



**TEMENOS**  
THE BANKING SOFTWARE COMPANY

# Hook Routines for Version Exit Points

Customer How-to Guide 1.0 February 2020

# Contents

- 1 About this how-to guide ..... 3
  - 1.1 Prerequisites ..... 3
  - 1.2 Legal ..... 3
  - 1.3 History ..... 4
- 2 Introduction ..... 5
- 3 Determining the superclass and method ..... 6
  - 3.1.1 Javadoc updates ..... 8
- 4 Writing the Java implementation ..... 9
  - 4.1 Create the new Java project..... 9
  - 4.2 Extend the superclass identified in Javadoc..... 11
  - 4.3 Write the Java Implementation..... 12
  - 4.4 Place the implementation in a library and load in JBoss Classpath ..... 15
  - 4.5 Link the Java method to the version template flow via exit point..... 17
- 5 Test the version exit point ..... 20

# 1 About this how-to guide

The **Hook Routines for Version Exit Points Customer How-to Guide** describes how to write hook routines in Java for the version exit points.

## 1.1 Prerequisites

This guide assumes that:

- You have already read the **Java Extensibility Framework Overview**.
- You have a basic understanding of Java.
- Have verified JD product is installed in SPF.

## 1.2 Legal

© Copyright 2020 Temenos Headquarters SA. All rights reserved.

The information in this guide relates to TEMENOS™ information, products and services. It also includes information, data and keys developed by other parties.

While all reasonable attempts have been made to ensure accuracy, currency and reliability of the content in this guide, all information is provided "as is".

There is no guarantee as to the completeness, accuracy, timeliness or the results obtained from the use of this information. No warranty of any kind is given, expressed or implied, including, but not limited to warranties of performance, merchantability and fitness for a particular purpose.

In no event will TEMENOS be liable to you or anyone else for any decision made or action taken in reliance on the information in this document or for any consequential, special or similar damages, even if advised of the possibility of such damages.

TEMENOS does not accept any responsibility for any errors or omissions, or for the results obtained from the use of this information. Information obtained from this guide should not be used as a substitute for consultation with TEMENOS.

References and links to external sites and documentation are provided as a service. TEMENOS is not endorsing any provider of products or services by facilitating access to these sites or documentation from this guide.

The content of this guide is protected by copyright and trademark law. Apart from fair dealing for the purposes of private study, research, criticism or review, as permitted under copyright

## Hook Routines for Version Exit Points Customer How-to Guide 1.0

law, no part may be reproduced or reused for any commercial purposes whatsoever without the prior written permission of the copyright owner. All trademarks, logos and other marks shown in this guide are the property of their respective owners.

### 1.3 History

Version	Date	Change	Author
1.0	February 2020	Initial release	Lizen Bista

## 2 Introduction

The use case in this **Hook Routines for Version Exit Points Customer How-to Guide** uses the version exit points CHECK.REC.RTN and INPUT.ROUTINE to:

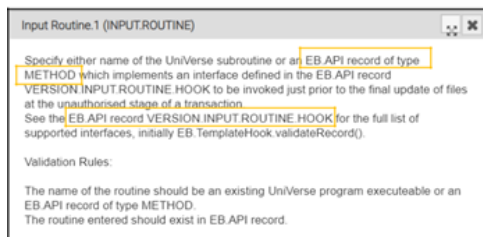
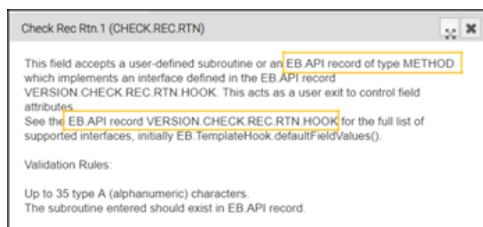
- Return errors before record display and default field values.
- Raise errors in multi-value fields.
- Raise overrides.

### 3 Determining the superclass and method

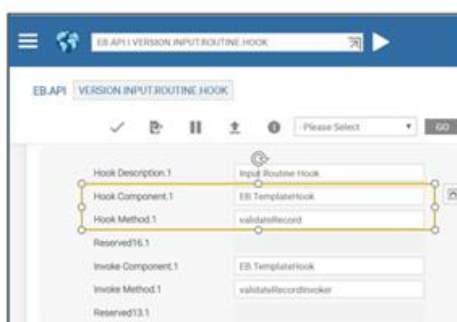
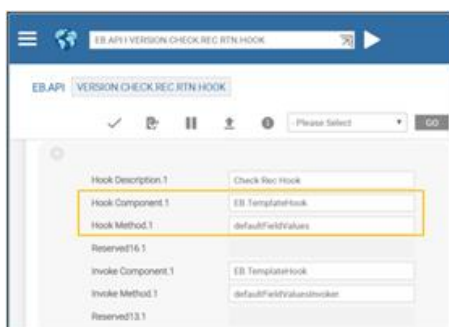
This section shows you how to determine the superclass and method you need to implement.

#### Procedure

1. Identify the exit points in the version to attach hook routines. In this case, CHECK.REC.RTN and INPUT.ROUTINE.
2. The help text for the exit point field tells you if a Java routine can be attached to it.



3. View the EB.API record indicated in the help text. This record gives the name of the hook method to override in the Java implementation of the exit point routine.

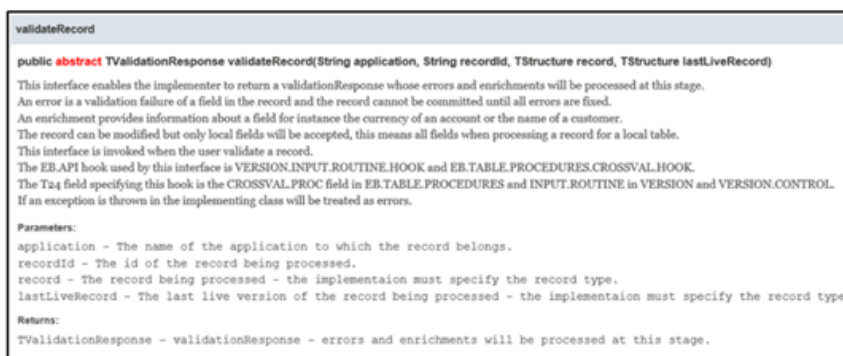
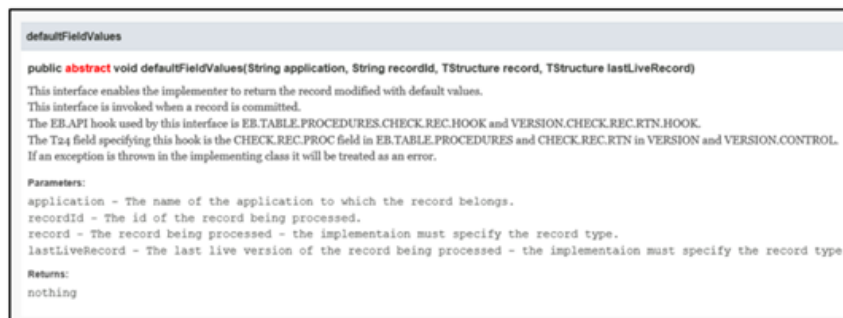
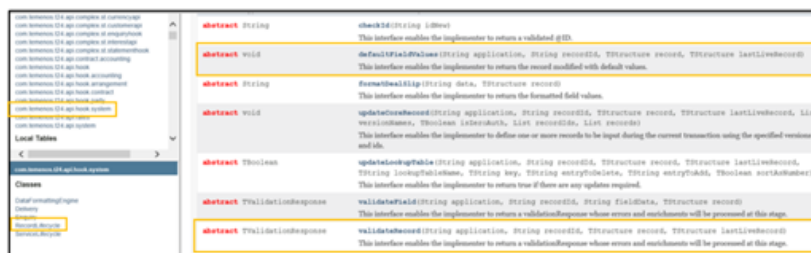


## Hook Routines for Version Exit Points Customer How-to Guide 1.0

- The L3 API documentation provides a detailed description of the hook method and its parameters. You can download Javadocs for L3 API from either the Temenos customer support portal (TCSP) or the Temenos partner support portal (TPSP) as a zip file.

After you have downloaded the zip file:

- Extract `T24.javadoc.jar` from the zip file.
  - Double-click `T24.javadoc.jar`. This extracts the contents of the jar into a newly created subdirectory called `T24.javadoc`.
  - Double-click `T24.javadoc/index.html` to view the complete L3 API documentation in the browser.
- Select the package `com.temenos.t24.api.hook.system` and the class `RecordLifecycle`. The documentation provides a detailed description of all the methods in the class and its parameters.



### 3.1.1 Javadoc updates

Javadoc updates are packaged with the T24 updates zip that you download from the Updates portal. The updates are component-wise Javadoc updates, for example, `EB_TemplateHook.javadoc.jar` in the `Transact_L3_Javadoc` folder.

#### Procedure

1. Extract the jar.
2. Use the `{TAFJ_HOME}/bin/tJavadocMerge` tool to perform the merge.

```
tJavadocMerge -merge <List of component-wise.javadoc.jar>  
<Base javadoc jar> -o <new_jar_name>
```

#### Example 1

```
tJavadocMerge -merge  
C:\DocUpdates\EB_TemplateHook.javadoc.jar  
C:\JavaDoc\T24.javadoc.jar -o C:\JavaDoc\T24-1.javadoc.jar
```

#### Example 2

```
tJavadocMerge -merge C:\DocUpdates\*.javadoc.jar  
C:\JavaDoc\T24.javadoc.jar -o C:\Javadoc\T24-2.javadoc.jar
```

In both examples, `T24.javadoc.jar` is the L3 API document base.



## 4 Writing the Java implementation

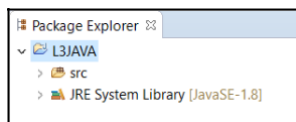
This section describes how to write the Java implementation for the application exit point routine.

To write the hook routine in Java, start Design Studio and switch to the Java perspective.

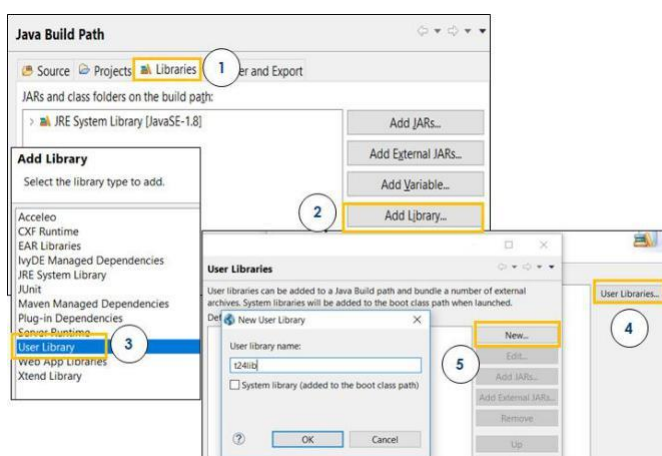
### 4.1 Create the new Java project

#### Procedure

1. In Design Studio, create the new Java project (**File > New > Java project**). In the wizard, supply a project name.



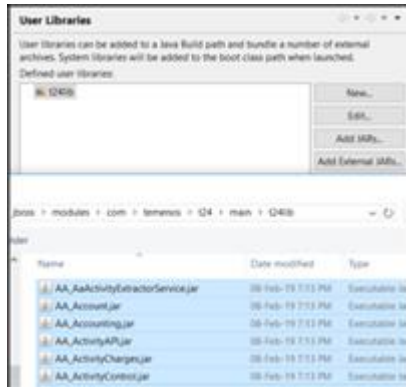
2. Configure the build path settings for the Java project to add dependent T24 and TAFJ libraries.
  - a. Right click the **project > Build path > Configure build path**.
  - b. Click the **Libraries tab > Add Library > User Library > User Libraries**.
  - c. In the **User Libraries** window, click **New** and give the library a name, for example **t24lib**.



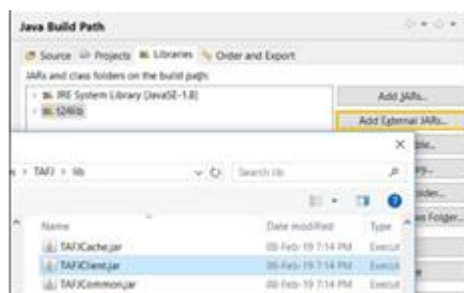
## Hook Routines for Version Exit Points Customer How-to Guide 1.0

- d. Click **Add external jars**. Navigate to the T24 libraries folder under %JBOS\_HOME%/modules. Select all the jars and click **Open**. Click **OK** and finish.

Alternatively, you can add the required T24 hook jars, such as `EB_TemplateHook.jar` and so on, using **Add External JARS**.



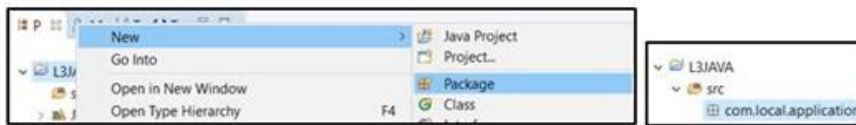
- e. Click **Add External JARS** and add TAFJClient.jar from %TAFJ\_HOME%/lib



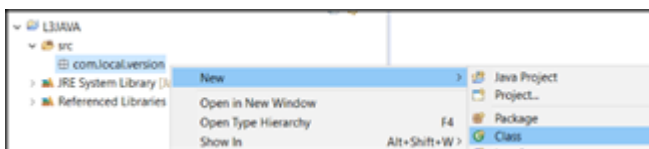
## 4.2 Extend the superclass identified in Javadoc

### Procedure

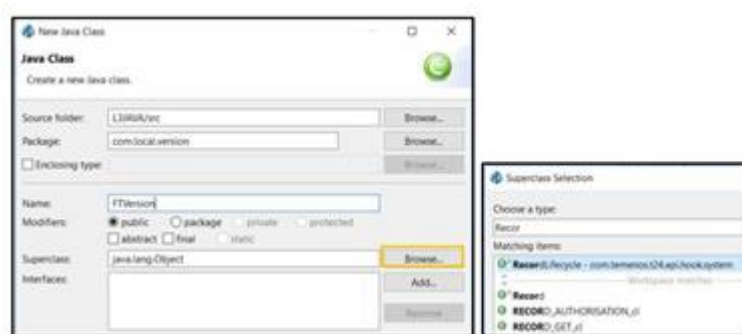
1. Create a new Java package. Right click **Project > New > Package** and give the package a name.



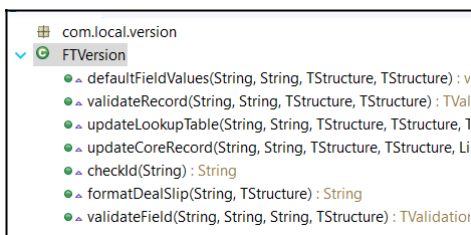
2. Right click the package and add a new Class to the package. Provide a class name, for example, `FTVersion`.



3. Extend the superclass `RecordLifecycle` for your class. Click **Finish**.



4. The superclass `RecordLifecycle` has several methods that can be overridden. The EB.API record mentioned in the exit point field help text, identifies the specific method to provide definition in Java.



### 4.3 Write the Java Implementation

#### Procedure

1. Use the inherited method `defaultFieldValues()` to do the following:
  - a. Default TRANSACTION.TYPE to AC.
  - b. Default DEBIT.CURRENCY with the local currency of the logged in company.
  - c. Raise a form level error if the DISPO.OFFICER of the logged in USER is not set.
2. Use the inherited method `validateRecord()` to do the following:
  - a. Raise an error in the multi-value field ORDERING.CUST field.
  - b. Raise an override if the transaction is input beyond a specified time.
  - c. Raise an override if DEBIT.VALUE.DATE does not match T24 TODAY date.
  - d. Raise an override if CREDIT.VALUE.DATE falls on a holiday.
3. Click CTRL + S to save the code. See code sample below.

```
package com.local.version;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.List;
import java.util.ListIterator;
import java.util.logging.Logger;

import com.temenos.api.TBoolean;
import com.temenos.api.TDate;
import com.temenos.api.TField;
import com.temenos.api.TString;
import com.temenos.api.TStructure;
import com.temenos.api.TValidationResponse;
import com.temenos.api.exceptions.T24CoreException;
import com.temenos.t24.api.hook.system.RecordLifecycle;
import com.temenos.t24.api.records.customer.CustomerRecord;
```

```
import
com.temenos.t24.api.records.fundstransfer.Fun
dsTransferRecord; import
com.temenos.t24.api.system.DataAccess; import
com.temenos.t24.api.system.Date;
import com.temenos.t24.api.system.Session;

public class FTVersion extends RecordLifecycle {

    Logger logger = Logger.getLogger("T24");

    @Override
    public void defaultFieldValues(String application, String recordId,
TStructure record, TStructure lastLiveRecord) {
        FundsTransferRecord ft = new
FundsTransferRecord(record);
        Session session = new
Session(this);
        if
            (session.getUserDispoOffice
r().compareTo("") == 0) {
            throw new
T24CoreException("", "EB-
DOFF.REQUIRED");
        } else {
            if (ft.getTransactionType().getValue() == "") {
                ft.setTransactionType("AC");
            }
            if (ft.getDebitCurrency().getValue() == "") {
                TField localCurrency =
session.getCompanyRecord().getLocalCurrency();
                ft.setDebitCurrency(localCurrency.getValue());
            }
            record.set(ft.toStructure());
        }
    }

    @Override
    public TValidationResponse validateRecord(String application,
String recordId, TStructure record, TStructure lastLiveRecord) {

        FundsTransferRecord ft = new FundsTransferRecord(record);
        DataAccess da = new DataAccess(this);
        Date date = new Date(this);
        Session session = new Session(this);

        List<TField> ordCustMV =
ft.getOrderingCust();
        ListIterator<TField> iterator =
ordCustMV.listIterator(); while
(iterator.hasNext()) {
            TField ordCustomerId = iterator.next();
            try {
                new CustomerRecord(da.getRecord("CUSTOMER",
ordCustomerId.toString())); } catch
(T24CoreException e) {
                ordCustomerId.setError("FT-ACRC.MISSING.CUST");
            }
        }
    }
}
```

## Hook Routines for Version Exit Points Customer How-to Guide 1.0

```
TField creditValueDate = ft.getCreditValueDate();
TDate crDate = new TDate(creditValueDate.toString());
String creditValueDateType = date.getDayType(crDate);

TField debitValueDate = ft.getDebitValueDate();
String debDate = debitValueDate.getValue();
String currentDate = session.getCurrentVariable("!TODAY");

if (isAfterCutoffTime("18:30:00")) {
    ft.getTransactionType().setOverride("Beyond working hours, message
delivered");
}
if (!debDate.equals(currentDate)) {
    debitValueDate.setOverride("Debit Value Date is not Today's date");
}
if (creditValueDateType.equals("HOLIDAY")) {
    creditValueDate.setOverride("Credit Value Date falls on a holiday");
}
return ft.getValidationResponse();
}

private boolean
isAfterCutoffTime(String
cutoffTime) {
    java.util.Date cutOff;
    java.util.Date systemTime;
    DateFormat df = new SimpleDateFormat("HH:mm:ss");
    Calendar calobj = Calendar.getInstance();
    String formattedDate = df.format(calobj.getTime());
    try {
        systemTime = df.parse(formattedDate);
        cutOff = df.parse(cutoffTime);
    } catch (ParseException e) {
        throw new RuntimeException("Cannot parse time!");
    }
    return systemTime.after(cutOff);
}

@Override
public TBoolean updateLookupTable(String application, String
recordId, TStructure record, TStructure lastLiveRecord, TString
lookupTableName, TString key, TString entryToDelete, TString
entryToAdd, TBoolean sortAsNumber) {
    // TODO
    Auto-
    generated
    method
    stub
    return
    null;
}

@Override
public void updateCoreRecord(String application, String recordId,
TStructure record, TStructure lastLiveRecord, List<String> versionNames,
TBoolean isZeroAuth, List<String> recordIds, List<TStructure> records) {
    // TODO Auto-generated method stub
}
```

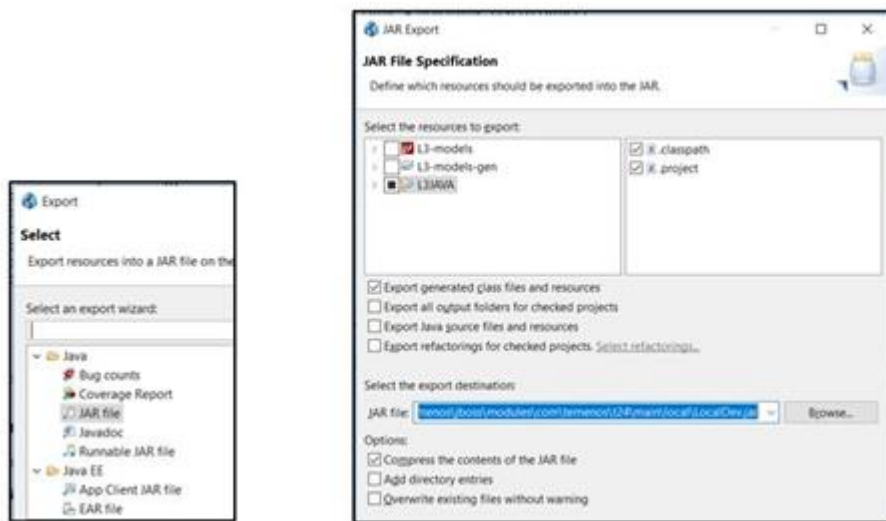
```
    }
    @Override
    public void updateCoreRecord(String application, String recordId,
        TStructure record, TStructure lastLiveRecord, List<String> versionNames,
        TBoolean isZeroAuth, List<String> recordIds, List<TStructure> records) {
        // TODO Auto-generated method stub
    }
    @Override
    public String checkId(String idNew) {
        // TODO
        Auto-generated
        method stub return
        null;
    }
    @Override
    public String formatDealSlip(String data, TStructure record) {
        // TODO
        Auto-generated
        method stub return
        null;
    }
    @Override
    public TValidationResponse validateField(String application, String
        recordId, String fieldData, TStructure record) {
        // TODO
        Auto-generated
        method stub return
        null;
    }
}
```

### 4.4 Place the implementation in a library and load in JBoss Classpath

#### Procedure

1. After the Java code is written, right click the **project** > **Export** > **JAR** file. Select the export destination.

## Hook Routines for Version Exit Points Customer How-to Guide 1.0



2. Update `module.xml` in `%JBoss_HOME%/modules` with the new jar path and name. Restart JBoss.

```
<module xmlns="urn:jboss:module:1.0" name="com.temenos.t24">
  <resources>
    <!-- Insert resources here -->
    <resource-root path="/local/LocalDev.jar" />
    <resource-root path="/t24lib/AA_AccountsData.jar" />
    <resource-root path="/t24lib/AA_AUC.jar" />
    <resource-root path="/t24lib/AA_ActivityExtractorServ.jar" />
    <resource-root path="/t24lib/AA_Account.jar" />
  </resources>
</module>
```



## 4.5 Link the Java method to the version template flow via exit point

### Procedure

1. Create EB.API records for the hook routines. A record is created for each of the two overridden methods `defaultFieldValues` and `validateRecord`.

2. Create a version - for example, FUNDS.TRANSFER, L3 - in Design Studio with the following fields and deploy to target environment.

Field Name	Translation	Row Position	Column Position	Text Max Length
TRANSACTION.TYPE	Transaction Type	1	0	35
DEBIT.ACCT.NO	Debit Account	2	0	35
DEBIT.CURRENCY	Debit Currency	3	0	35
DEBIT.AMOUNT	Debit Amount	4	0	35
DEBIT.VALUE.DATE	Debit Value Date	5	0	35
CREDIT.ACCT.NO	Credit Account	6	0	35
CREDIT.VALUE.DATE	Credit Value Date	7	0	35
ORDERING.CUST	Ordering customer	8	0	35

## Hook Routines for Version Exit Points Customer How-to Guide 1.0

3. In the **Screen** tab, select the application fields listed in the table above. Supply a user defined field name in the **Translations** tab and the **Text Max Length**.

Application: FUNDS.TRANSFER.L3

Screen Name: L3

T24 Name: FUNDS.TRANSFER.L3

Screen Group:

Number of Approvers:

Field Selection

Application Fields

Mandatory Fields

Selected Fields

TRANSACTION.TYPE

DEBIT.ACCT.NO

DEBIT.CURRENCY

DEBIT.AMOUNT

DEBIT.VALUE.DATE

CREDIT.ACCT.NO

CREDIT.VALUE.DATE

ORDERING.CUST-1

Widget Properties

Presentation Pane

Layout type: None

Display Type: None

Input Behavior: None

Case Convention: None

Expansion: None

Text Max Length: 35

Enrichment Length: 10

Positioning

Row Position: 1

Column Position: 0

Label Column Position: 0

Constraints

Mandatory

Rekey Required

Validation

Hot Validate

Hot Field

4. In the **Presentation** tab, set **Records per page = 1** and **Fields per line = MULTI**. Supply **Row Position** and **Column Position** for the fields.

Records per page: 1

Fields per line: MULTI

Initial Cursor Position:

Browser Toolbar:

Languages:

English

French

German

Header

Header 1

Header 2

User Information (Header)

Footer

Edit Translation

Remove Translation

Language

Translation

English

French

German

Field Positions

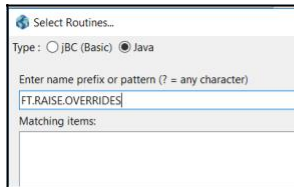
Field Name	Row Position	Column Position	Label Column Position
TRANSACTION_TYPE	1	0	
DEBIT_ACCT_NO	2	0	
DEBIT_CURRENCY	4	0	
DEBIT_AMOUNT	3	0	
DEBIT_VALUE_DATE	5	0	
CREDIT_ACCT_NO	6	0	
CREDIT_VALUE_DATE	7	0	
ORDERING_CUST	8	0	

5. In the **API** tab, expand **Input Routines** and click **Add**. A **Select Routines** dialog is displayed:
  - a. Select **Type = Java**.
  - b. Enter the EB.API record ID created earlier, for example, FT.RAISE.OVERRIDES.

## Hook Routines for Version Exit Points Customer How-to Guide 1.0

On deployment, the EB.API gets updated (incorrectly) with **Source Type = Java**. You need to manually modify the record in T24 to set the **Source Type = METHOD**. Currently, support for EB.API records with Type = METHOD is not available..

c. Repeat the above steps for Pre Process API.



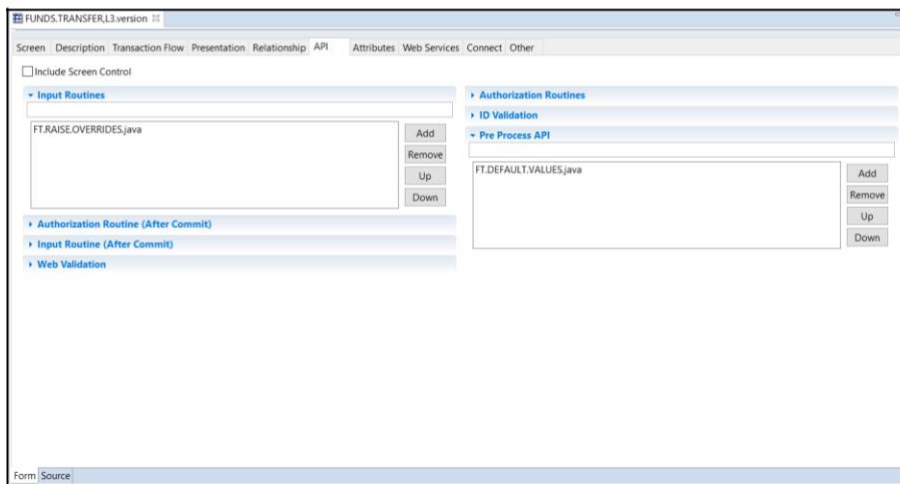
Select Routines...

Type: ☐ JBC (Basic) ☒ Java

Enter name prefix or pattern (? = any character)

FT.RAISE.OVERRIDES

Matching items:



FUNDSTRANSFERL3 version 33

Screen Description Transaction Flow Presentation Relationship API Attributes Web Services Connect Other

☐ Include Screen Control

**Input Routines**

FT.RAISE.OVERRIDES.java

Add Remove Up Down

**Authorization Routine (After Commit)**

**Authorization Routines**

**ID Validation**

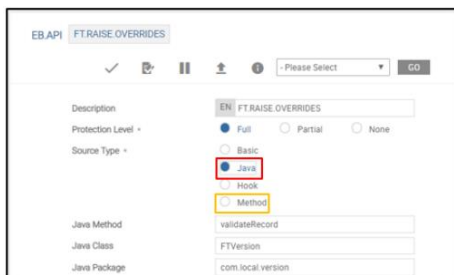
**Pre Process API**

FT.DEFAULT.VALUES.java

Add Remove Up Down

Form Source

6. Modify the EB.API records FT.RAISE OVERRIDES and FT.DEFAULT.VALUES using Browser and reset **Source Type = METHOD**.



EB.API FT.RAISE.OVERRIDES

Description EN FT.RAISE.OVERRIDES

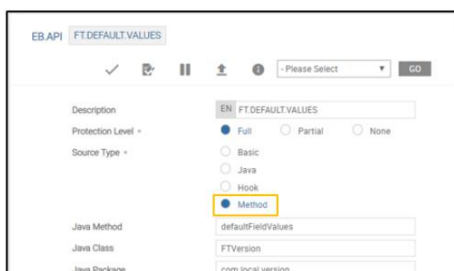
Protection Level ☒ Full ☐ Partial ☐ None

Source Type ☐ Basic ☒ Java ☐ Hook ☐ Method

Java Method validateRecord

Java Class FTVersion

Java Package com.local.version



EB.API FT.DEFAULT.VALUES

Description EN FT.DEFAULT.VALUES

Protection Level ☒ Full ☐ Partial ☐ None

Source Type ☐ Basic ☐ Java ☐ Hook ☒ Method

Java Method defaultFieldValues

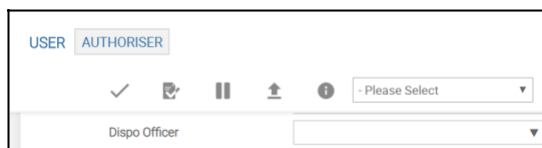
Java Class FTVersion

Java Package com.local.version

## 5 Test the version exit point

### Procedure

1. Launch the version.
2. When the DISPO.OFFICER field of the logged in USER is not set, the template flow returns with the error raised from L3 java code.



USER AUTHORISER

✓ ⚙ ⏸ ⬆ ⓘ - Please Select

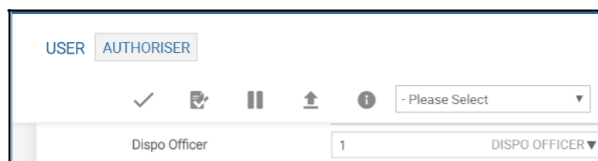
Dispo Officer



TEMENOS FT.L3 I F3

Officer required for DISPO processing

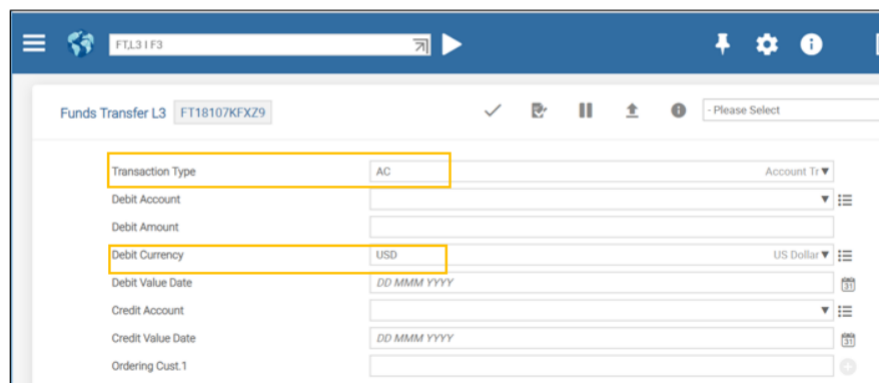
3. When the Dispo Officer is set, the java implementation of the check record routine executes and defaults fields.



USER AUTHORISER

✓ ⚙ ⏸ ⬆ ⓘ - Please Select

Dispo Officer 1 DISPO OFFICER



Funds Transfer L3 FT18107KFXZ9

Transaction Type AC Account Tr

Debit Account

Debit Amount

Debit Currency USD US Dollar

Debit Value Date DD MMM YYYY

Credit Account

Credit Value Date DD MMM YYYY

Ordering Cust.1

4. Supply field values and commit. An error is raised against the 2nd multi-value field position with validation error.

# Hook Routines for Version Exit Points Customer How-to Guide 1.0

The screenshot shows the 'Funds Transfer L3' form with the transaction ID 'FT18107KFXZ9'. The form fields are as follows:

Field	Value
Transaction Type	AC
Debit Account	21067
Debit Amount	100.00
Debit Currency	USD
Debit Value Date	18 APR 2018
Credit Account	20737
Credit Value Date	22 APR 2018
Ordering Cust.1	100100
Ordering Cust.2	UNDEFINED

An error message box is displayed in the center: 'Please resolve the errors below to proceed: Ordering Cust.2: CUSTOMER IS MISSING'. The 'Ordering Cust.2' field is highlighted in red, and the text 'CUSTOMER IS MISSING' is shown below it.

5. Supply valid values and commit. Overrides from the Java code are raised.

The screenshot shows the 'Funds Transfer L3' form with the transaction ID 'FT18107KFXZ9'. The form fields are as follows:

Field	Value
Transaction Type	AC
Debit Account	21067
Debit Amount	100.00
Debit Currency	USD
Debit Value Date	18 APR 2018
Credit Account	20737
Credit Value Date	22 APR 2018
Ordering Cust.1	100100
Ordering Cust.2	100105

A warning message box is displayed in the center: 'CREDIT VALUE NOT A WORKING DAY WITHDRAWAL MAKES A/C BAL LESS THAN MIN BAL Unauthorised overdraft of USD 975378.45 on account 21067'. Below the warning, there is a yellow box containing the text: 'Beyond working hours, message may not be delivered', 'Debit Value Date is not Today's date', and 'Credit Value Date falls on a holiday'. The 'ACCEPT' button is highlighted in blue.