

| Screen Hooks

Transact Extensibility for Java (English)

2020 Q1



©2020 Temenos Headquarters SA - all rights reserved.

| Copyright Notice

©2020 Temenos Headquarters SA - all rights reserved.

Warning: This document is protected by copyright law and international treaties. Unauthorised reproduction of this document, or any portion of it, may result in severe and criminal penalties, and will be prosecuted to the maximum extent possible under law.

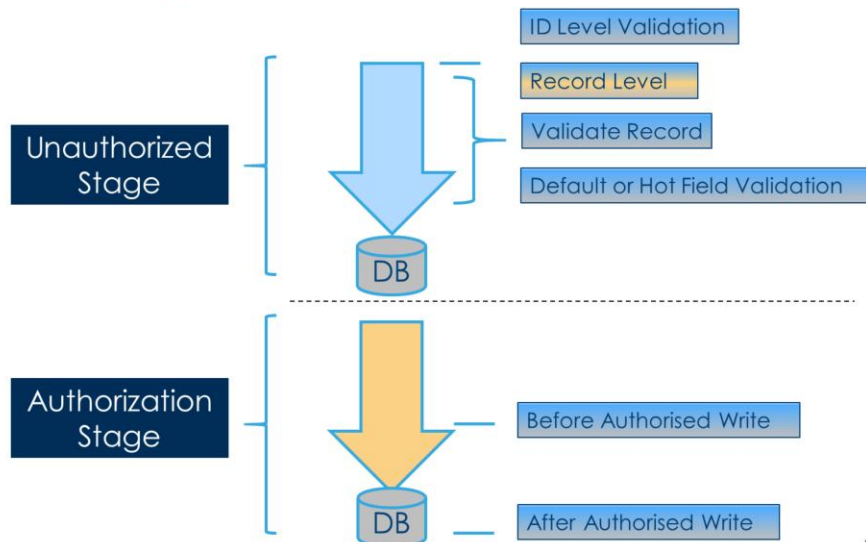
©2020 Temenos Headquarters SA - all rights reserved.



| Lesson Overview



Screen or Application Exit Points



©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS
Learning Community

All applications in T24 are designed with extensibility as a feature. T24 allows creation of versions and versions are provided with exit points - points where a user defined routine can be attached/invoked. The application exits from the usual flow, does what the custom routine says, and then depending on the flow comes back to its original flow again. Sometimes called as trigger points.

Code can be written in a version to achieve

Validations based on the record ID – Further checks can be performed on the record ID and the request processed or denied.

Record level – The CheckRec routine can be used to check values in the record before it is displayed to the user or default values before the record is displayed to the user. Auto Field routines are used to default/modify values in field before the record is displayed to the user

Field Validations - Additional field validations can be performed using validation hooks. Existing validations cannot be changed.

Record Validations - Additional record level validations can be performed and overrides can be generated. These validations are written in Input routines

Processing after commit of a record - Information can be written onto other files using After Unauth routines

Checks during authorization - Additional checks can be performed before

authorization using Before auth routines. New files can be written using authorisation routines.

Fields in VERSION to which hooks can be attached

Version Field	Java Method	Description
ID.RTN	checkId	hook to validate Record ID
CHECK.REC.ROUTINE	defaultFieldValues	record level hook to check values or default values in the record before the record is displayed to the user
AUT.NEW.CONTENT	defaultFieldValues	field level hook to default/modify values in field before the record is displayed to the user
INPUT.RTN	validateRecord	record level hook to perform additional validations
BEFORE.AUTH.RTN	updateCoreRecord or updateLookupTable	Invoked just prior to the final update of files at the authorized stage of a transaction
AUTH.ROUTINE	postUpdateRequest	invoked after the final update of files at the authorized stage of a transaction

©2020 Temenos Headquarters SA - all rights reserved.



ID.RTN – hook to validate ID

CHECK.REC.RTN – record level hook to check values or default values in the record before the record is displayed to the user

AUT.NEW.CONTENT – field level hook to default/modify values in field before the record is displayed to the user

VALIDATION.RTN – field level hook to validate data in field

INPUT.RTN – record level hook to perform additional validations

DEFAULT.ROUTINE – Invoked on Record Validation and Hot field Validate

BEFORE.AUTH.RTN - Invoked just prior to the final update of files at the authorized stage of a transaction

AUTH.ROUTINE – invoked after the final update of files at the authorized stage of a transaction

| Field/Record Validation – validateRecord()

- This interface enables the implementer to return a validationResponse whose errors and enrichments will be processed at this stage.
- This interface is invoked when the user validate a record
- The EB.API hook used by this interface is VERSION.INPUT.ROUTINE.HOOK and EB.TABLE.PROCEDURES.CROSSVAL.HOOK
- The T24 field specifying this hook is the CROSSVAL.PROC field in EB.TABLE.PROCEDURES and INPUT.ROUTINE in VERSION and VERSION.CONTROL

This interface enables the implementer to return a validationResponse whose errors and enrichments will be processed at this stage.

This interface is invoked when the user validate a record

The EB.API hook used by this interface is VERSION.INPUT.ROUTINE.HOOK and EB.TABLE.PROCEDURES.CROSSVAL.HOOK

The T24 field specifying this hook is the CROSSVAL.PROC field in EB.TABLE.PROCEDURES and INPUT.ROUTINE in VERSION and VERSION.CONTROL

| validateRecord() – Parameters

- Parameters:
 - **application** - The name of the application to which the record belongs.
 - **currentRecordId** - The id of the record being processed.
 - **currentRecord** - The record being processed - the implementation must specify the record type.
 - **unauthorisedRecord** - The last unauthorised version of the record being processed - the implementation must specify the record type.
 - **liveRecord** - The last live version of the record being processed - the implementation must specify the record type.
 - **transactionContext** - Transaction related
- Returns:
 - **TValidationResponse** - validationResponse - errors, overrides and enrichments will be processed at this stage.

©2020 Temenos Headquarters SA - all rights reserved.



Parameters:

application - The name of the application to which the record belongs.

currentRecordId - The id of the record being processed.

currentRecord - The record being processed - the implementation must specify the record type.

unauthorisedRecord - The last unauthorised version of the record being processed - the implementation must specify the record type.

liveRecord - The last live version of the record being processed - the implementation must specify the record type.

transactionContext - Transaction related

Returns:

TValidationResponse - validationResponse - errors, overrides and enrichments will be processed at this stage.

| updateCoreRecord()

- This interface enables the implementer to define one or more records to be input during the current transaction using the specified versions and ids
- This interface is invoked when a record is authorized
- The T24 fields specifying this hook are the BEF.AUTH.PROC field in EB.TABLE.PROCEDURES and BEFORE.AUTH.RTN in VERSION and VERSION.CONTROL.

©2020 Temenos Headquarters SA - all rights reserved.



This interface enables the implementer to define one or more records to be input during the current transaction using the specified versions and ids.

If no version is specified then no action will be taken.

This interface is invoked when a record is authorised.

The EB.API hooks used by this interface are EB.TABLE.PROCEDURES.BEF.AUTH.HOOK and VERSION.BEFORE.AUTH.RTN.HOOK.

The T24 fields specifying this hook are the BEF.AUTH.PROC field in EB.TABLE.PROCEDURES and BEFORE.AUTH.RTN in VERSION and VERSION.CONTROL.

If an exception is thrown in the implementing class it will be treated as an error.

| updateCoreRecord() – Parameters

- Parameters:
 - **application** - The name of the application to which the record belongs.
 - **currentRecordId** - The id of the record being processed.
 - **currentRecord** - The record being processed - the implementation must specify the record type.
 - **unauthorisedRecord** - The last unauthorised version of the record being processed - the implementation must specify the record type.
 - **liveRecord** - The last live version of the record being processed - the implementation must specify the record type.
 - **versionNames** - The Version names to be processed.
 - **isZeroAuth** - Returns true if it is Zero auth otherwise false.
 - **currentRecordIds** - The id's of the record to be processed.
 - **currentRecords** - The records to be processed.
 - **transactionContext** - Transaction related
- Returns:
 - **nothing**

©2020 Temenos Headquarters SA - all rights reserved.



Parameters:

application - The name of the application to which the record belongs.

currentRecordId - The id of the record being processed.

currentRecord - The record being processed - the implementation must specify the record type.

unauthorisedRecord - The last unauthorised version of the record being processed - the implementation must specify the record type.

liveRecord - The last live version of the record being processed - the implementation must specify the record type.

versionNames - The Version names to be processed.

isZeroAuth - Returns true if it is Zero auth otherwise false.

currentRecordIds - The id's of the record to be processed.

currentRecords - The records to be processed.

transactionContext - Transaction related

Returns:

TValidationResponse - validationResponse - errors, overrides and enrichments will be processed at this stage.

| postUpdateRequest()

- This interface enables to post the request to update the records of any table in asynchronous mode
- This interface writes the request into the file OFS.MESSAGE.QUEUE
- To post this request into the T24, user has to run the OFS.MESSAGE.SERVICE service
- The versionId should not be a comma version in transaction data
- The T24 fields specifying this hook are the AFTER.AUTH.PROC field in EB.TABLE.PROCEDURES and AUTH.ROUTINE in VERSION and AUTH.RTN in VERSION.CONTROL.

©2020 Temenos Headquarters SA - all rights reserved.



This interface enables to post the request to update the records of any table in asynchronous mode

This interface writes the request into the file OFS.MESSAGE.QUEUE

To post this request into the T24, user has to run the OFS.MESSAGE.SERVICE service

The versionId should not be a comma version in transaction data

The T24 fields specifying this hook are the AFTER.AUTH.PROC field in EB.TABLE.PROCEDURES and AUTH.ROUTINE in VERSION and AUTH.RTN in VERSION.CONTROL.

| postUpdateRequest() – Parameters

- Parameters:
 - **application** - The name of the application to which the record belongs.
 - **currentRecordId** - The id of the record being processed.
 - **currentRecord** - The record being processed - the implementation must specify the record type.
 - **transactionData** – List of the transaction data to post the request
 - **currentRecords** - A list of records for the data at the corresponding index in transactionData
 - **transactionContext** - Transaction related
- Returns:
 - **nothing**

©2020 Temenos Headquarters SA - all rights reserved.



Parameters:

application - The name of the application to which the record belongs.

currentRecordId - The id of the record being processed.

currentRecord - The record being processed - the implementation must specify the record type.

transactionData – List of the transaction data to post the request

currentRecords - A list of records for the data at the corresponding index in transactionData

transactionContext - Transaction related

Returns:

nothing

| Lesson Summary



| Practice 5.1 - validateRecord

- Raise an **Override** or Error Message when the Credit Currency and Debit Currency are not same in FUNDS.TRANSFER transaction

Raise an Override or Error Message when the Credit Currency and Debit Currency are not same in FUNDS.TRANSFER transaction

Solution 5.1

```
public class CurrencyValidation extends RecordLifecycle {

    @Override
    public TValidationResponse validateRecord(String application, String currentRecordId, TStructure currentRecord,
        TStructure unauthorisedRecord, TStructure liveRecord, TransactionContext transactionContext) {

        FundsTransferRecord ftRecord = new FundsTransferRecord(currentRecord); //Cast TStructure to Valid Application Record

        TField debitCurrency = ftRecord.getDebitCurrency(); // Read Debit Currency in a TField Object
        TField creditCurrency = ftRecord.getCreditCurrency(); // Read Credit Currency in TField Object

        if (!debitCurrency.getValue().equals(creditCurrency.getValue())) { // Compare the both the currencies. getValue() is used to
            // read value from the object

            List<String> parameters = new ArrayList<>(); // Override or Error message can be set in Array List

            parameters.add("FT.CCY.DIFF"); // param 1 = OVERRIDE ID
            parameters.add("1"); // param 2 = Curr No
            parameters.add(creditCurrency.toString()); // param 3 = value to &
            parameters.add(debitCurrency.toString()); // param 4 = value to &

            creditCurrency.setOverride(parameters.toString()); // Set the Override message to the field
        }

        return ftRecord.getValidationResponse(); //Return the validated response
    }
}
```

The requirement is to raise a override when the debit and credit currency are not the same. Hence, `creditCurrency.setOverride` is used to raise the override. Note – we have raised a override with dynamic values from the transaction

Solution 5.1 – EB.API

EB.API	
FT.CCY.DIFF.ERROR	
Description	EN FT Currency Validation throw Error
Protection Level	Full
Source Type	Method
Java Method	validateRecord
Java Class	CurrencyValidationError
Java Package	com.temenos.training
Curr No	1
Inputter.1	7_AUTHORISER_OFS_BROWSERTC
Date time.1	12 APR 20 22:40
Authoriser	7_AUTHORISER_OFS_BROWSERTC
Company	GB0010001
Dept Code	1

©2020 Temenos Headquarters SA - all rights reserved.

 **TEMENOS**
Learning Community

Create the EB.API record with the Source Type as Method and add the Class, Method and Package names

Solution 5.1 – Version

*FUNDS.TRANSFER.AC.version

Screen Form

Screen	Description	Transaction Flow	Presentation	Relationship	API	Attributes
<input type="checkbox"/> Include Screen Control						
Input Routines						
FT.CCY.DIFF.ERROR.java						<div><div>Add</div><div>Remove</div><div>Up</div><div>Down</div></div>
▶ Authorization Routine (After Commit)						
▶ Input Routine (After Commit)						

Select Routines...

Type : ☐ jBC (Basic) ☐ Java ☒ Method

Enter name prefix or pattern (? = any character)

FT.CCY.DIFF.ERROR

Matching items:

OK Cancel

In the API tab, Attach the EB.API Record created in the previous slide as java Method

Solution 5.1

Transfer Between Accounts FT19107X09W5

Transfer Between Accounts Audit

Debit Account *	75841 Harry Crisp	Debit Amount	12.00
Debit Currency	USD US Dollar	Debit Narrative	
Debit Value Date	17 APR 2019 17 APR 2019	Cheque Number	
Cheque Type			
Ordered By.1			
Credit Account *	11223 NIKE	Credit Amount	
Credit Currency	GBP Pound Ster	Credit Narrative	
Credit Value Date	17 APR 2019 17 APR 2019	Customer Spread	
Treasury Rate	1.427074	Customer Rate	1.427074
Rounding Type			
Signatory.1			
Amount Debited		Amount Credited	
Delivery Reference.1			

Please resolve the errors below to proceed:

✗ Credit Coy GBP and Debit Coy USD are not same - Java Framework

Override is raised as the credit and debit currency are not the same.



TEMENOS
Learning Community

thank.you

tlc.temenos.com

