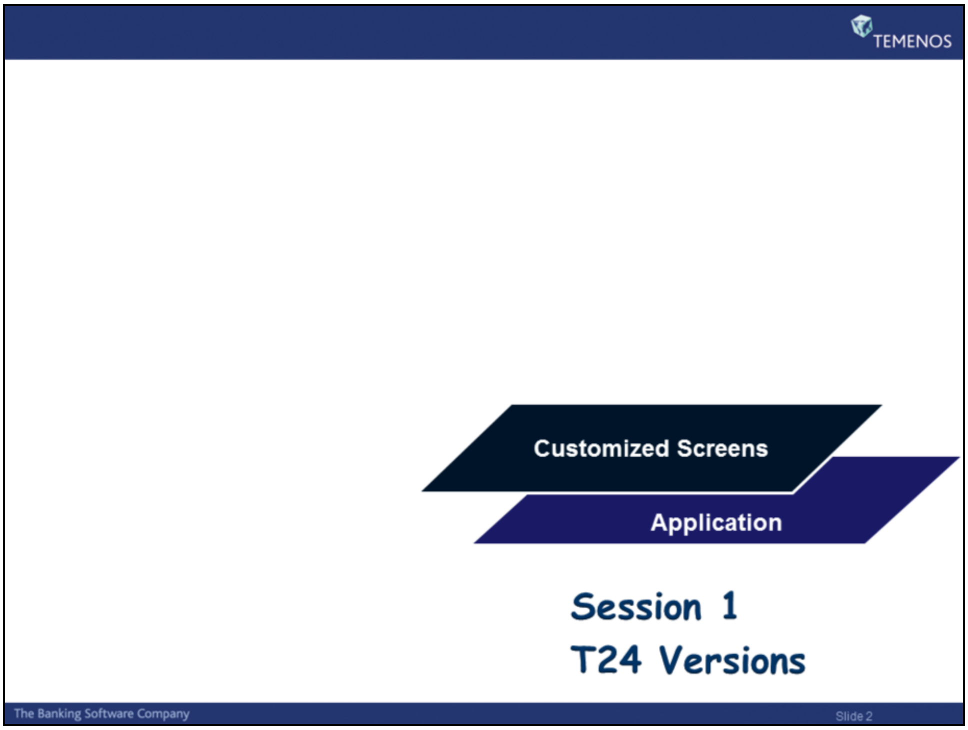
Day 4 – Session1
T24 Versions

TEMENOS EDUCATION CENTRE

TEMENOS
The Banking Software Company


TEMENOS
Customized Screens
Application
Session 1
T24 Versions
The Banking Software Company
Slide 2

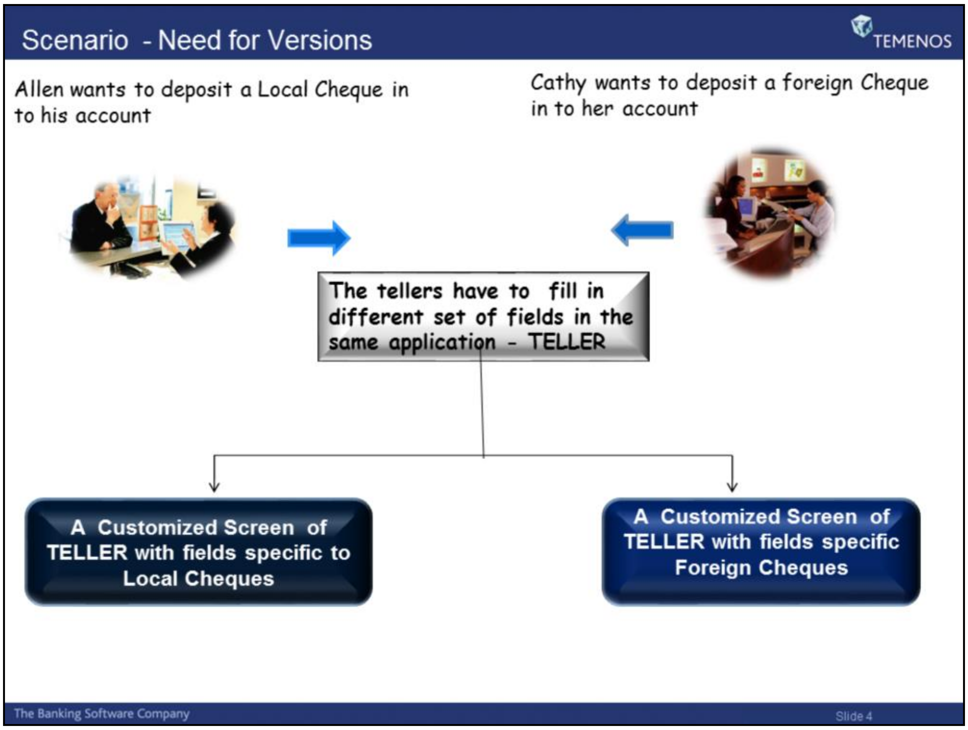
Session - Objectives

TEMENOS

- Know the T24 Customization Scope
- Understand the need for Versions
- Learn to create a version
- Elaborate the additional features of a version
- Tabbed Screens in versions
- Understand the use of Re-Key feature

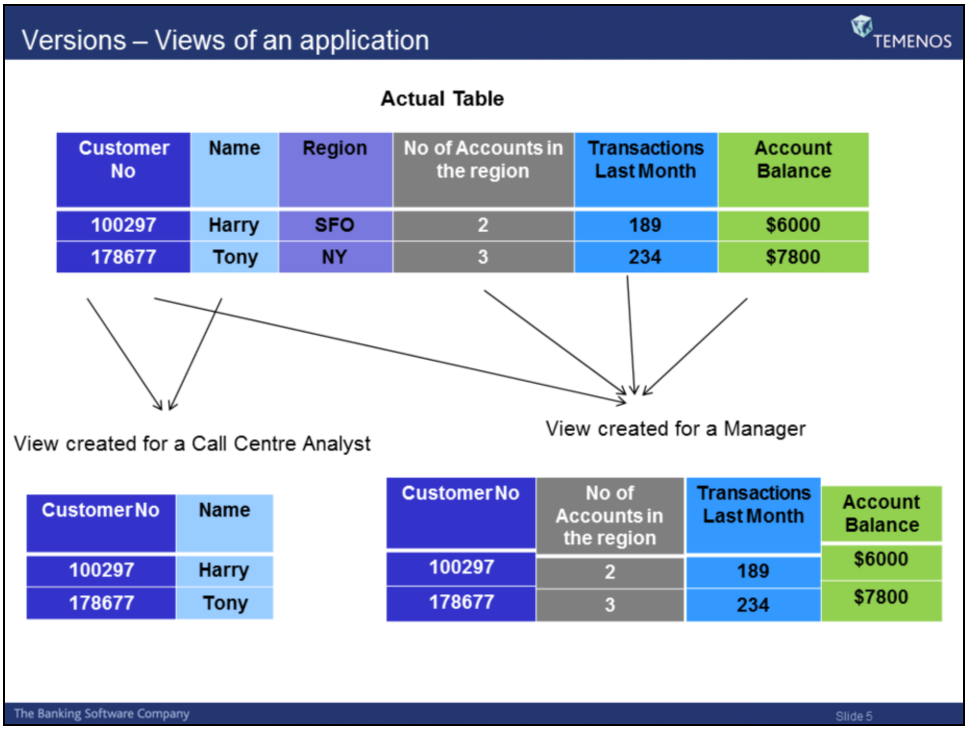
The Banking Software Company

Slide 3

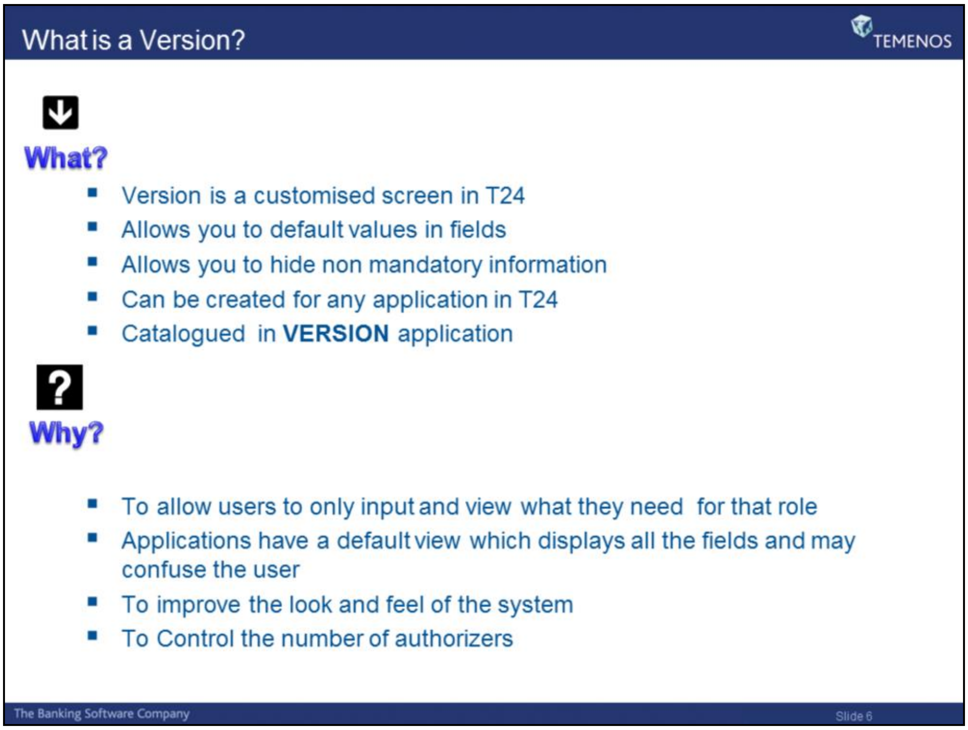


Allen and Cathy want to deposit cheques in bank ABC. Allen wants to deposit a local cheque ,whereas Cathy has a foreign cheque . The teller has to fill these details in a TELLER application. There are some fields that are required for a foreign cheque but not for a local cheque. Example : Currency . How can we make things easier ?

T24 gives us the luxury of designing our own screens from a base application. Here, TELLER is the base application. We cull out fields required for a local cheque and design a screen .Similarly we can design another one for foreign cheques. Now the tellers can invoke the screen they want ! Simple, isn't it?



Versions can be compared to the concept of “Views “ in Database Management system . A table may consist of several columns(fields) that might not be required for all users. So the users can create views, consisting of the columns that they need . When we use a spreadsheet with multiple columns, we hide the columns that might not be necessary and project those we need. Versions help us in creating screens with the fields we want.

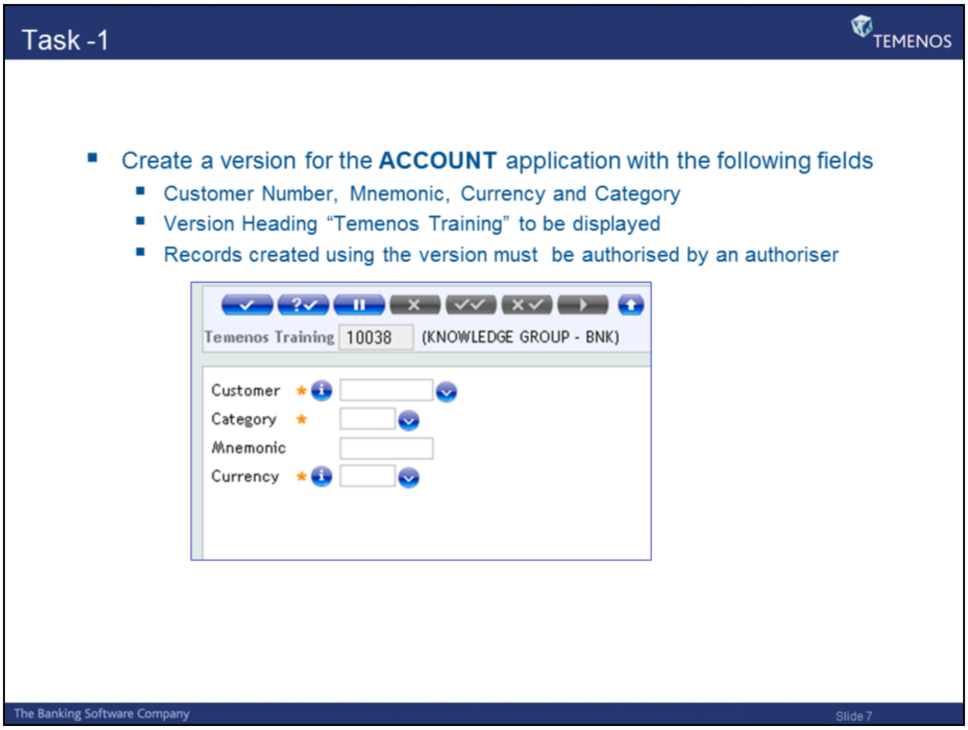


Imagine that you have to input a funds transfer transaction in T24. You will use the FUNDS.TRANSFER application, and the input function to do so. The screen that is displayed to you contains all the fields in the FUNDS.TRANSFER application some of which may not be relevant for the transaction that is being input. A first time user may not be able to figure out which fields to use and which not to. What if you had a customized screen that showed you only the fields that you require to input?

You can create customized application screens in T24. They are called versions. The VERSION application is used to create customized application screens. Your version you can have the mandatory fields of the FUNDS.TRANSFER application.

Can you view data using versions?

Versions behave like the application, but only look different. Thus all functions can be used with a version. So, yes you can view data using a version. Versions work with all functions of T24.



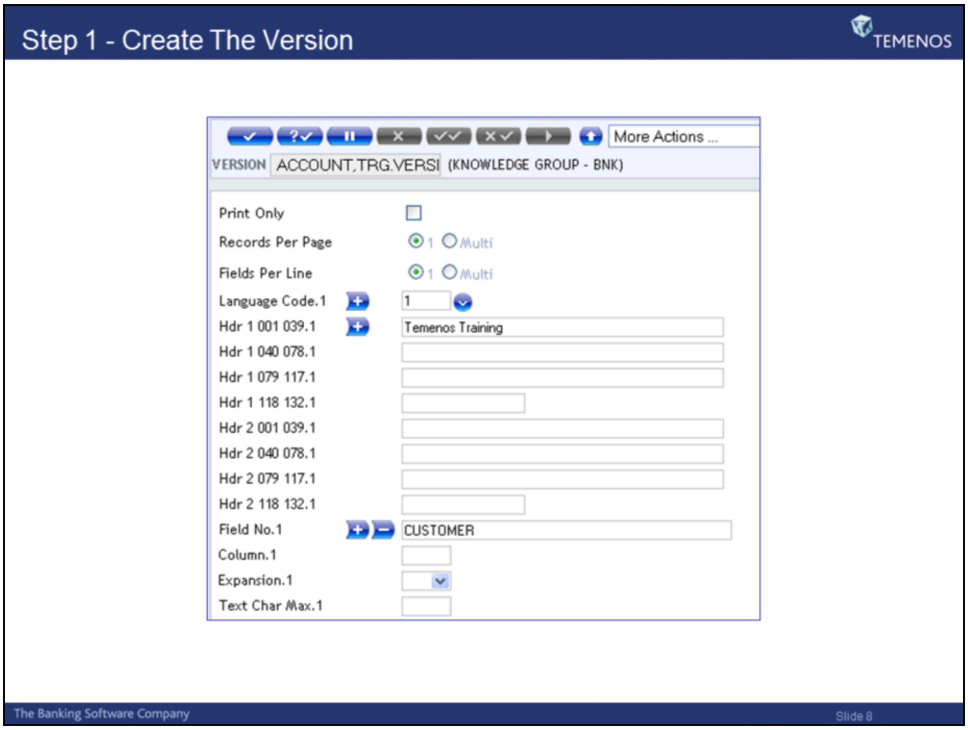
This task teaches you how to create a simple version of the ACCOUNT application in T24.

The requirements of the task are listed below:

1. The version must comprise the following fields of the ACCOUNT application
   Customer Number, Mnemonic, Currency and Category.

2. The header "Temenos Training" must be displayed.

3. The accounts created using the version should be authorised by another user.



To create a version in T24, you must create a record in the VERSION application. The format of the ID for the VERSION application is T24ApplicationName,VersionName. The application name and the comma are mandatory, the optional part of the ID is the version name. If the first portion of the ID is an invalid application name, an error message is displayed. Here the version id created is ACCOUNT,TRG.VERSION1 where ACCOUNT is a valid T24 application and TRG.VERSION1 is the version name.

PRINT.ONLY - The first field indicates whether the version is created for printing reports. By default it is blank stating that the version is just created for a customized screen

Before creating a version you must decide what fields from the application will be included in the version. Ensure that you have all the mandatory fields of the application in your version. You will now learn the use of some of the fields in the VERSION application. You can also view records using the version.

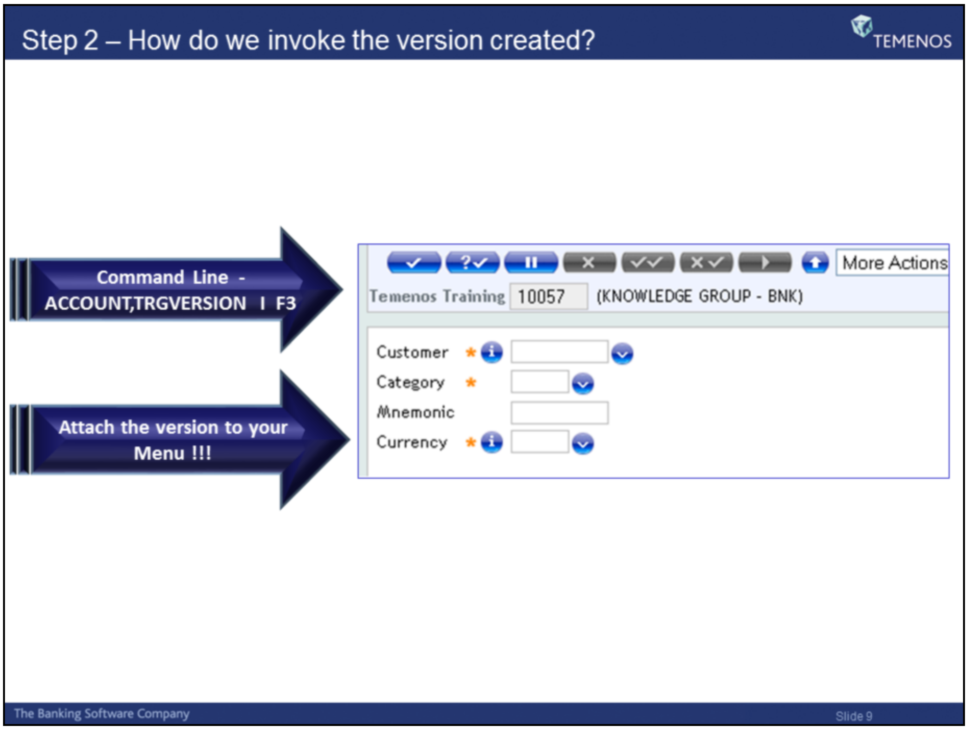
RECORDS.PER.PAGE – Indicates whether more than one record may be shown on the same page or screen in the Version that the User is creating

FIELDS.PER.LINE- is used to specify the number of fields that will be displayed in each line. 1 indicates that fields will be displayed one below the other in the version. Multi is used to display more than one field per line.

LANGUAGE.CODE - The language used to display field labels for each field in the version is specified in LANGUAGE.CODE. It is not a mandatory field and if left blank, the LANGUAGE from the user profile of the person committing the version record will be defaulted

HDR - You may specify headers for your version in different languages. The field HDR holds the text that will be displayed as header. It is a multi-value field. The header is displayed in a column range of (1) one to (132) one thirty two. The header shown in the example is displayed from column one to thirty nine.

FIELD.NO - Now you have to specify the fields of your version. This is specified in the field called FIELD.NO, where you specify the actual field name as given in the application. It is a multi-value field. You can multi-value and specify the other fields also. In this example we have chosen four fields namely, CUSTOMER, MNEMONIC, CURRENCY and CATEGORY.



1. After you authorise the version record, launch the version by specifying the version ID - ACCOUNT,TRG.VERSION1 in the command line.

2. The version behaves like the underlying application so you can either open a new record or view an existing record using the created Version.

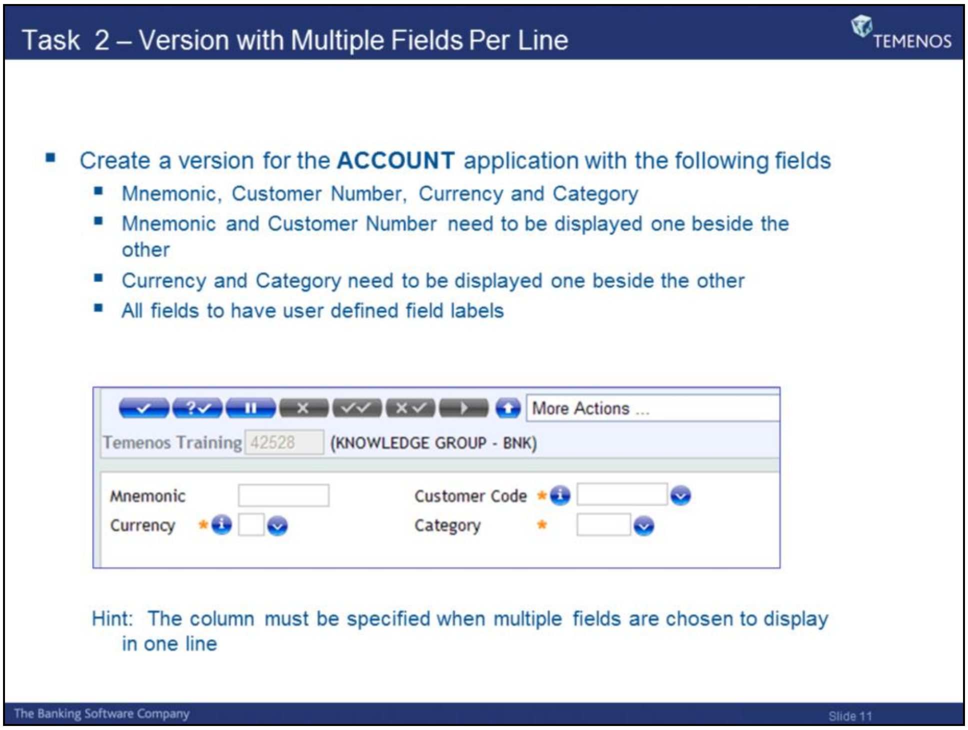
3. You can also launch the version from a menu

Workshop:

1. Create a version for the CUSTOMER application.

2. The name of the version will be CUSTOMER,XXX (XXX is your initial)

3. The Fields that must be part of the version are MNEMONIC,SHORT.NAME,NAME.1,STREET,SECTOR,ACCOUNT.OFFICER,INDUSTRY,TARGET,NATIONALITY,CUSTOMER.STATUS,RESIDENCE,LANGUAGE.



This task teaches you to create version that displays multiple fields in a line.

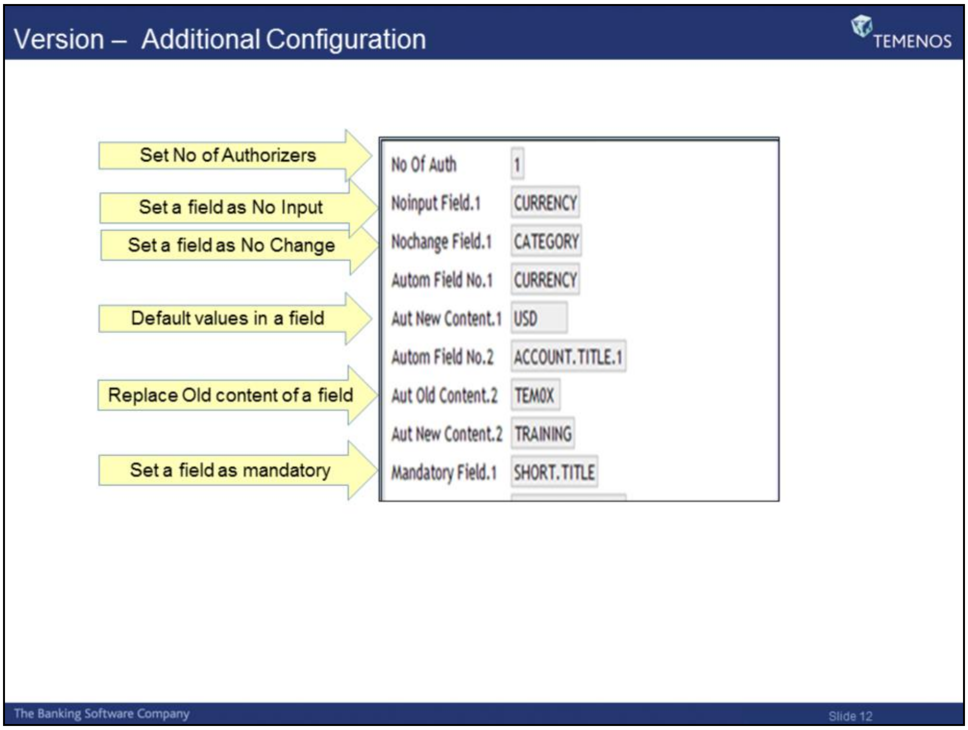
Requirement:

1. Create a version for the **ACCOUNT** application with the following fields - Mnemonic, Customer Number, Currency and Category.

2. The fields Mnemonic and Customer Number need to be displayed one beside the other

3. The fields Currency and Category need to be displayed one beside the other.

4. Labels must be user-defined.



When a new record is committed in T24, it is created in INAU status and has to authorised. Now what if you want to skip the INAU stage? Will a T24 application allow you to authorise a record straight away? The answer is No. But a version offers you a work around.

T24 allows a maximum of 2 levels of authorisation. The default is 1 for an application. In other words, one user must use the 'A' function for the record to move to Live status

NO.OF.AUTH - The number of authorisers is set in NO.OF.AUTH field. It enables the user to specify the number of authorisers required when using this version. Value for this field is defaulted based on the value of the field DEFAULT.NO.OF.AUTH in the COMPANY application. The minimum value is (0) zero and the maximum value is (2) two for this field. When it is set to 0 your version becomes an Auto Auth or self-authorisation version

When NO.OF.AUTH is set to two, two different users must authorise the record. When the record is first authorised, it moves to INA2 status. In order to make it as a LIVE record, the record has to be authorised by a different user again.

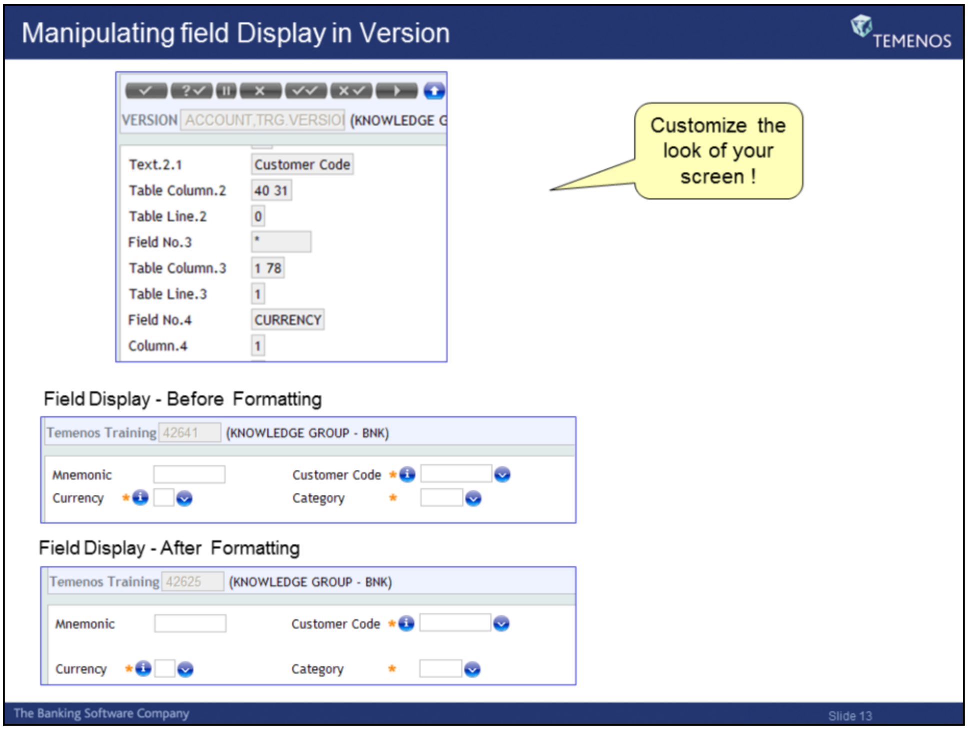
In this task you will learn how to default values into fields using a version. You will also learn how to set special field properties.

NOCHANGE.FIELD – The value in a NOCHANGE field cannot be modified after the record is authorised.

To default a value into a field, use AUTOM.FIELD.NO and AUT.NEW.CONTENT fields. The

CURRENCY field will be defaulted to USD in the above example. To replace the old content in a field with new content, specify the old content in AUT.OLD.CONTENT. Here, the value 'TEM0X' in the field AUT.OLD.CONTENT denotes that any value starting with TEM in the filed ACCOUNT.TITLE will be replaced with the content given in AUT.NEW.CONTENT.

Any fields that must be treated as mandatory are specified in the field MANDATORY.FIELD

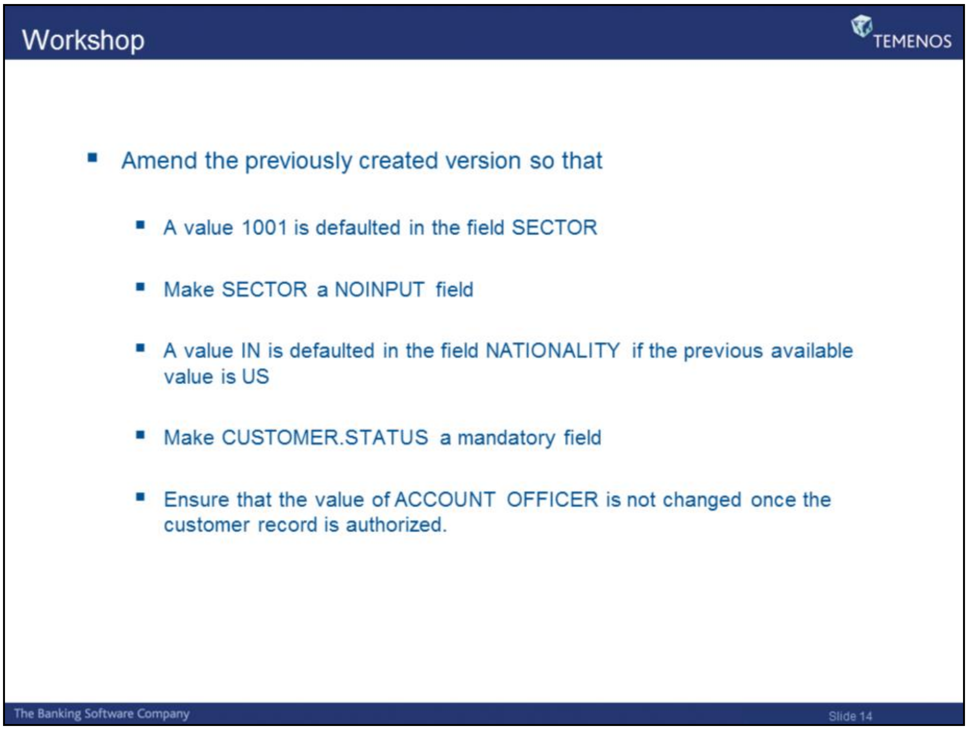


You can format the display of fields in version. You can have blank lines displayed between fields. The previous example is modified to just include a blank line between the fields.

1. How do you make the CURRENCY field to appear after a blank line?

   Specify an asterisk (*) in the field FIELD.NO to get a blank line. Since this blank line is to be displayed before CURRENCY field , specify asterisk (*) before providing the values for CURRENCY field in the associated multi-value set.

2. The output shows the blank line displayed between the two fields.



1. Amend the previously created version so that

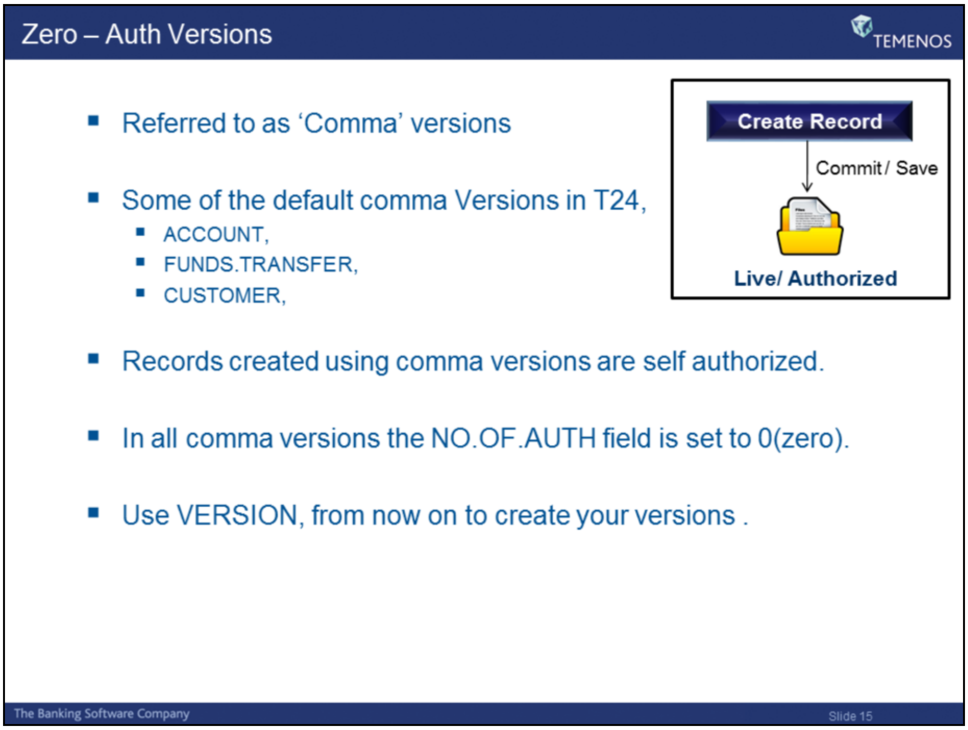
    1.1 A value thousand(1001) is defaulted in the field SECTOR

    1.2 Make SECTOR a NOINPUT field

    1.3 A value IN is defaulted in the field NATIONALITY if the previous available value is US

    1.4 Make CUSTOMER.STATUS a mandatory field

    1.5 Ensure that the value of ACCOUNT OFFICER is not changed once the customer record is authorized.



1. Comma Versions are Auto-Auth Versions in T24.

2. Some of the comma Versions in T24, are
   - 2.1 ACCOUNT,
   - 2.2 FUNDS.TRANSFER,
   - 2.3 CUSTOMER,

3. Records created using comma Versions are self-authorised records. Once you commit the records they move to LIVE status .

4. In all comma Versions the NO.OF.AUTH field is set to 0(zero). To create Zero auth versions, it is enough if we specify the name of the application in the VERSION record. Individual fields need not be added to the zero auth version. The NO.OF.AUTH field must be set to zero.

5. VERSION, is the comma Version which is used to create auto-auth Versions in T24. Use VERSION, from now on to create your versions .

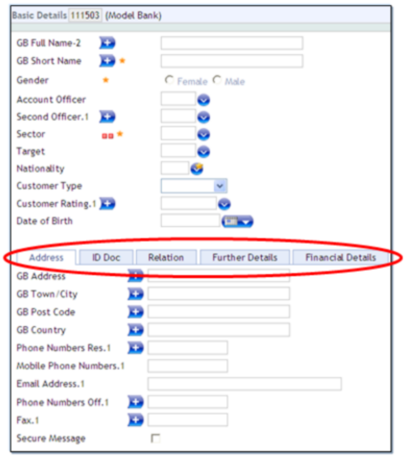
Tabbed Screens in Versions

TEMENOS

- A list of VERSIONs can be associated with the current VERSION in the form of tabs
- These associated versions must relate to the same application as the version being input

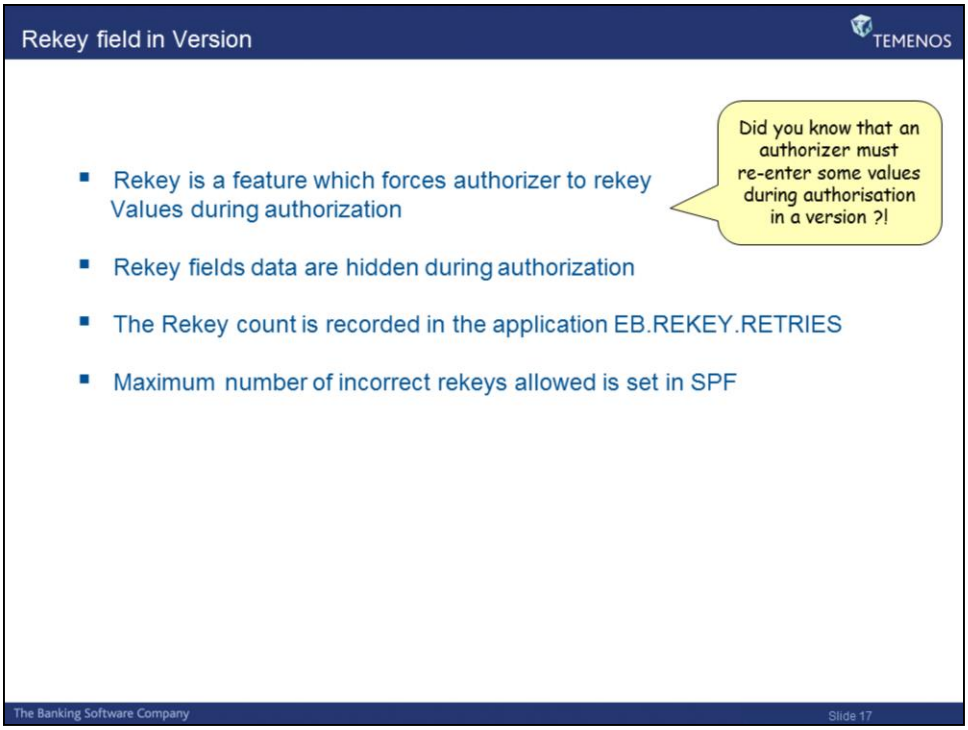
VERSION CUSTOMER,INPUT (Model Bank)

Assoc Version.1	CUSTOMER,ADDRESS
Assoc Version.2	CUSTOMER,INTRO.DETS
Assoc Version.3	CUSTOMER,RELATION
Assoc Version.4	CUSTOMER,CRM.FURTHER.DETS
Assoc Version.5	CUSTOMER,CRM.FINANCIAL
Assoc Version.6	CUSTOMER,CRM.RESIDENCE
Assoc Version.7	CUSTOMER,CRM.COMM
Assoc Version.8	CUSTOMER,CRM.KYC
Assoc Version.9	CUSTOMER,CRM.OTHERS
Assoc Version.10	CUSTOMER,REPORT.DET
Assoc Version.11	CUSTOMER,AUDIT



The Banking Software Company

Slide 16



1. The Rekey field is used in Version during authorisation

2. The authoriser must re-enter the data in specific fields to re-confirm the values entered by the Inputter. For example in an FUNDS.TRANSFER application, there are sensitive data like amount, debit account no etc which are entered by an Inputter.

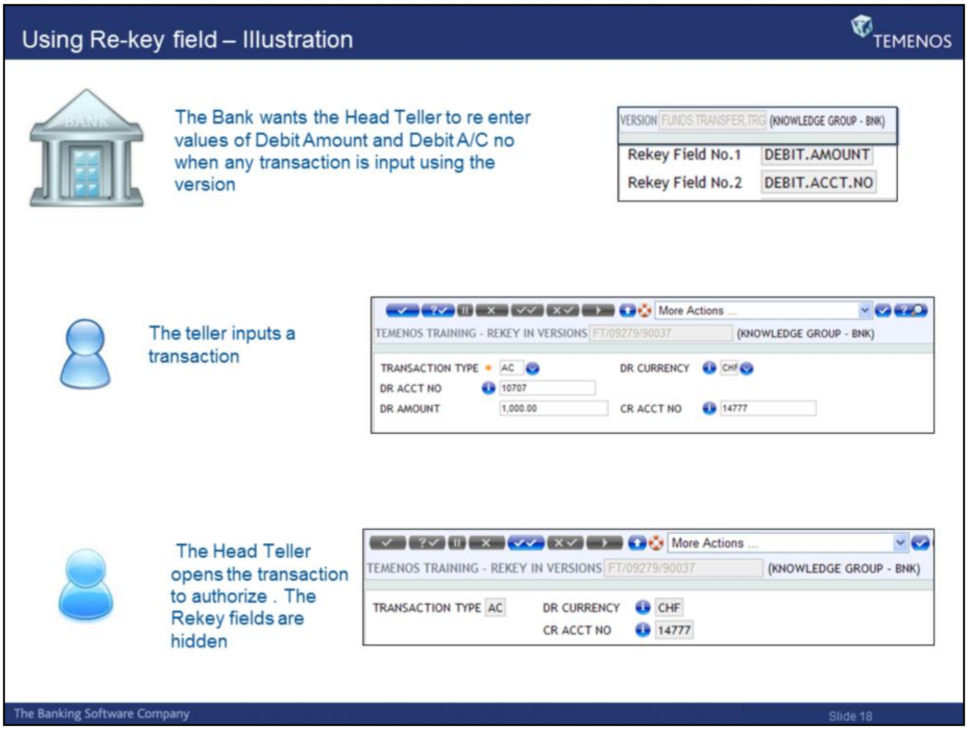
3. The rekey feature prompts the Authoriser to re-enter the data for confirmation. If the data entered by the Authoriser does not match with the data entered by the Inputter, then the record does not get authorised.

4. The data for the rekey fields are hidden during authorisation and the Authoriser needs to compulsorily re-enter the information

5. What happens when a USER who is trying to authorize the record enters incorrect rekey values multiple times ? The system records the number of incorrect rekey attempts by the same authorizer, in a file called EB.REKEY.RETRIES. A record is created in this file whenever a version that contains a rekey field is being authorized. The record id of EB.REKEY.RETRIES would be of the format COMPANY*APPLICATION*RECORD.ID.

6. The maximum number of incorrect retries allowed for a USER who tries to authorize a record can be restricted using the field NO.OF.RETRIES in SPF

application. When this field is left blank, there is no limit set for incorrect rekey attempts.

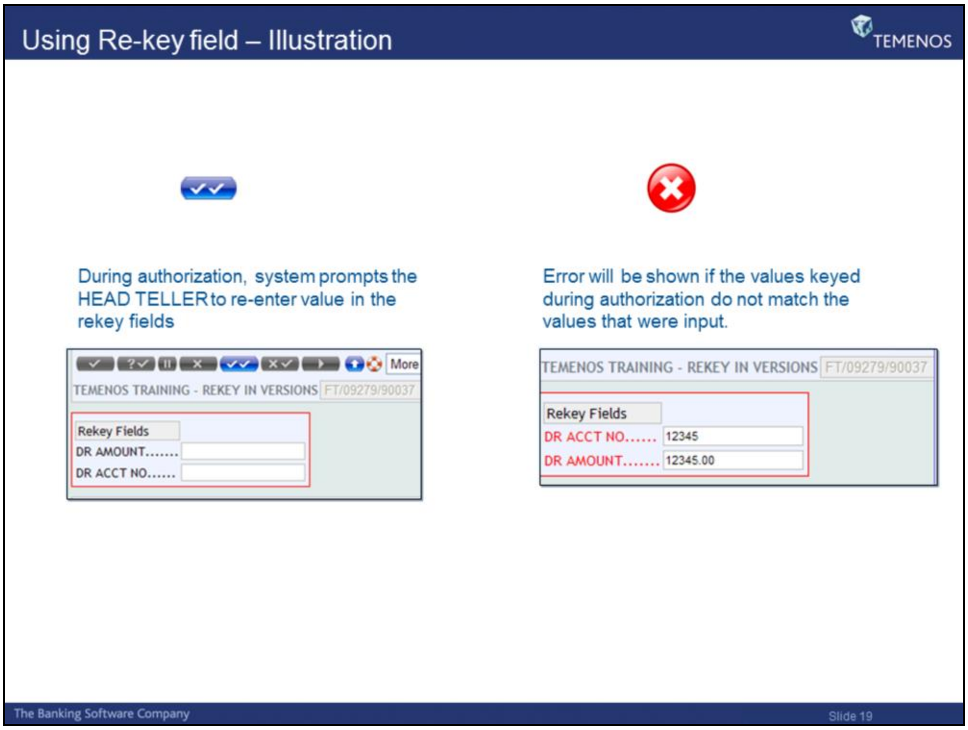


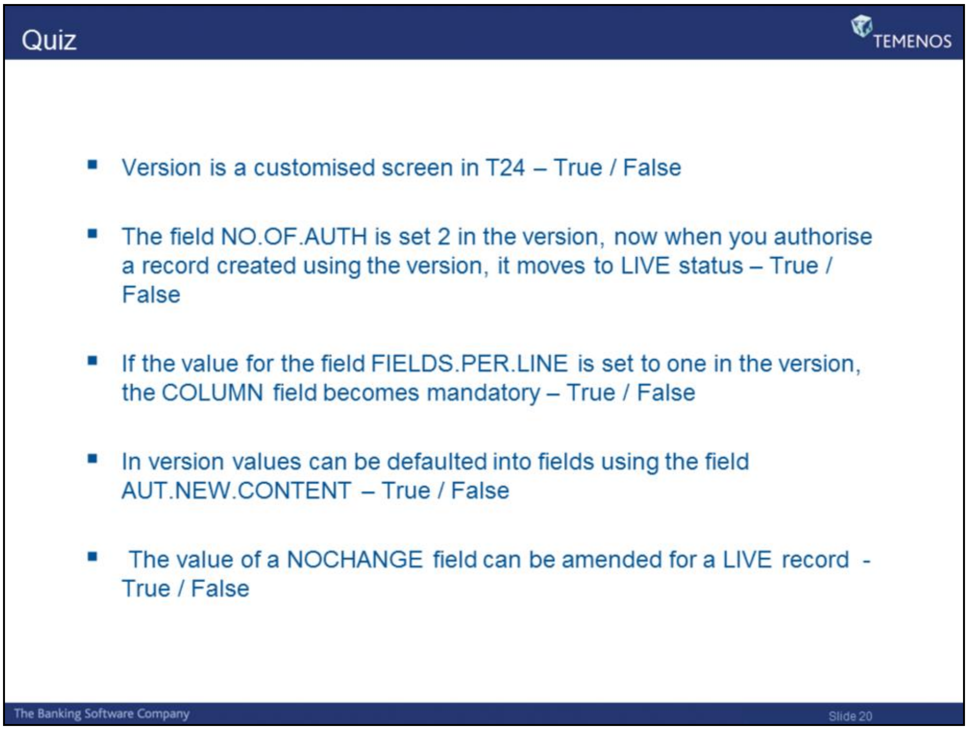
1. Input the data for Transaction type, Debit currency, Debit Account No, Debit Amount, Credit Account No and commit the record.

2. Note the record key (ID) of the Funds Transfer that was created

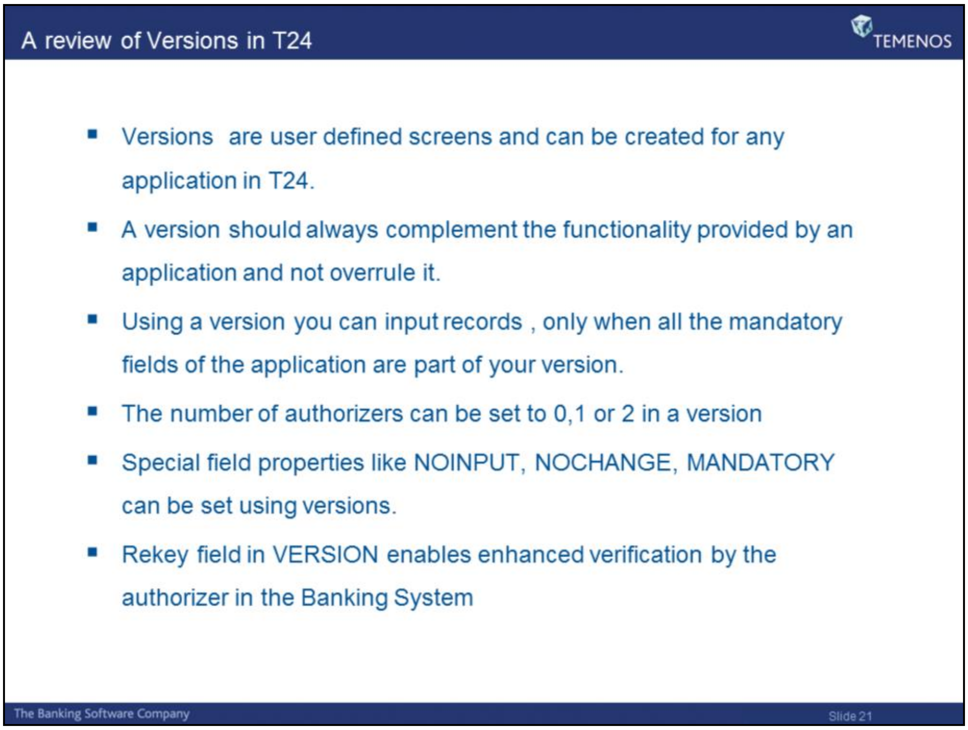
3. If the number of incorrect rekeys by the same authoriser exceeds the NO.OF.RETRIES specified on SPF, an error message “Exceeded Maximum Number of Retries” is displayed. Also, the inputter field of the transaction is updated with the authoriser’s user name. This is done so that the authoriser who exceeded the retries would not be able to authorise anymore, even with the correct rekey value.

4. Hence, a different authorizer is required to authorize the same record.


Using Re-key field – Illustration
TEMENOS
During authorization, system prompts the HEAD TELLER to re-enter value in the rekey fields
More
TEMENOS TRAINING - REKEY IN VERSIONS
FT/09279/90037
Rekey Fields
DR AMOUNT.......
DR ACCT NO......
Error will be shown if the values keyed during authorization do not match the values that were input.
TEMENOS TRAINING - REKEY IN VERSIONS
FT/09279/90037
Rekey Fields
DR ACCT NO......
12345
DR AMOUNT.......
12345.00
The Banking Software Company
Slide 19



1. True
2. False
3. False
4. True
5. False



1. View is a user defined screen and can be created for any application in T24.

2. A version should always complement the functionality provided by an application and not overrule it.

3. Using a version you can input records , only when all the mandatory fields of the application are part of your version.

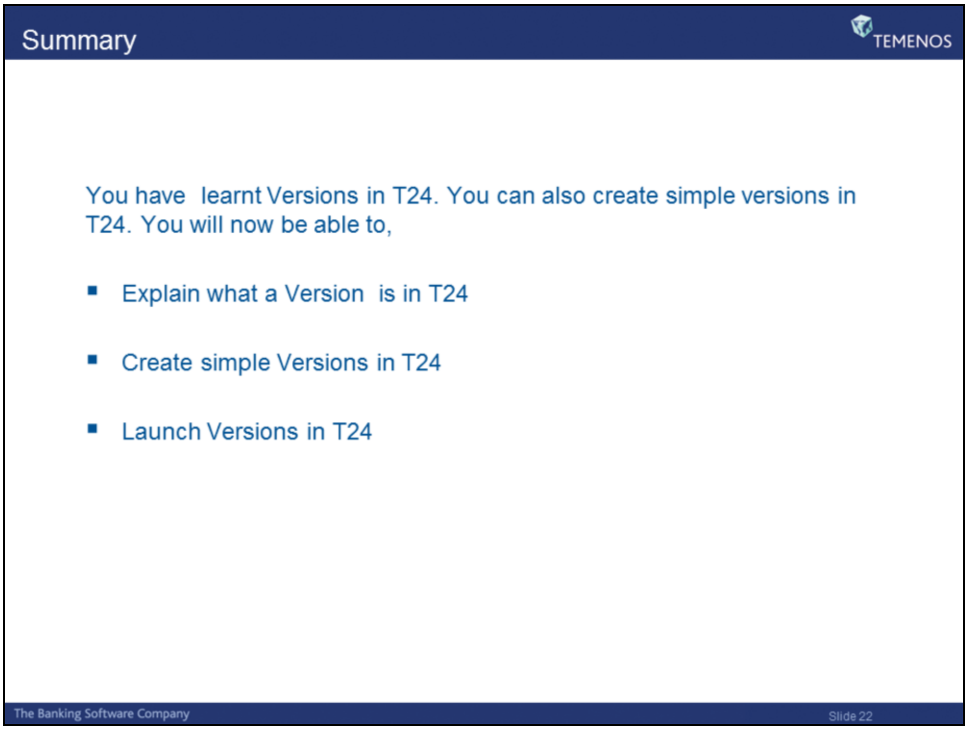
4. The number of authorisers can be set to 0,1 or 2 in a version.

5. Special field properties like NOINPUT, NOCHANGE and MANDATORY can be set using versions.

6. You can also default values into fields using versions.

7. Any additional functionality that an application must perform can be achieved by writing local routines. But how are these routines executed? For example, for any date field in T24, all that T24 does is validate the date according to the application business logic. What if the client wants an additional validation? The **VERSION** application allows you to attach these routines to perform these validations. These routines are known as User exits in T24.

8. Rekey field in VERSION enables enhanced verification by the authoriser in the Banking System



You have learnt Versions in T24. You can also create simple versions in T24. You will now be able to,

1. Explain what a Version is in T24

2. Create simple Versions in T24

3. Launch Versions in T24

Thank You

TEMENOS EDUCATION CENTRE

TEMENOS
The Banking Software Company