record every incoming payment that is sent by the bank and then processed by Temenos Transact for making the settlements.
- The `FIIPMT.PAYMENT.INFO` application has been introduced to store the status of each payment under an entry in the staging application. The `FIIPMT.PAYMENT.INFO,VIEW` version can be used to view the existing records from the application.
- The FIIPMT.INCOMING.PAYMENT.INFO.DETAILS enquiry has been provided to view the details of the records from the `FIIMPT.INCOMING.PAYMENT.FILE` staging application.
- The FIIPMT.PAYMENT.INFO.DETAILS enquiry is provided for the `FIIPMT.PAYMENT.INFO` application to list all the records that are in a specific status.

The topic related to this feature is given below:

Incoming Payment Processing

India Model Bank

Lending Compliance » Loan Validations

The priority sector lending is an important role given by the Reserve Bank of India (RBI) to the banks for providing a specified portion of the bank lending to few specific sectors like agriculture and allied activities, micro and small enterprises, individuals, students for education, and other low-income groups and weaker sections. This is essentially meant for the all-around development of the economy.

Every bank in India is given targets and sub-targets for lending under the priority sector. The priority sector lending comprises of different categories such as agriculture, micro enterprises, education, housing, social infrastructure.

This functionality allows banks to not charge penal interest for loans to borrowers under the priority sector up to 25,000 RS.

The *Priority Sector Class* field has been added to the XPSA.ADVACNES soft property class which will list the classifications of the priority sector. Also, a new menu has been created with the new products.

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

Israel Model Bank

Derivatives Strike Price Management » Derivatives Premium or Strike Price Quoting

In Israel, all the derivatives, like the premium payments and the settlements, are traded in NIS (New Israeli Shekel), however for some of the derivatives, the strike price is quoted in Agorot (1/100th of NIS).

This functionality allows for the automatic assignment of options to customers who are on the primary side of a trade and have sold the options concerned.

The following items were introduced as part of this functionality:

- The *System Exercise Ag* and the *System Expiry Ag* fields were added to the `DX.CONTRACT.MASTER` application to enable the system expiry or system exercise configurations.
- The `DX.CO.ASSIGN.AUTO,ILDXPR.INPUT` (Auto Assign Options) version has been created to allow the automatic assignment of options to customers who are on the primary side of a trade and have sold the options concerned.

> **The topic related to this feature is given below:**
>
> Derivatives Strike Price Management

Fees Optimisation » Non-Execution and Minimum Fee

This functionality allows banks to calculate the execution fee (min max fees), and non-execution fees. The fees will be fairly calculated so that the customer won’t be overcharged and the bank won’t be undercharged.

The following items have been introduced as part of this functionality:

- The ILFEES.PARAMETER application allows users to indicate the execution and non-execution fees.
- New fields have been added to the FT.COMMISSION.TYPE application to store the currency in which the minimum and maximum fees are specified and to store the minimum and maximum fee amounts applicable for the commission. These fields are expected to be configured to ensure that a boundary is set for the min max fees (execution fee).

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

New Zealand Model Bank

Deposits » Calculate the Break-Cost Interest on Full and Partial Breaks

This functionality allows banks to calculate the break costs on full and partial breaks when a customer requests for an early withdrawal in term deposits.

The following items have been introduced as part of this functionality:

- The `NZDEPO.PARAMETER` parameter application has been introduced to define the bank business rules based on which the term deposit break will be applied. A parameter version (`NZDEPO.PARAMETER,INPUT`) and its authorisation version (`NZDEPO.PARAMETER,AUTH`) have also been introduced.
- The `NZDEPO.BREAK.DETAILS` application has been introduced to store the multiple break details of a deposit arrangement. The `NZDEPO.BREAK.DETAILS,INPUT` version allows users to input the term deposit break details.
- The NZDEPO.TD.BREAK.REQUEST.UNAUTH enquiry allows users to view, delete or authorise records created in the `NZDEPO.BREAK.DETAILS` application.
- The NZDEPO.TD.BREAK.REQUEST enquiry has been created to display the break details of a deposit arrangement.

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

Spain Model Bank

Teller » Non-Customer Cash Transactions

Non-customers can perform cash transactions like cash deposits and cash withdrawals (through bearer cheque) to or from the bank customer's accounts at the bank's branches. During the withdrawal with bearer cheque, if the amount exceeds the threshold, then the non-customer's *Legal Id* will be verified and if the amount is within the threshold, then the *Legal Id* is non-mandatory.

This functionality allows banks to capture the non-customer information for all cash transactions and a report will be generated to display the date-wise transactions performed by the non-customers.

The following fields have been introduced as part of this functionality:

- The *Noncust Withdrawal Threshold* field has been added to the `ESTELL.PARAMETER` application to store the threshold that will be defined for the cash withdrawal.
- The *Payee Name* field has been added to the `TELLER,LCY.CASHCHQ.ACCT.NON.CUSTOMER`, `TELLER,FCY.CASHCHQ.FCY.ACCT.NON.CUSTOMER` and `TELLER,LCY.CASHCHQ.FCY.ACCT.NON.CUSTOMER` versions to identify a bearer cheque for cash withdrawals by non-customer.

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

Tunisia Model Bank

Clearing » SIBTEL Inward Cheque Clearing

Société Interbancaire de Télécompensation (SIBTEL) is the Clearing House in Tunisia and has introduced the e-clearing system. The e-clearing system is part of the modernization of the Tunisian banking system. It aims to reduce recovery times by means of a reliable, secure, efficient, and fast electronic clearing.

This functionality allows banks to process transfers, withdrawals, cheques, and bills of exchange using the e-clearing system.

The following items have been introduced as part of this functionality:

- The `PPTNCL.CHEQUE.ADDNL.DETAILS` application is used to store the blocked amount details of rejected cheques during the generation of file 82.
- The *Amt Block 84* field has been added to the `PP.CLEARING.RETURNCODE` application to allow users to select the Yes or No option for the block to be created for the transaction amount (cheque amount) using the `AC.LOCKED.EVENT` application.
- The PPTNCL.GENERATE.FILE81.MSG service has been created to identify the corresponding POR.CLG.IN.SETTLEMENT.PENDING record. The service will be triggered at the reject cut-off (09:00 PM) and pick all the cheque transactions.

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

United States Model Bank

ACH Framework » ACH Reg E Statements

Regulation E requires financial institutions to capture Reg E ACH transaction details, addenda record information, and display incoming ACH credit or debit transactions on customers' periodic statements.

This functionality allows for the display of the ACH return code description in the customer’s statement for incoming ACH returned transactions. The predetermined ACH transaction, addenda details are captured from incoming ACH NACHA files, and dynamically stored into the *Narratives* field from the `ACH.ENTRIES` application based on the SEC codes.

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

Retail » Statements

Financial Institutions include their branch contact details on the statements provided to customers. In case Financial Institutions don't use pre-printed paper for printing statements or if the branch’s address is not updated on the statements through a third party ancillary system then the information about the Financial Institution’s name and address will be fed from Temenos Transact. The xml statement files will be updated with the Financial Institution’s name, address, phone and web details as defined in Temenos Transact.

This functionality allows banks to generate the statement file with details captured in the `USCORE.COMPANY.DETAILS` application, including the company and branch information.

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

Retail

Arrangement Architecture » Combined Closure Triggered at Multi-Currency Account Level

An MCY arrangement can be closed along with its underlying sub-accounts by initiating closure at the MCA level. In such a closure, the system automatically triggers the closure of the underlying sub-accounts as well.

To perform combined closure of the MCY arrangement along with its underlying sub-accounts the user can:

- Simulate the combined closure of the MCY arrangement along with its underlying sub-accounts and execute the simulation.
- Perform direct live closure of the MCY arrangement along with its underlying sub-accounts.

> **The topics related to this feature are given below:**
>
> Combined Closure
>
> Multi-Currency Combined Closure

Arrangement Architecture and Retail Lending » Advance Repayments in Loan

Banks can now settle the bills in advance for all payment types except the routine and transaction amount-based payment types. During an advance payment, the system settles the bill as-is. Therefore, the customer does not receive any benefit from the principal or interest component paid in advance. During an advance settlement,

- The system parks the funds to the new ADV<ACCOUNT> balance type and triggers the Forward Apply Payment activities with value date as the actual bill due date . This is achieved by setting the *Advance Settle Date* field as Due in the Payment rule type of the advance payment rule condition.

- The system settles the principal and interest components of the bill as-is effective on the actual bill due date.
  - It settles the bill fully/partially based on the amount paid in advance and updates the bill status as SETTLED/ADVANCED respectively.
- The system updates the General Ledger (GL) balance only on the actual bill due date.

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

> **The topics related to this feature are given below:**
>
> Advance Settle Date
>
> Advance Payment with no Customer Benefit-Configuration
>
> Advance Payment with no Customer Benefit-Working with
>
> Advance Payment with no Customer Benefit
>
> Advance Settlement of Bills

Arrangement Architecture » Charges in MCY Arrangement

It is now possible to configure activity charges and scheduled charges in MCY arrangements. A flat amount can be defined as credit or debit charge and capitalised to the settlement account.

The Generate Charges Activity can be used to collect or pay charges in non-financial arrangements. The charge information is passed to the settlement account using *Context Name-Value* fields and capitalised in the liquidation account.

The topic related to this feature is given below:

Charges in MCY Arrangement

Technology

Design Framework

Temenos Workbench V1 (UXP) » Configuring Keycloak Authentication Service

Temenos Packager is now enhanced with Keycloak authentication mechanism, which is a standard authentication service implemented across all Temenos products.

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

Integration Framework

Integration Framework Designer » Maintaining a Stable Container Name in Integration Events

Integration Event holds data in the XML format. This data is formatted with the same level of field associations in Temenos Transact. The associated fields have a container name with a suffix group, subgroup or textGroup. These names change based on field additions to the integration flow and is a problem when writing XPath expressions to process the event XML.

Integration Framework event is now enhanced to capture the group, subgroup and textgroup container names after the first publish and uses the same name when a flow is republished, ensuring the container names are not changed when fields are added to the flow or removed from the flow. This facilitates the customers using integration events not to change their stylesheets or any mapping logic for each republish of the integration flow.

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

Interaction Framework

UXP Browser » Navigation Guide

The UXP Browser is enhanced to sort the enquiry columns by clicking the header of the enquiry table. The enquiry columns can be sorted on the same page by using the sort option. The user can view the sort option by clicking the column header and sort the data in ascending or descending order.

> **The topic related to this feature is given below:**
>
> Sorting of Enquiries

IRIS R18 » Generation and Deployment of API Artefacts from Config Microservices

In IRIS R18, workbench is designed using the Kony Visualiser that downloads the generated artefacts as a zip file.

Workbench now allows to download the generated artefacts to config microservices (if configured) instead of downloading them to local file system. IRIS has the capability to deploy the services and swagger files from the config microservice. The benefits of this functionality are:

- Ability to store the generated artefacts from workbench to config microservice.
- Ability to deploy the available services in config microservices.
- Ability to get the swagger or inventory json from config microservices.

> **The topic related to this feature is given below:**
>
> Generation and Deployment of API Artefacts from Config Microservices

IRIS R18 » Regex Engine

IRIS supports the regex validation of the API request payload based on Regex configuration stored in the Config microservices.

Now, IRIS validates the request payload value against the Regex configured in the RegexConfiguration record in Config microservices. If the validation is successful, IRIS sends the transaction request to Transact. If the validation fails, Regex Engine returns the decision object which contains the error codes and error details of the validation failure. IRIS then converts the transaction call into an Validate call and sends it to Transact, and the result of both the Transact response and Regex Engine response are sent to the client as a response with 400 status code. The API payload validation performed at the IRIS level is a benefit.

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

IRIS R18 » Rules Engine Support for UXPB

Mapping of the UXPB operation against rules is now standardized similar to

APIs where rules should be mapped against the operation ID instead of vice versa.

Users can now define their own rules against a UXPB operation. This enables rules validation at UXPB before performing any operation if rules are applicable to that specific resource. Users can also build and deploy their external rules and restrict the UXPB operation based on this rules validation.

> **The topic related to this feature is given below:**
>
> Rules Engine Support for UXPB

Platform Framework

Adapter Microservice » Deploying Adapter Microservice on Azure Kubernetes Service (AKS) and On-Premise

Adapter Framework provides a standard feature to Temenos Enterprise applications, to interface with external application that do not support event-based integration. A set of adapters are available in a template format, which can be used to design new interfaces. This feature supports the deployment and configuration of Adapter Microservice on Azure Kubernetes Service and On-premises environments.

> **The topic related to this feature is given below:**
>
> Deploying Adapter Microservice on Azure Kubernetes Service and On-Premise

|Technical Notes

Banking Framework

Accounts » Accounts – System-wide Jobs for COB Efficiency

The following jobs have been modified to run as TSA.SERVICE.COB at application stage rather than being run within the COB process:

- CM.EOD.LIQ.TO.HIS
- EB.LIMIT.TXNS.INSPECT
- REBUILD.ACCR.REV
- AC.ACCT.OVERDRAWN.CUST

Few system-wide jobs have been merged into composite job as below:

The EOD.ACCT.OVERDRAWN composite job has been added that executes the following jobs as actions for the account as applicable. These composite jobs select the records from `ACCOUNT` and execute each action only when the account is applicable to be processed under respective action:

- AC.ACCT.OVERDRAWN.PRE
- AC.ACCT.OVERDRAWN
- EOD.OD.ACCT.ACTIVITY

Collateral » Margin Valuation through Credit Policy Structure

The `CO.VALUATION.PARAMETER` application has been moved from the Collateral (CO) module to the Margin Valuation (MV) module. Any local processing referring `CO.VALUATION.PARAMETER` from the CO module (in case of componentised format) has to be changed to access it from the MV module.

Private Wealth

Securities » Fiduciary– Application-wide Jobs for COB Efficiency

Various application-wide Fiduciary (FD) jobs have been merged to run as a single master job. The FD.EOD.ORDER.CUST master job selects and processes the records from `FD.ORDER.CUST`. The following jobs are merged and executed as actions under the FD.EOD.ORDER.CUST job.

- FD.COMPLETE.AUTHORISATION
- FD.EOD.REPLACE.ORDER
- FD.UPDATE.GROUP.PAYMENT
- FD.EOD.NOTICE.SCHEDULES
- FD.EOD.LIQ.TO.HIS

The following list of jobs have been defined as COB Scheduler (TSA.SERVICES.COB) under FD.EOD.CYCLE.DATA batch:

- FD.EOD.CYCLE.GROUP.PAYMENT
- FD.EOD.CYCLE.GROUP.PAYMT.POST

The FD.EOD.STATIC.CHANGES.POST job has been removed from FD.END.OF.DAY and the logic has been merged within FD.EOD.CYCLE.GROUP.PAYMT.POST.

> **NOTE:** Each action is performed for the records, only when the criteria for the respective action is fulfilled.

Retail

Arrangement Architecture and Retail Lending » Advance Repayments in Loan

Read the AC.EVENTS section for more information on the events to be configured for the Advance payment using the *Advance Settle Date* field setup.

Trade Finance

Letter of Credit » Application-wide and Start of Day LC Jobs for COB Efficiency

Merge the jobs which have a common table select to reduce COB timings. The DR.EOD.PROCESS master job selects and processes the records from the DRAWINGS application. The following jobs are merged and executed as actions under the DR.EOD.PROCESS job.

- LC.EOD.PERFORM.DISCOUNT
- LC.EOD.PROCESS.SCHEDULES
- LC.EOD.ACCRUAL
- LC.MATURE.USANCE
- LC.EOD.PROCESS.CONSOL

The LC.EOD.PROCESS master job selects and processes the records from the LETTER.OF.CREDIT application. The following jobs are merged and executed as actions under the LC.EOD.PROCESS job.

- LC.AMORTISATION
- LC.EOD.PROCESS.SCHEDULES
- LC.EOD.ACCRUAL
- LC.EOD.REVOL.ACTIVITY
- LC.MATURE
- LC.EOD.REVOL.UTL.PROCESS
- LC.EOD.PROCESS.CONSOL
- LC.CLOSING.LCS

> **NOTE:** Certain jobs are shared between DR.EOD.PROCESS and LC.EOD.PROCESS, as the jobs process the records of both DRAWINGS and LETTER.OF.CREDIT.

The DR.SOD.PROCESS master job selects and processes the records from the DRAWINGS application. The following jobs are merged and executed as actions under the DR.SOD.PROCESS job.

- LC.SOD.PROCESS.SCHEDULES
- LC.SOD.ACCRUAL
- LC.MATURE.USANCE.SOD
- LC.START.OF.DAY

The LC.SOD.PROCESS master job selects and processes the records from the `LETTER.OF.CREDIT` application. The following jobs are merged and executed as actions under the LC.EOD.PROCESS job.

- LC.SOD.PROCESS.SCHEDULES
- LC.SOD.ACCRUAL
- LC.MATURE.SOD

> **NOTE:** Certain jobs are shared between DR.SOD.PROCESS and LC.SOD.PROCESS, as the jobs process the records of both `DRAWINGS` and `LETTER.OF.CREDIT`.

Factoring and Forfaiting » Application-wide and Start of Day BL Jobs for COB Efficiency

Merge the jobs which have a common table select to reduce COB timings. The BL.EOD.PROCESS master job selects and processes the records from the `BL.BILL` application. The following jobs are merged and executed as actions under the BL.EOD.PROCESS job.

- BL.EOD.AMORTIZATION
- BL.EOD.MATURITY
- BL.EOD.DELIVERY
- BL.EOD.MAT.TO.HIS

The BR.EOD.PROCESS master job selects and processes the records from the `BL.REGISTER` application. The following jobs are merged and executed as actions under the BR.EOD.PROCESS job.

- BL.REV.SUSP.INTEREST
- BL.EOD.DELIVERY

> **NOTE:** The BL.EOD.DELIVERY job is shared between BL.EOD.PROCESS and BR.EOD.PROCESS, as the job processes the records of both `BL.BILL` and `BL.REGISTER`.

The BL.SOD.PROCESS master job selects and processes the records from the `BL.BILL` application. The following jobs are merged and executed as actions under the BL.SOD.PROCESS job.

- BL.SOD.AMORTIZATION
- BL.SOD.MATURITY

|Extensibility APIs

Java Extensibility

Category: 🟩 New 🟨 Enhanced ⬛ Existing 🟥 Deprecated

Package	Class	Method name	Description	Hook*/ API
system	`ServiceLifecycle`	**`getTableCriteria`**	Enables the implementer to return a TableCriteria that is used to control the selection of ids from a table.	Hook
system	`DataFormattingEngine`	**`getHeaderData`**	Enables the implementer to return a string that is used as the header for the extracted file.	Hook
system	`DataFormattingEngine`	**`getTrailerData`**	Enables the implementer to return a string that is used as the trailer for the extracted file.	Hook
party	`Account`	**`lockContractBalancesRecord`**	Returns and locks the contractBalancesRecord for the account object, for the duration of the current transaction. This method is to be used only within a transaction boundary and throws an exception when called outside of a transaction boundary. The lock is released at the end of the transaction. If the account is for a syndicated or club loan, multi-currency processing occurs.	API
party	`Account`	**`getContractBalancesRecord`**	Returns the contractBalancesRecord for the account object.	API

Package	Class	Method name	Description	Hook*/ API
party	`Account`	**`getAvailableBalance`**	Returns the usable balance of an account incorporating the balance type, limit and locked amounts. This method supports High Volume Transaction (HVT) accounts.	API
contract	`Account`	**`convertReportFieldValue`**	Enables the implementer to return a converted field value for the transaction report. This interface is invoked from the authorise method of enabled applications within the transaction lifecycle. It is also invoked from the TX.TXN.BASE.UPDATE method at the local processing stage of the transaction report.	Hook
**Hooks are placeholders in Transact where routines can be attached to an application. For example, version, enquiry, delivery and so on.*				