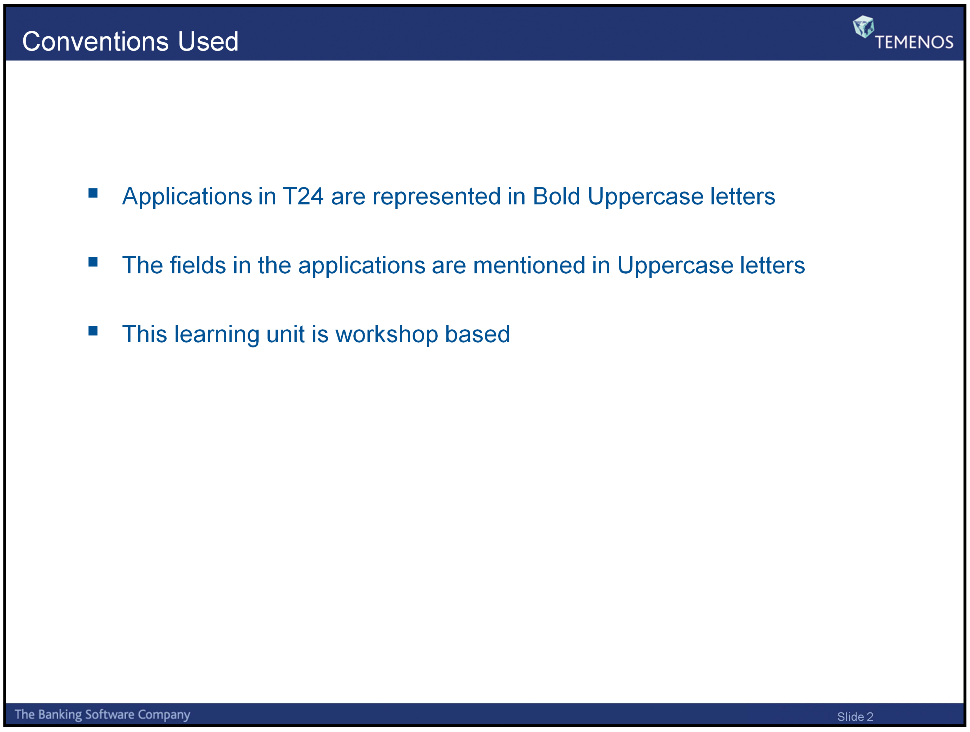
Versions in T24 – An Introduction

TEMENOS EDUCATION CENTRE

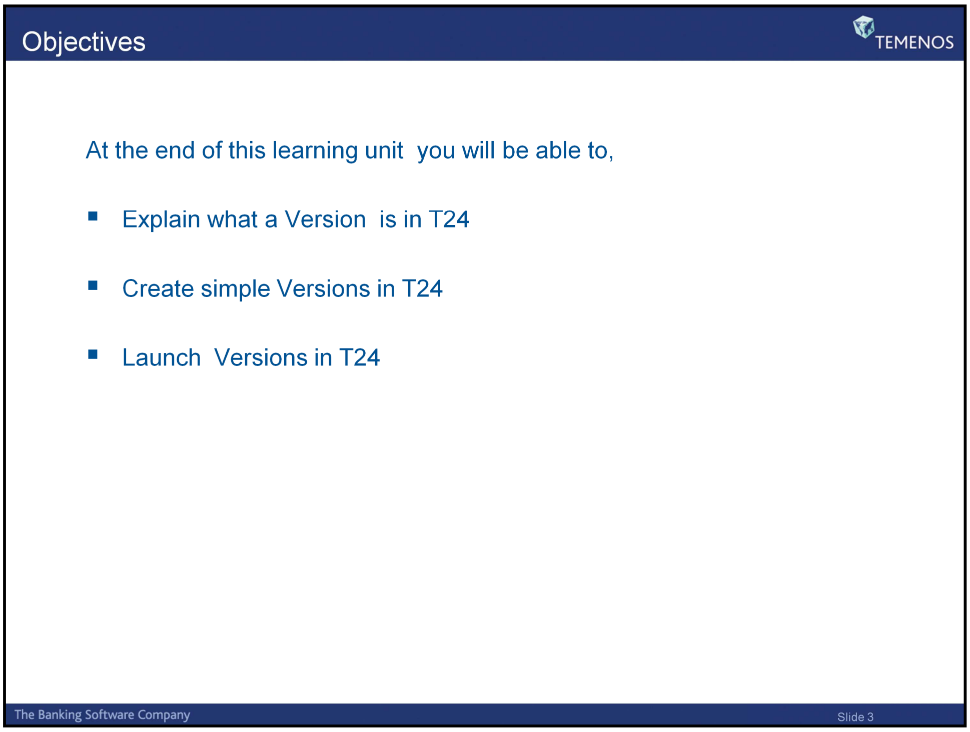
TEMENOS
The Banking Software Company

Welcome to the learning unit on Versions in T24 . In this learning unit you will learn about Versions in T24. You will also create simple versions in T24



The conventions used in this learning unit are,

1. Applications in T24 are represented in bold uppercase letters.
2. The fields in the applications are mentioned in uppercase letters.
3. This learning unit is example based. First an example output will be shown then you will be taught the steps to create the Version. At the end of each section a workshop is given to test your level of understanding.

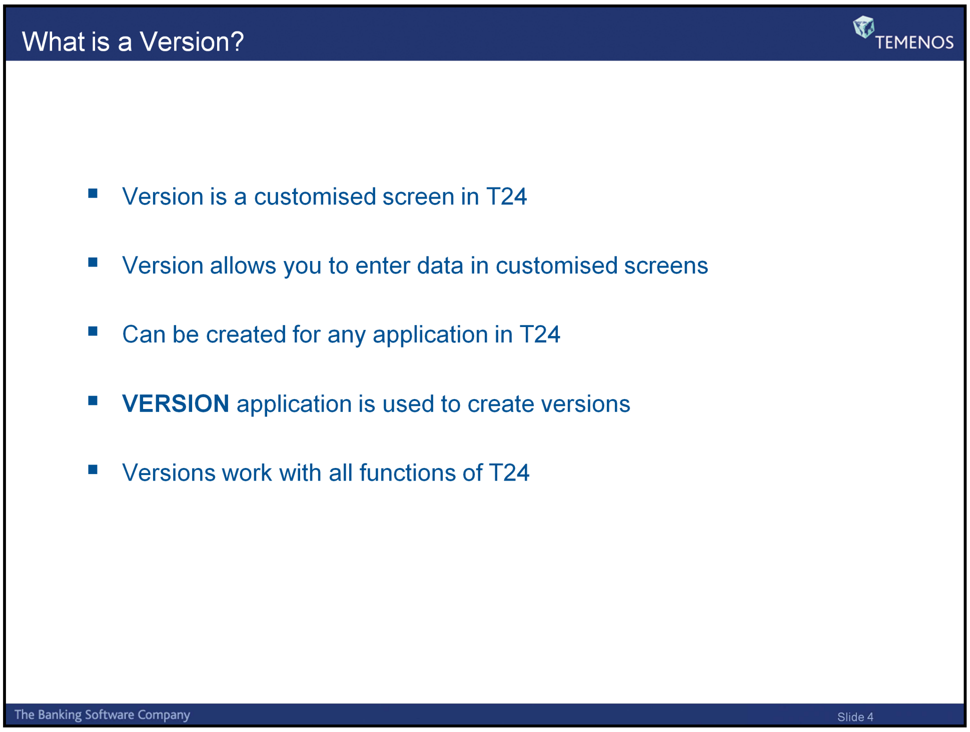


At the end of this learning unit you will be able to,

1.Explain what a Version is in T24.

2. Create simple Versions in T24.

3. Launch Versions in T24.

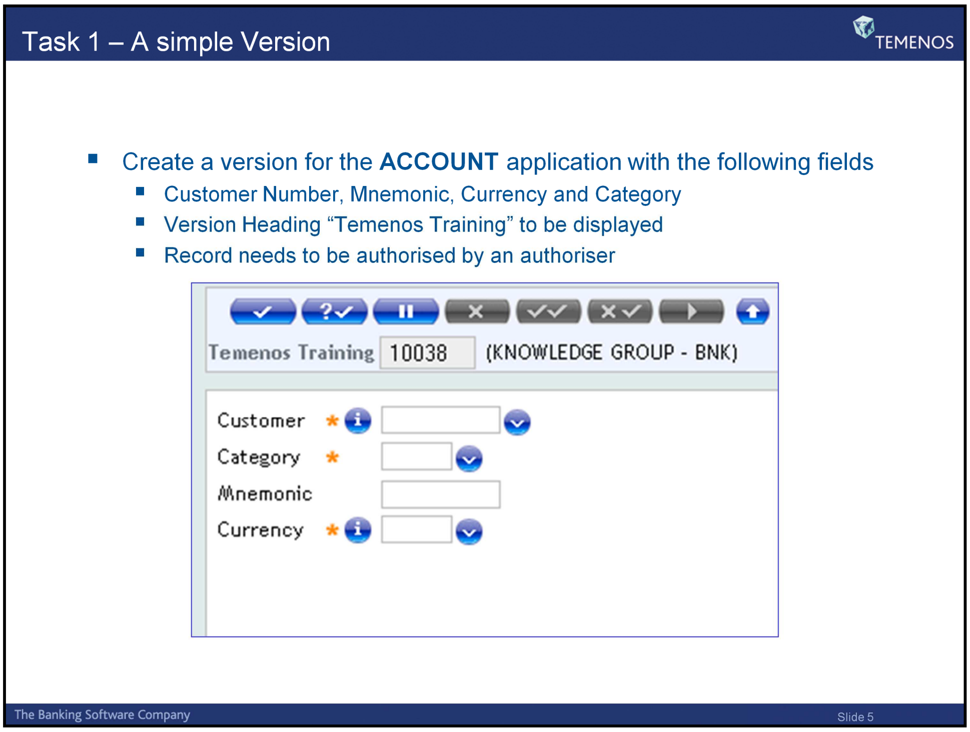


1,2 Imagine that you have to input a funds transfer transaction in T24. You will use the **FUNDS.TRANSFER** application, and the input function to do so. The screen that is displayed to you contains all the fields in the **FUNDS.TRANSFER** application some of which may not be important to the end user. Being the first time, you cannot figure out which fields to use and which not to. What if you had a customised screen that showed you only the fields that you require to input?

3,4 You can create customised application screens in T24. They are called versions. The **VERSION** application must be used to create customised application screens. Your version you can have just the mandatory fields in **FUNDS.TRANSFER** application.

5. Can you view data using versions?

Versions behave like the application itself, but only looks 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



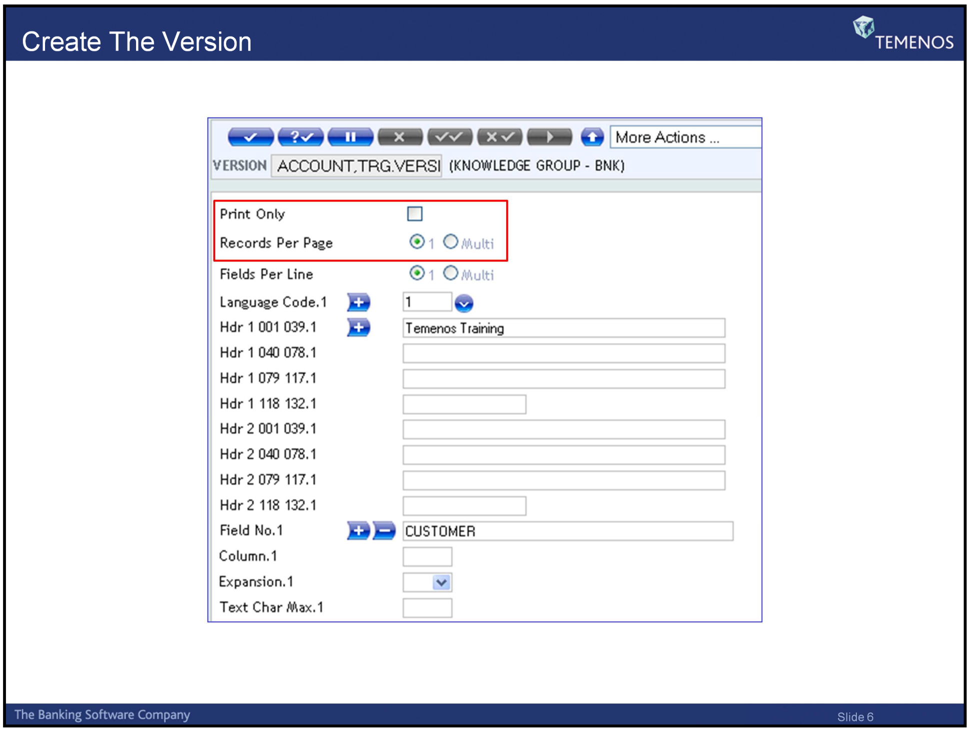
This task teaches you how to create a simple version of the **ACCOUNT** application in T24.

The requirements of the task are listed below:

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

2. The header “Temenos Training” need to be displayed.

3. The customer accounts created 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.

Are Versions only used for creating customised screens to be used for input? Versions are also used to define printed reports.

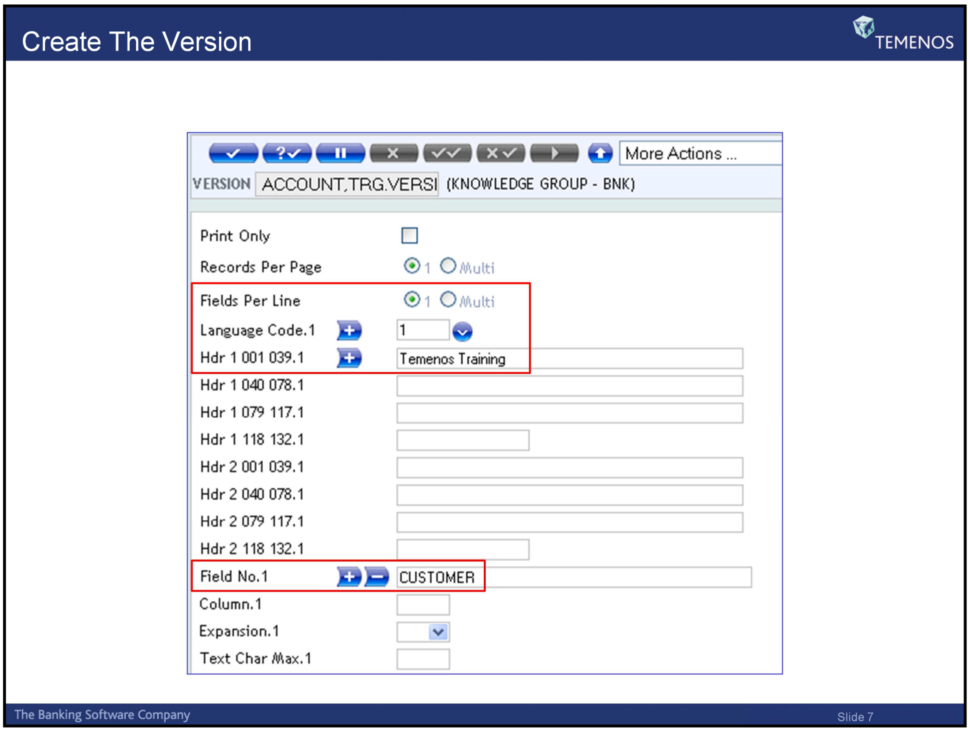
PRINT.ONLY - The first field indicates whether the version is created for printing reports. By default it is blank stating that the version is just created for a customised screen

Before creating a version you must decide what fields are going to be part of it. Ensure that you have all the mandatory fields of your 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 customised screens.

Do you want the version to display one record at a time or multiple records? Where do you specify this?

RECORDS.PER.PAGE –The RECORDS.PER.PAGE field has options that lets you to have one or multiple records per page. When RECORDS.PER.PAGE is set to one, one record will be displayed per page.

Note: If RECORDS.PER.PAGE is set to multi, multiple records will be displayed. This field works only with the DESKTOP.



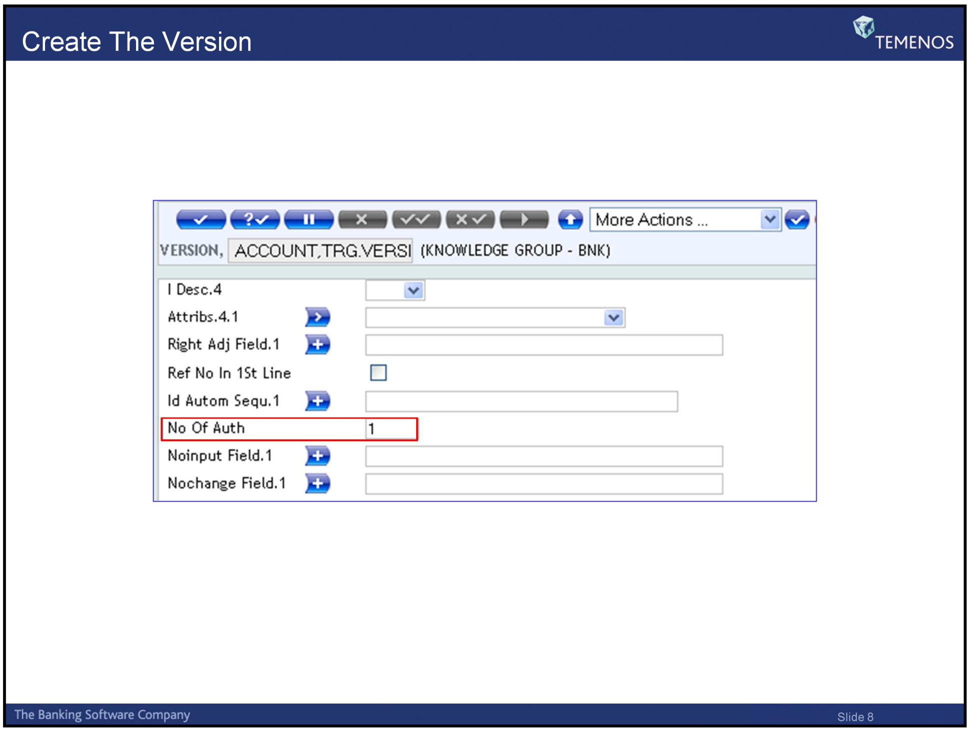
FIELDS.PER.LINE- Next how do you want the fields to be displayed – one below the other or on the same line? Specify this in the field FIELDS.PER.LINE. Choose one as the option for the field FIELDS.PER.LINE . This will make the fields to appear one after the other.

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

Where do you specify the headers for your version?

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.



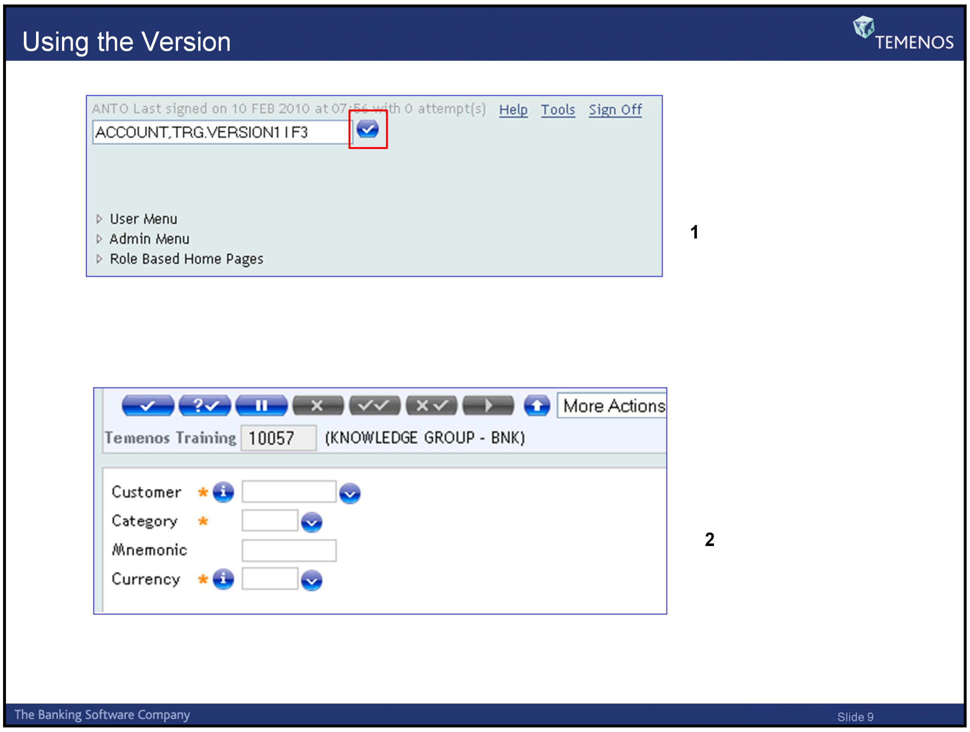
When using an application in T24 to create a new record, the record status is INAU when the record is first committed and then has to authorised. Now what if you want to skip the INAU stage? Will a T24 application allow you to authorise a record straight away? The answer is No. But a version offers you 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.

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 Version. Auto auth version is nothing but self-authorisation.

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

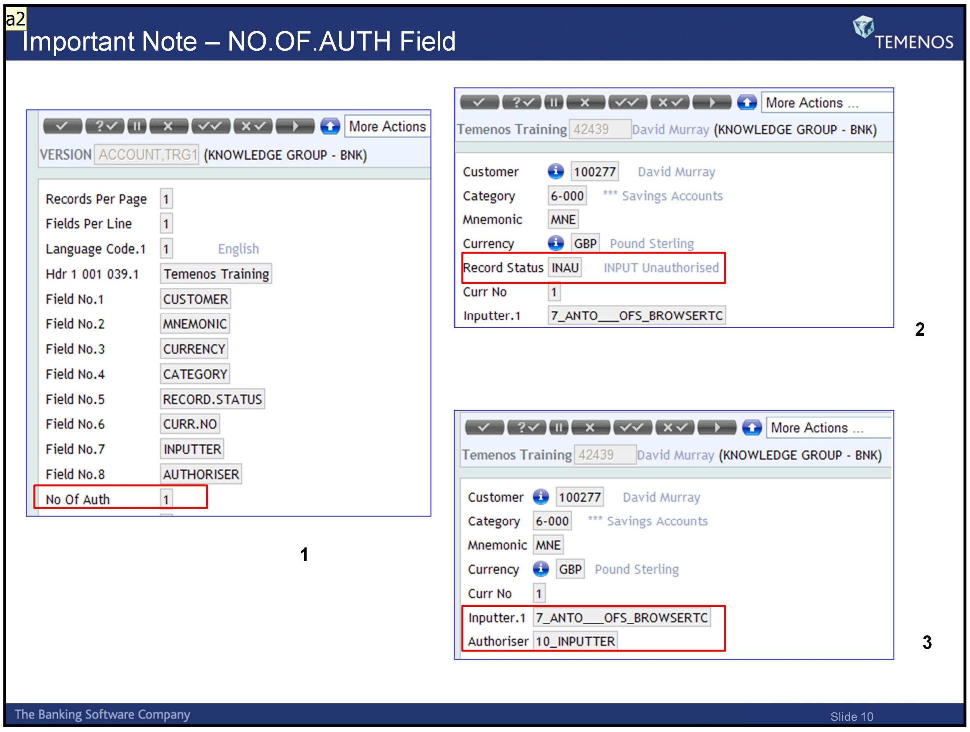
You have now created the customised screen. You must authorise your Version record next. It is now ready for use.



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

2. The version behaves like the under lying 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



1.In the **VERSION** record the NO.OF.AUTH field is set to 1(one) .

2.A record created using this version is shown. You can see that the record has moved to INAU status.

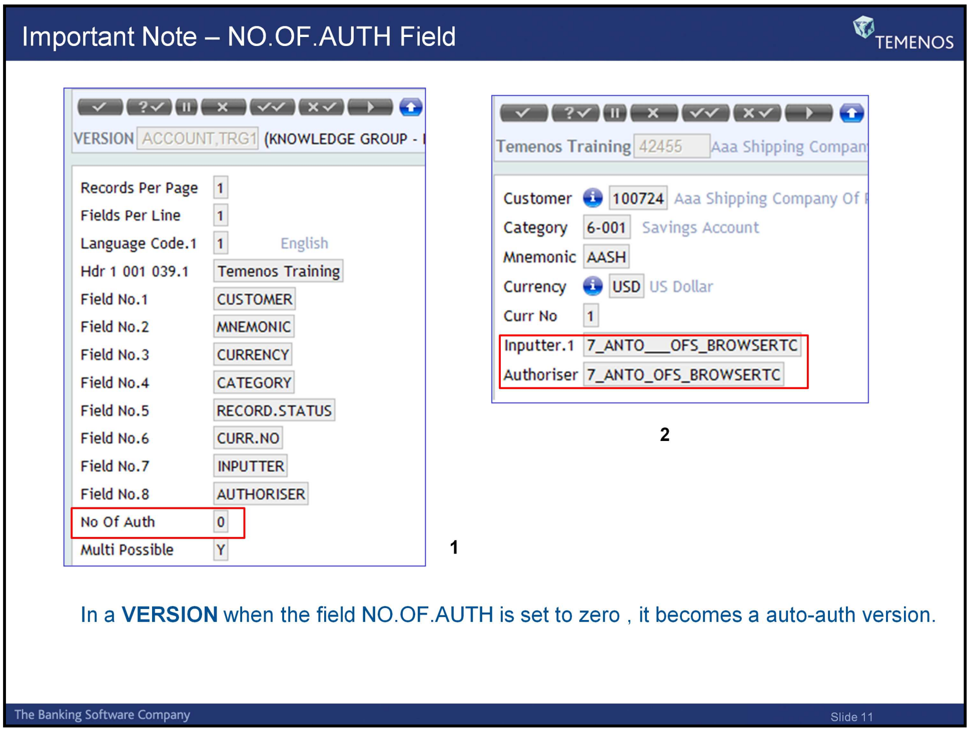
3.After you authorise the record it moves to LIVE status. You can see the name of the inputter and authoriser of the record.

Note : For those records in LIVE status the RECORD.STATUS field is not displayed.

Slide 10

a2 In the following 2 slides the functionality of NO.OF.AUTH field is explained with the values 1,0 & 2 set. The slides are animated ones.

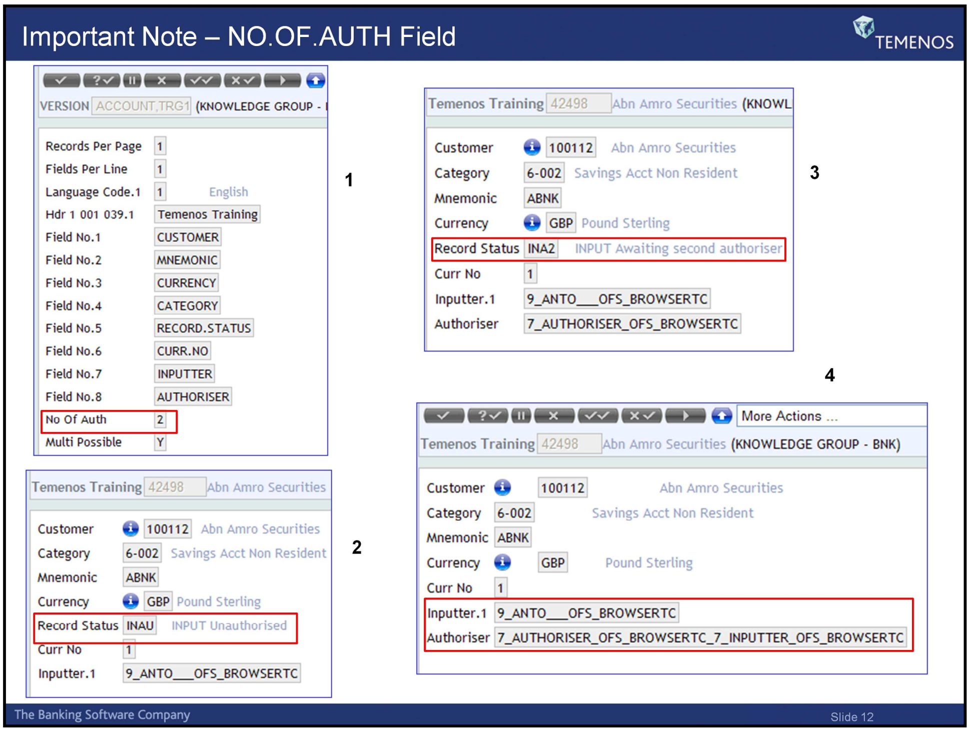
assispriya, 7/30/2008



1.Here the NO.OF.AUTH field is set to zero.

2.When you use this version to create records, the record immediately moves to LIVE status as soon as you commit the record. This is a auto-auth version.

Note: **VERSION**, (version comma) is an auto-auth version for creating versions in T24.



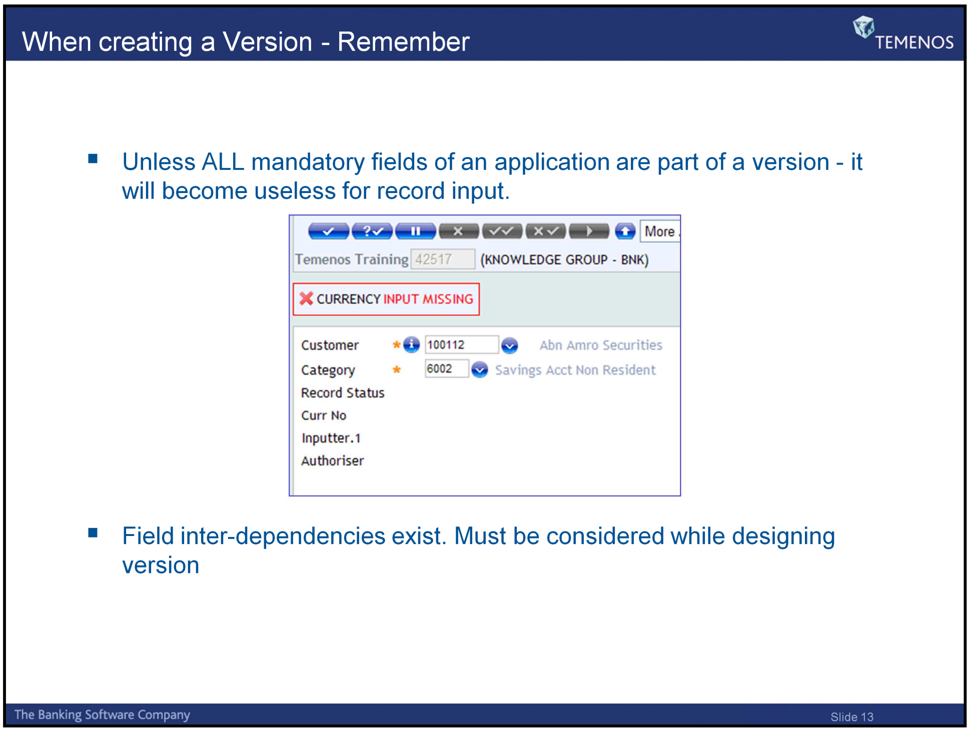
1. Here the NO.OF.AUTH field is set to 2(two)

2. Now when you create a record using this version it moves to INAU status.

3. Once you authorize the record, the record does not move to LIVE status. It moves to INA2 status.

4. When the record is authorized by a second authorizer, only then it moves to LIVE status.

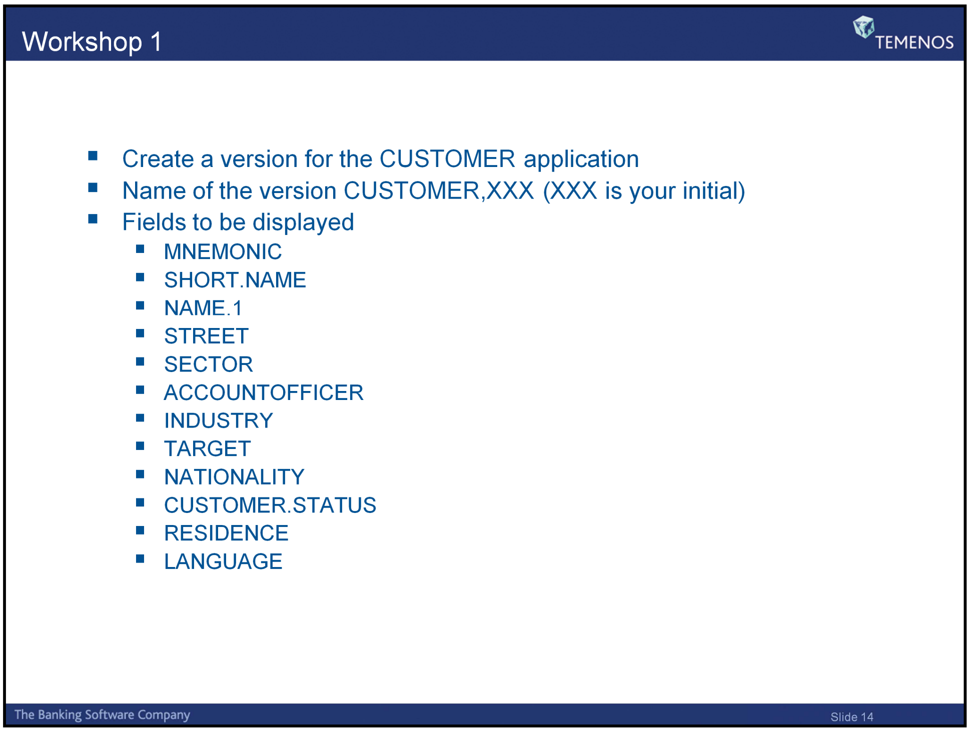
Note that the name of the second authorizer is appended in the field AUTHORISER.



1. Unless all mandatory fields of an application are specified in a version, a record cannot be committed using the version.

The example shows you the creation of an Account using a version. It displays an error message when you commit the record. CURRENCY is a mandatory field in the **ACCOUNT** application and that field is not part of the version used. If you miss out any of the mandatory fields in the Version, when you create records using the Version T24 does not allow you to commit the record. The version may be a customised screen but the underlying **ACCOUNT** application works the same way in T24.

2. Certain applications in T24 has inter-dependent fields. Depending on values entered in some fields, other fields are populated by the system. If the values defaulted can be edited, they must be part of the screen to enable a user to do so.



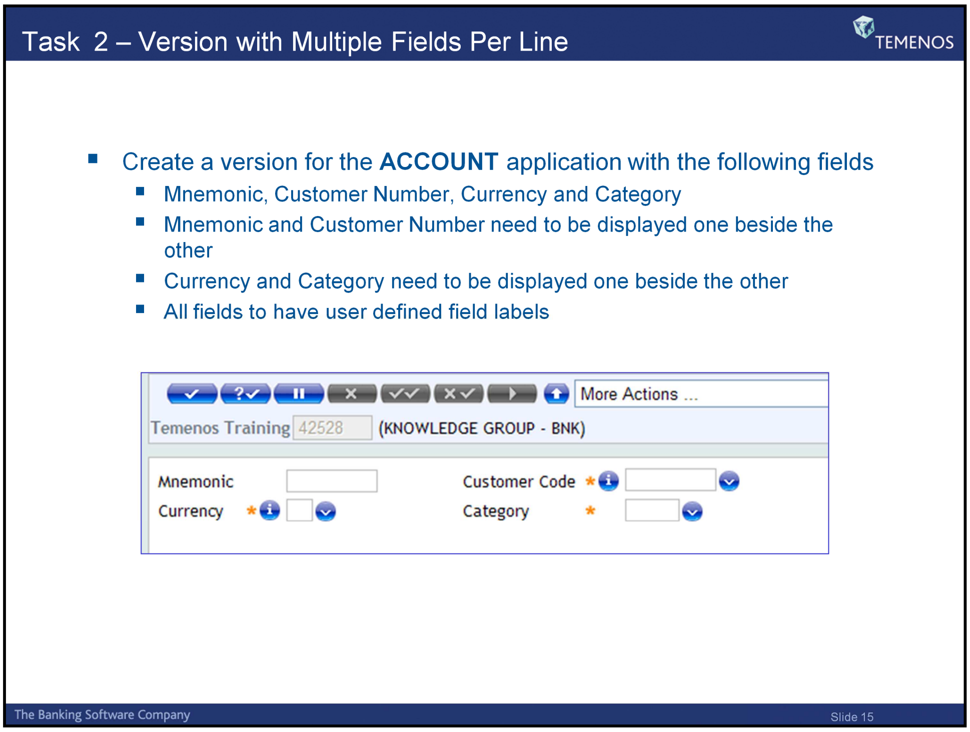
This is the workshop section of the learning unit.

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.

Refer Captivate -Version-WS1_demo.cp



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

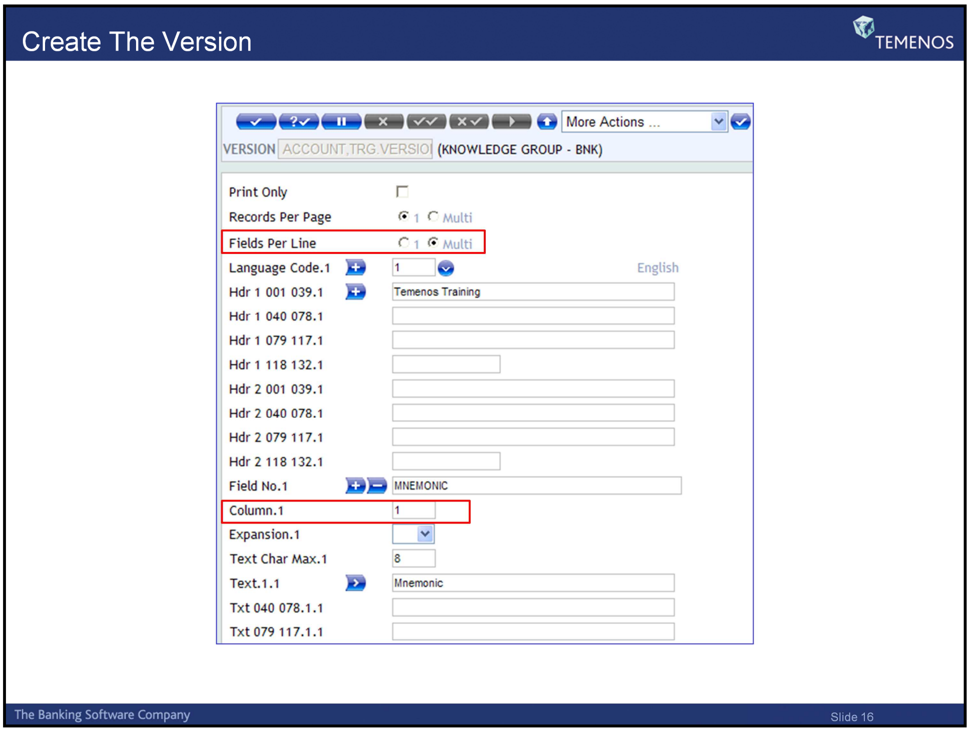
The requirement of the task is to:

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.

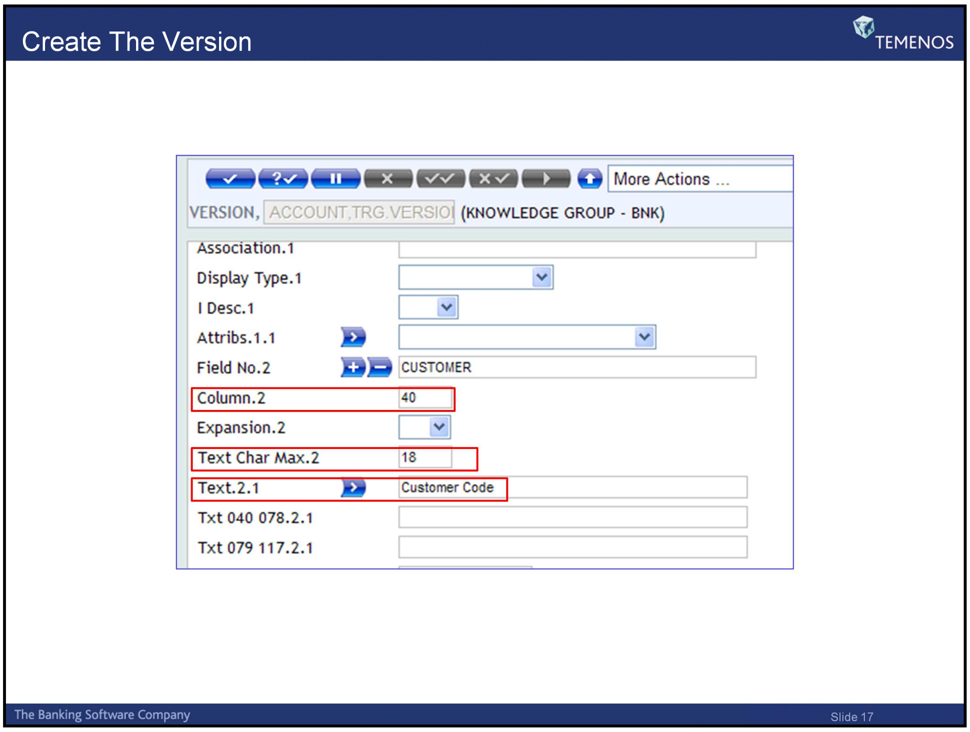
4. Labels must be user-defined.



FIELDS.PER.LINE – When set to MULTI enables the display of multiple fields on the same line. The fields MNEMONIC and CURRENCY are to be displayed on the same line.

Important Note: For muti-value fields the field name is appended with -1(hyphen one) in the field FIELD.NO. For sub-value fields append the field name with .1(dot one).

COLUMN - Value required only if FIELDS.PER.LINE is set to MULTI. The first field MNEMONIC will be displayed in column one. Also the field CURRENCY will be displayed in column one.

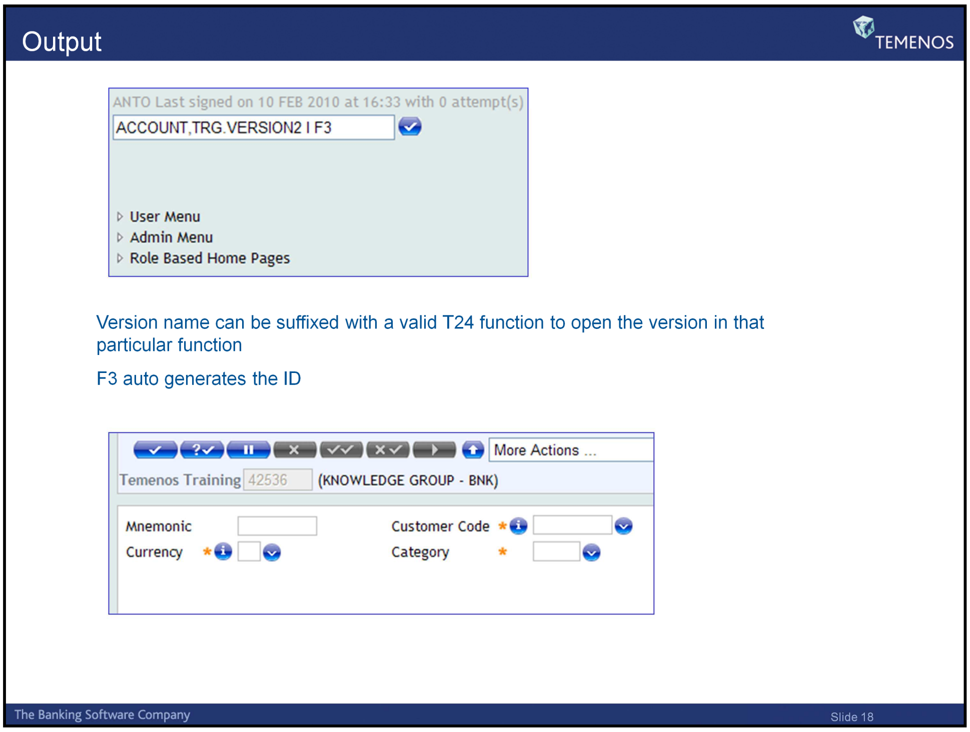


TEXT.CHAR.MAX – Defines the length of your label . The length defined here determines the total number of characters included in the TEXT Field which includes spaces also. In the example shown here, fields MNEMONIC and CURRENCY hold the value as eight(8) in the COLUMN field . This makes the fields to be aligned properly in the version.

TEXT - You have to specify user defined labels for the fields. Specify the label in the field TEXT. This is a sub-value field which can be expanded by itself to allow the label to be entered in the various languages defined in LANGUAGE.CODE field. If no Text is entered by the User, the Version screen will be presented without any Text for this specific field.

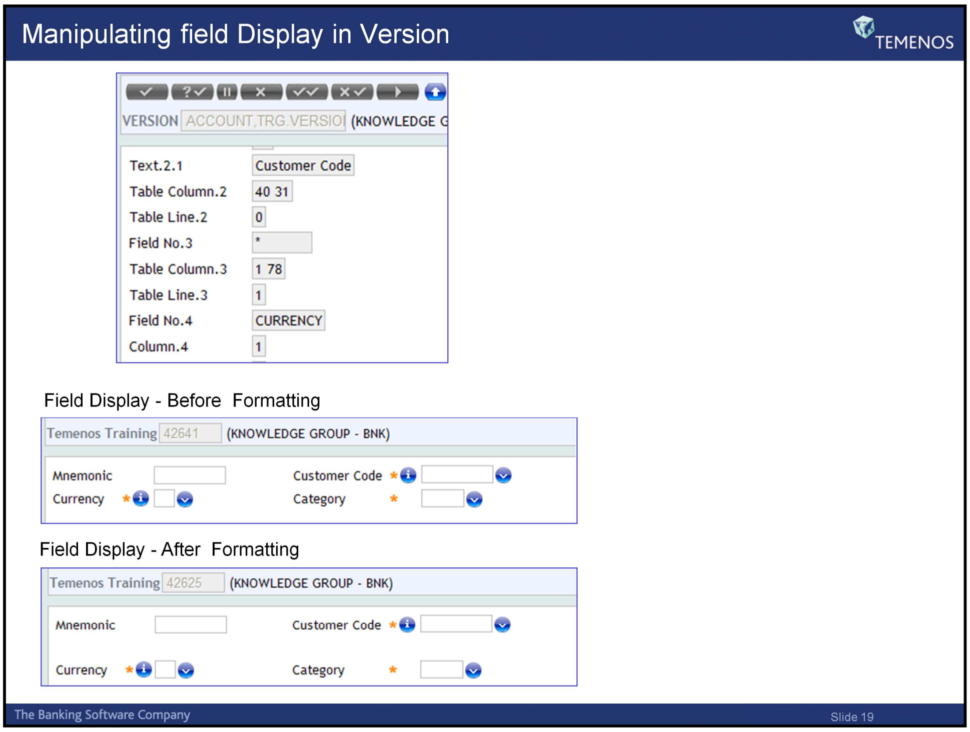
COLUMN - For the next field CUSTOMER COLUMN value is forty(40). Specify the other fields also.

Do not forget to authorise your version record.



1.Launch the version by specifying the version name in the command line. The version name can also be suffixed with a valid T24 function to open the version in that particular function mode. When you want to create a new account, you can specify ACCOUNT,TRG.VERSION2 I F3 where, I denotes the input mode and F3 for auto id generation.

2. You can see the new account created using the version.

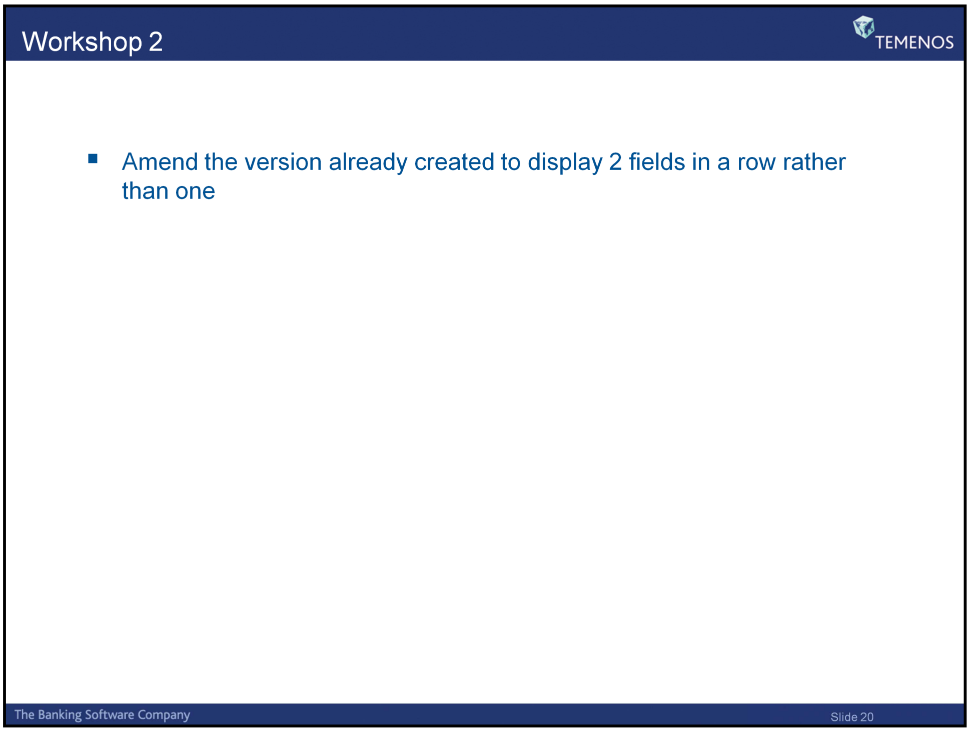


You can format the display of fields in version. You can have blank lines displayed between fields. The previous example is modified to just include a blank space between the fields.

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

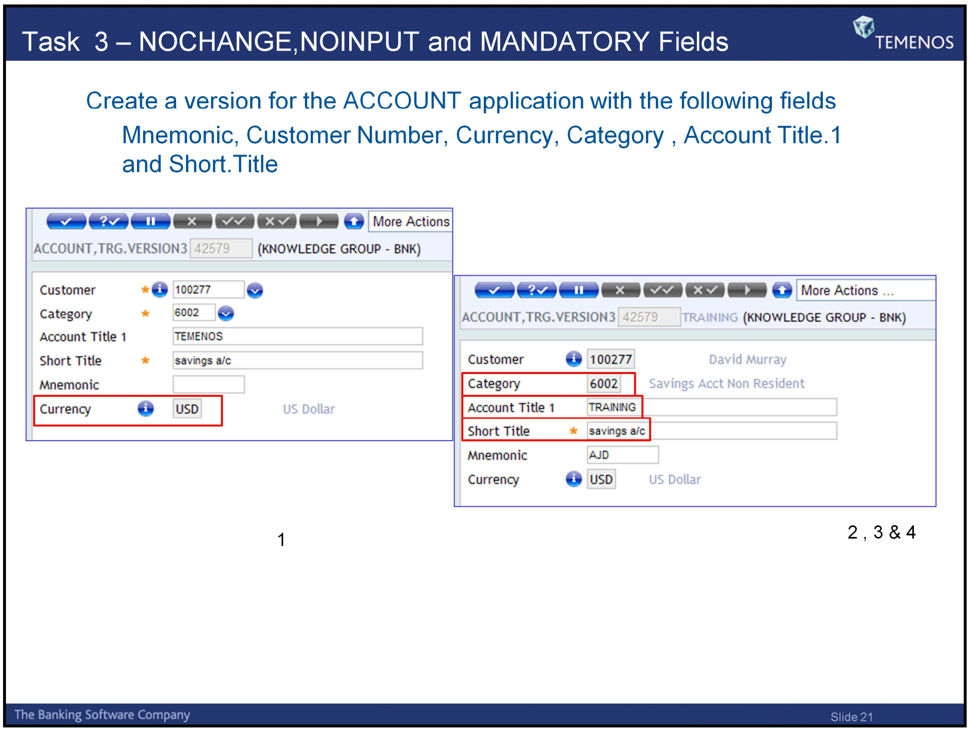
   Specify an asterisk (*) in the field FIELD.NO to get a blank line. Since this blank space is to be displayed before CURRENCY field , specify asterisk (*) before providing the values for CURRENCY field in the associated multi-value set.

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



1. Amend the version already created to display 2 fields in a row rather than one.

Refer Captivate -Version-WS2_demo.cp



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.

The bank wants to create a user friendly version that has the following features:

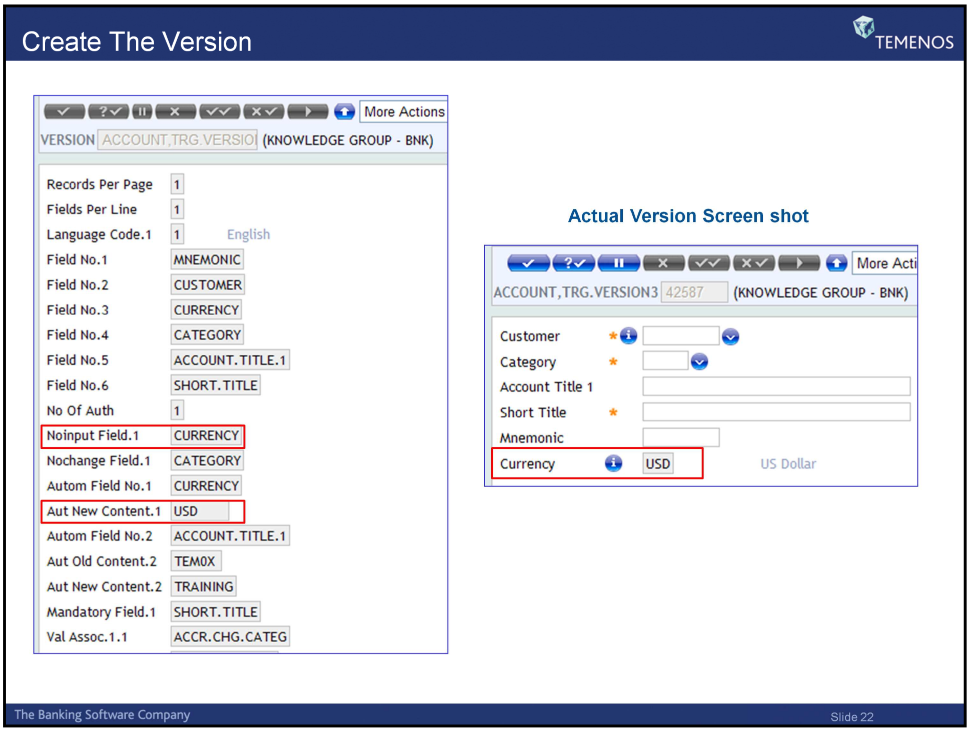
1.The CURRENCY field should be defaulted to USD.

2.Also the user should not be allowed to change the CATEGORY field ,once the record is authorised.

3. If there is an already existing value in the field ACCOUNT.TITLE.1 which begins with TEM, then a value “Training” should get defaulted overwriting the previous value.

4. The SHORT.TITLE field should be a mandatory field in the version even though it is not in the **ACCOUNT** application itself.

Can you design a customised screen for this requirement?



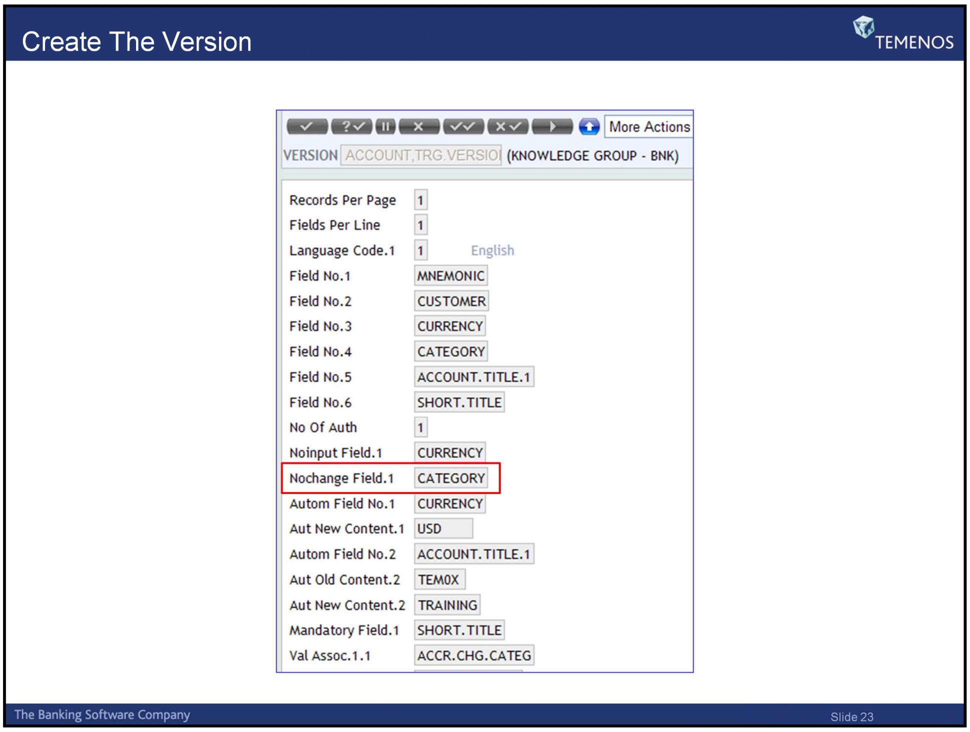
T24 allows you to default values into fields using the VERSION application. When you open a record with the version, the field will already have a value in it instead of being blank. To accomplish this you need to use the fields AUTOM.FIELD.NO and AUT.NEW.CONTENT.

AUTOM.FIELD.NO – Specify the name of the field for which you need to default the value.

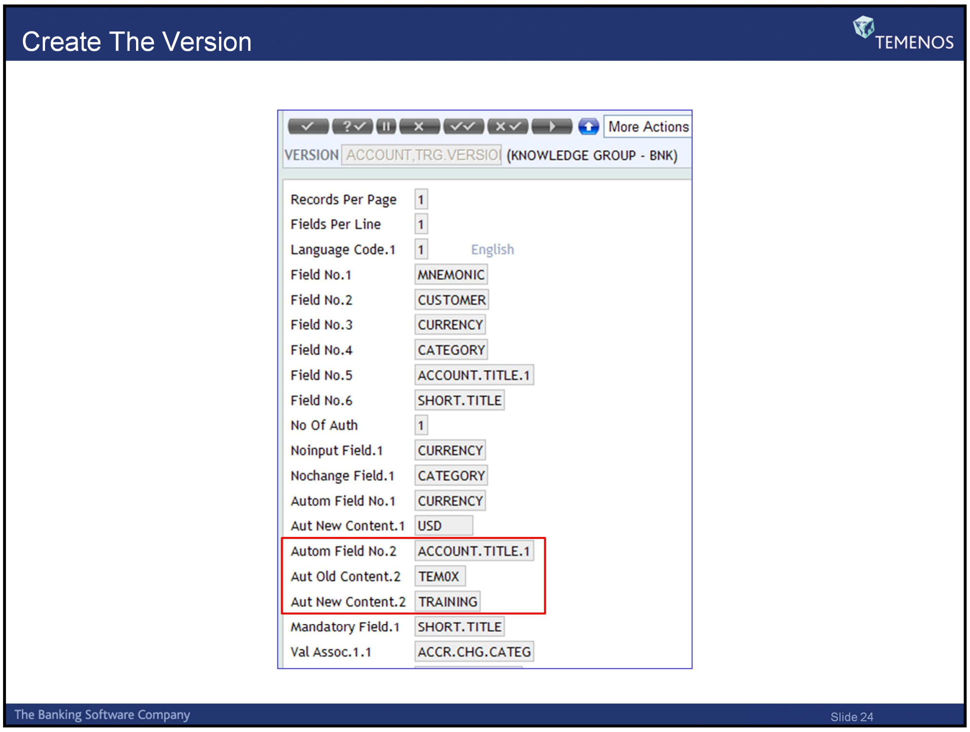
AUT.NEW.CONTENT – Specify the value that is to be defaulted here. (i,e) USD.

There is one more property that you may choose to set for such a field – NOINPUT. The NOINPUT property will prevent the users from inputting or amending the contents of this field. NOINPUT fields appear greyed out when displayed. It is a multi value field hence multiple fields can be made NOINPUT fields.

If you choose to make a mandatory field as a NOINPUT field, ensure that you default a value for that field else you will not be able to commit a record using that version.



You are required to ensure that the value in the CATEGORY field is not modified once the record is authorised. To do this you use the property of NOCHANGE fields. Specify CATEGORY as a NOCHANGE Field. Once the record is authorised fields defined here will appear greyed out when displayed. In short, NOCHANGE fields are fields that become NOINOUT fields after authorisation. Multiple fields can be displayed as NOCHANGE fields.



If the value in the field ACCOUNT.TITLE.1 begins with TEM, then it should default the value to TRAINING . How do you replace an existing value in a field? . To accomplish this you need to use the fields AUTOM.FIELD.NO , AUT.OLD.CONTENT and AUT.NEW.CONTENT. The value specified in AUT.OLD.CONTENT is replaced with the value in AUT.NEW.CONTENT. This works like the find and replace option in windows.

AUTOM.FIELD NO – Here you will specify the field name as ACCOUNT.TITLE.1.

AUT.OLD.CONTENT- Specify the old value in AUT.OLD.CONTENT as TEM0X where,

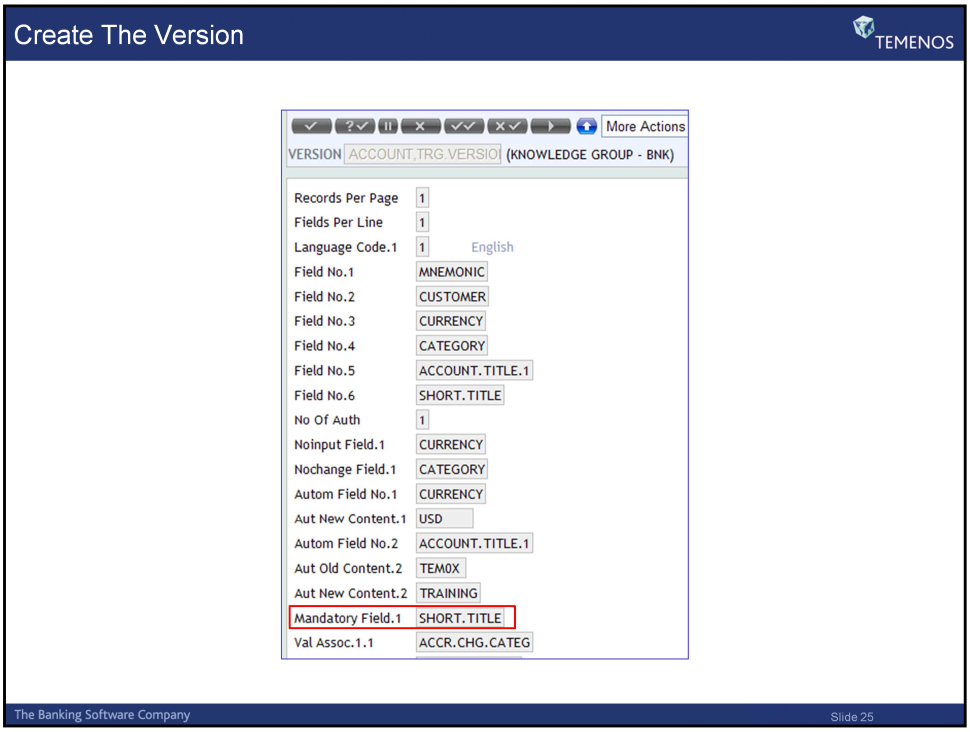
TEM is the starting string

0X is like a wildcard character.

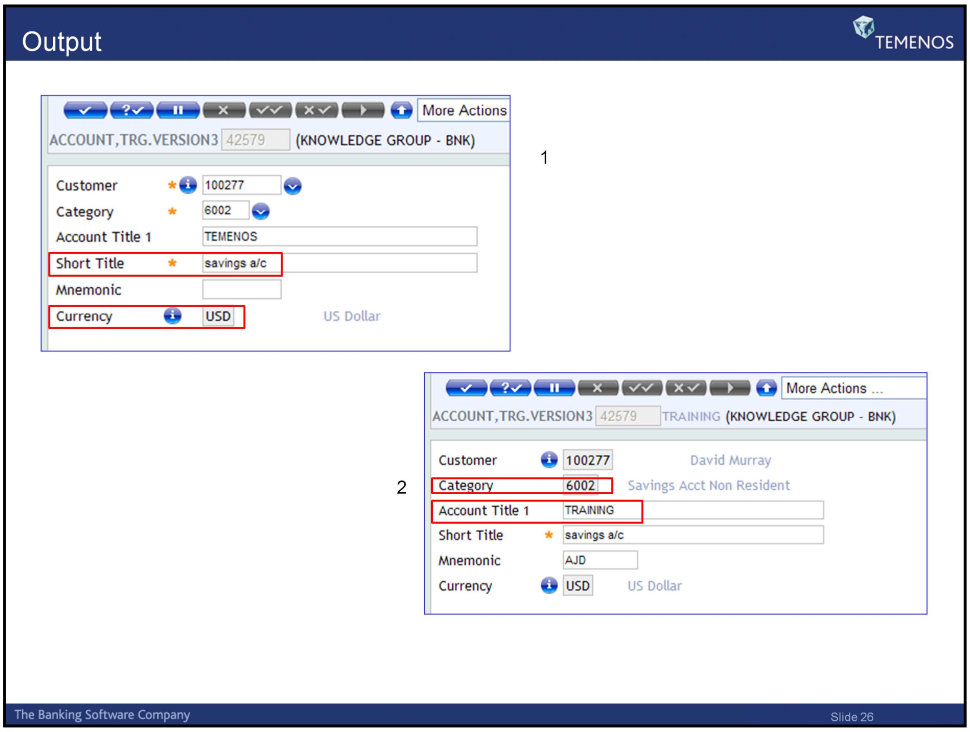
0X stands for any character and any number of characters.

AUT.NEW.CONTENT - Specify the new value as Training in the field AUT.NEW.CONTENT.

To default USD in the field CURRENCY no matter whether it is a new record or an existing record with a value in the field CURRENCY, specify 0X in the field AUT.OLD.CONTENT .



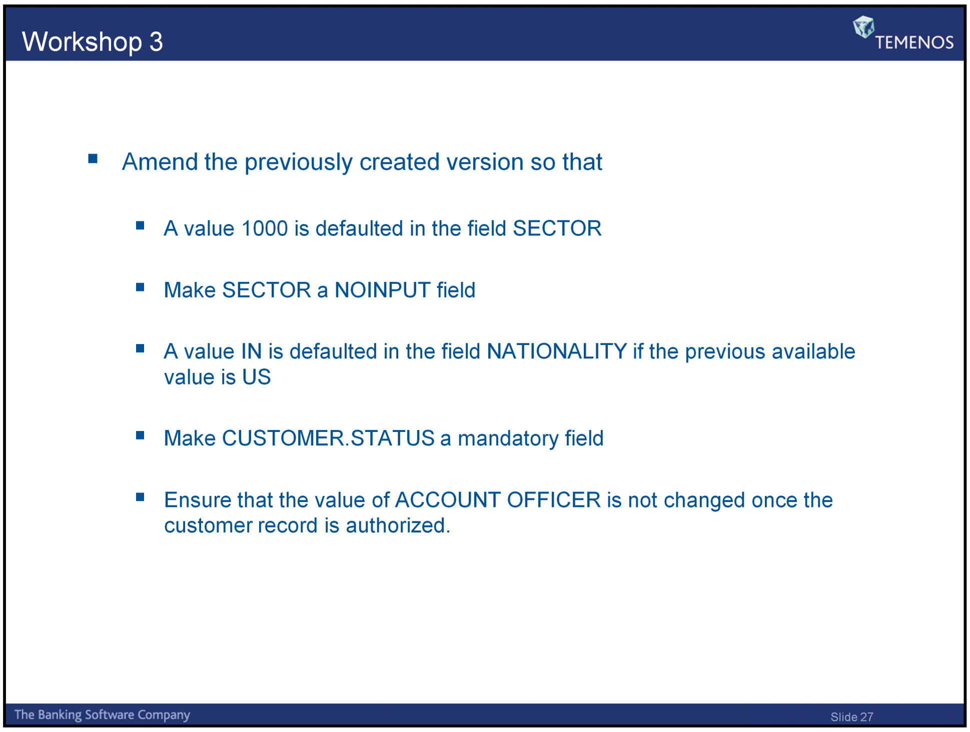
According to the task you have to make SHORT.TITLE as a mandatory field. To make a field as mandatory you specify the field name in the MANDATORY.FIELD of your version. This forces the user to enter value in that field else he will not be able to commit the record. A non mandatory field in the application can be made a mandatory field in a version. Vice versa is not possible.



Here is the output,

1. When you use the version to create an account record, you could see that the value for the field CURRENCY is defaulted as USD . SHORT.TITLE is a mandatory field in the version.

2. The field CATEGORY has become a NOCHANGE field . The value for the field ACCOUNT.TITLE has been modified from TEMENOS to TRAINING .



1. Amend the previously created version so that

    1.1 A value thousand(1000) is defaulted in the field SECTOR

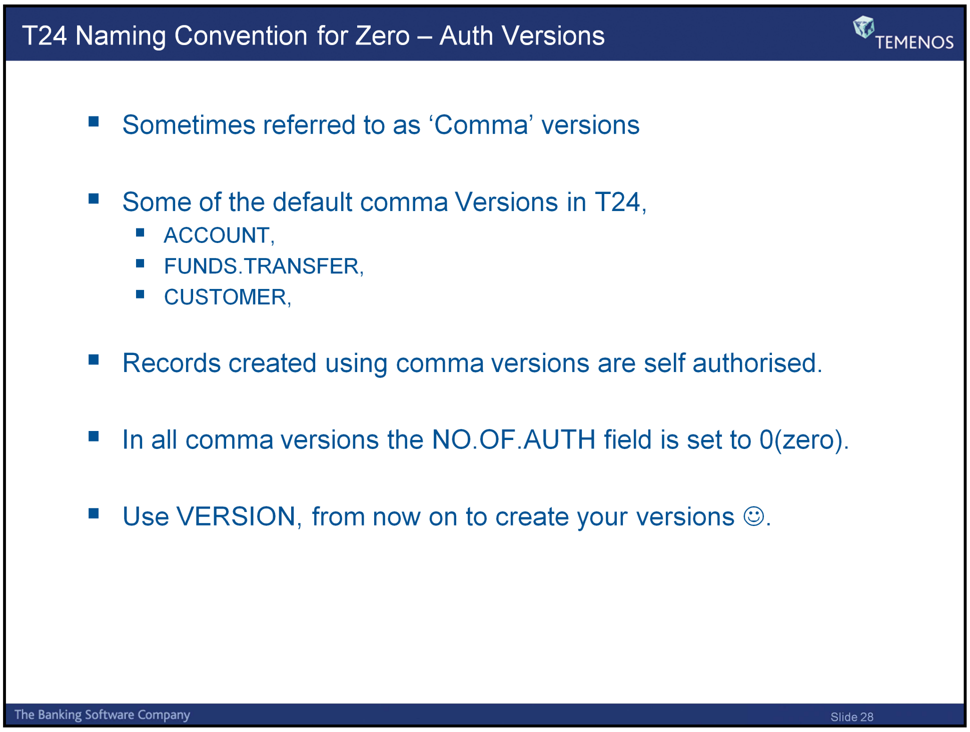
    1.2 Make SECTOR a NOINPUT field

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

    1.4 Make TEXT a mandatory field

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

Refer Captivate -Version-WS3_demo.cp



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). These Versions does not contain any fields.

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 .

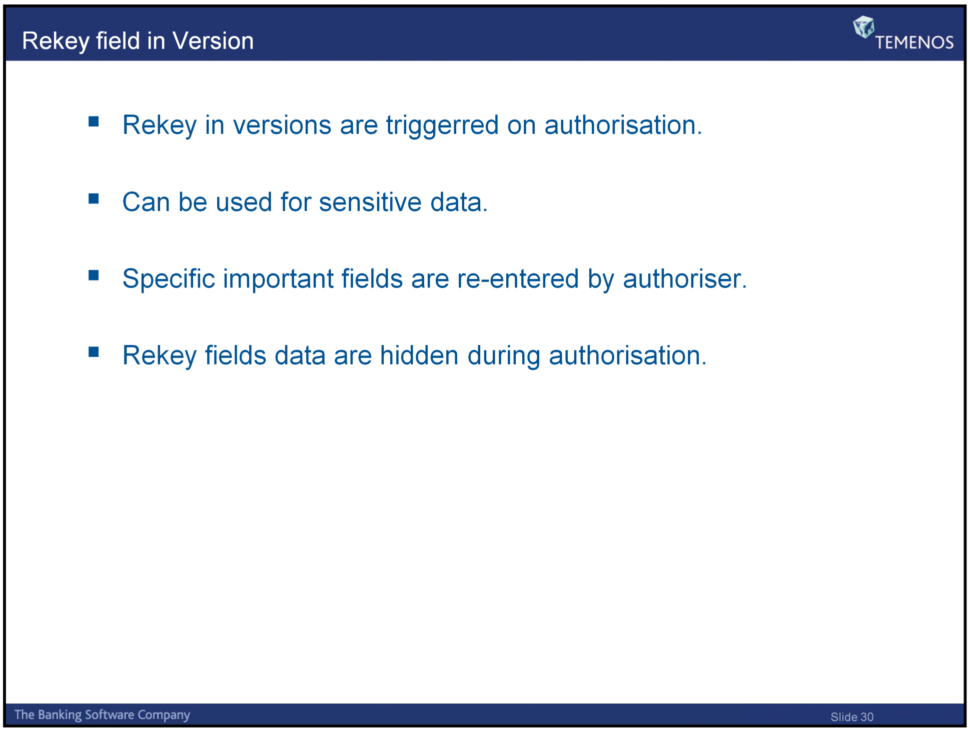
Quiz

TEMENOS

- Version is a customised screen in T24 – True / False
- The field NO.OF.AUTH is set 2 in the version, now when you authorise a record created using the version, it moves to LIVE status – True / False
- If the value for the field FIELDS.PER.LINE is set to one in the version, the COLUMN field becomes mandatory – True / False
- In version values can be defaulted into fields using the field AUT.NEW.CONTENT – True / False
- The value of a NOCHANGE field can be amended for a LIVE record - True / False

The Banking Software Company Slide 29

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

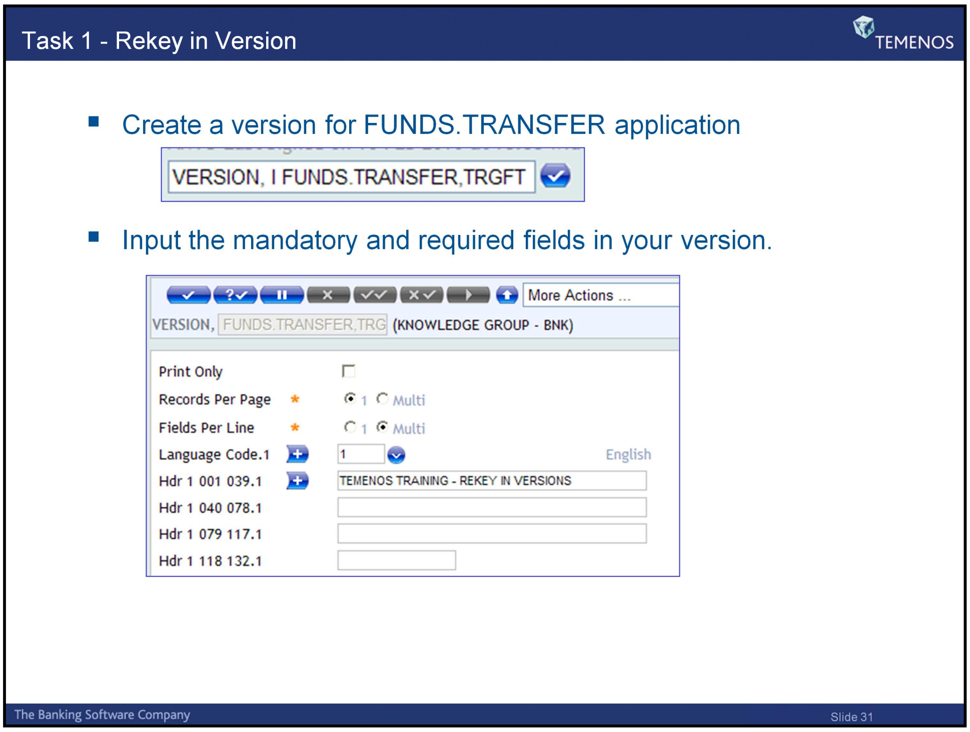


1. The Rekey field is used in Version at the authorisation level.

2. This enables authorisers to re-enter the data in specific fields so as to re-confirm the entry done 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 can prompt the Authoriser to re-enter the data for confirmation. If the data entered by the Authoriser does not match with the Inputter, then the record does not get authorised.

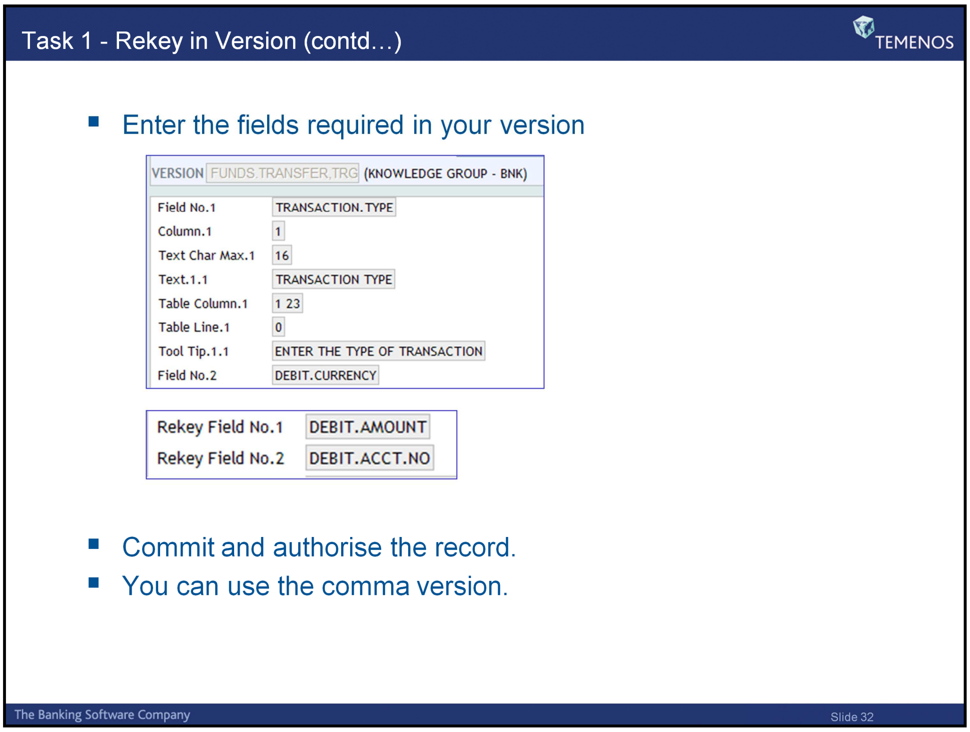
4. The data for the rekey fields are hidden during authorisation and the Authoriser needs to compulsorily re-enter the information.



This task helps in understanding the Rekey feature in version. Lets take an example of FUNDS.TRANSFER application.

1. To create a version for funds transfer, you need to input a command as: VERSION, FUNDS.TRANSFER,<VERSION NAME> for ex: VERSION, FUNDS.TRANSFER,TRGFT in our case

2. Input the mandatory and required fields.



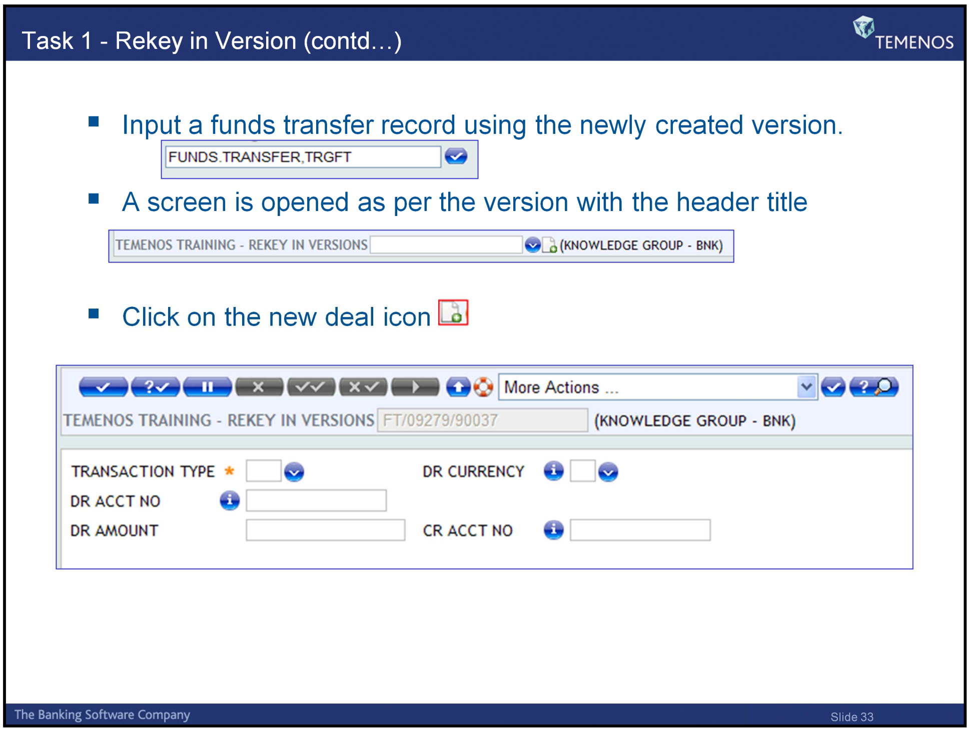
Enter all the fields required in your version.

Field No.1: The name of the field in the FUNDS.TRANSFER application.

Text.1 : The name as it should appear in the version.

Tool Tip: On pointing the mouse the field description is displayed.

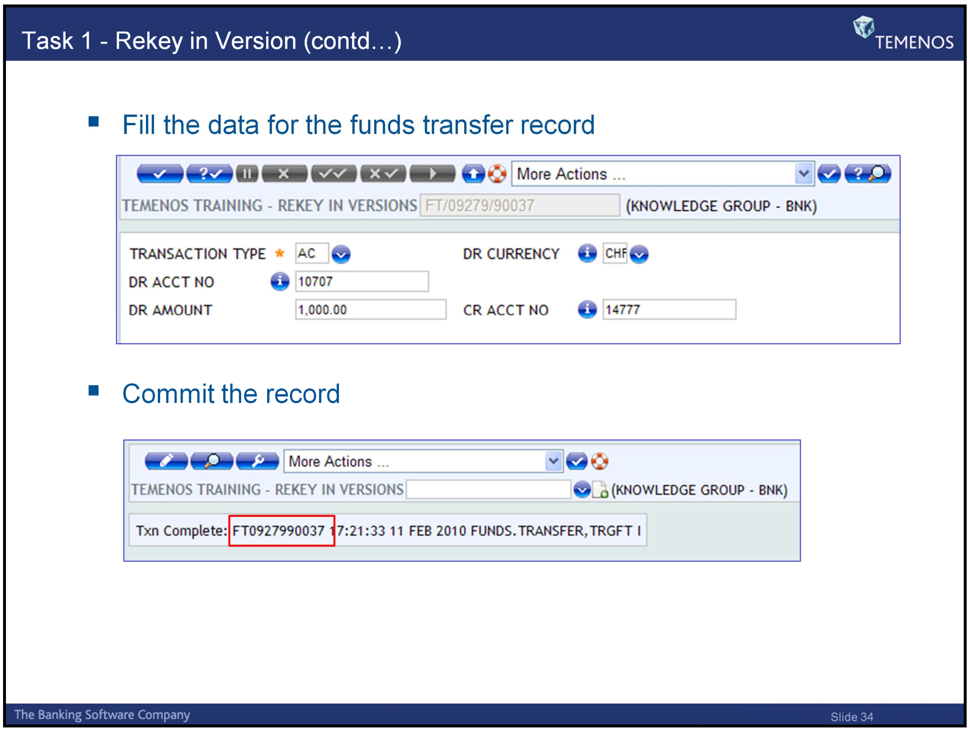
Rekey Field: These are the fields where the system will prompt the authoriser to re-enter data for confirmation.



1. Create a funds transfer record by typing the command FUNDS.TRANSFER,TRGFT

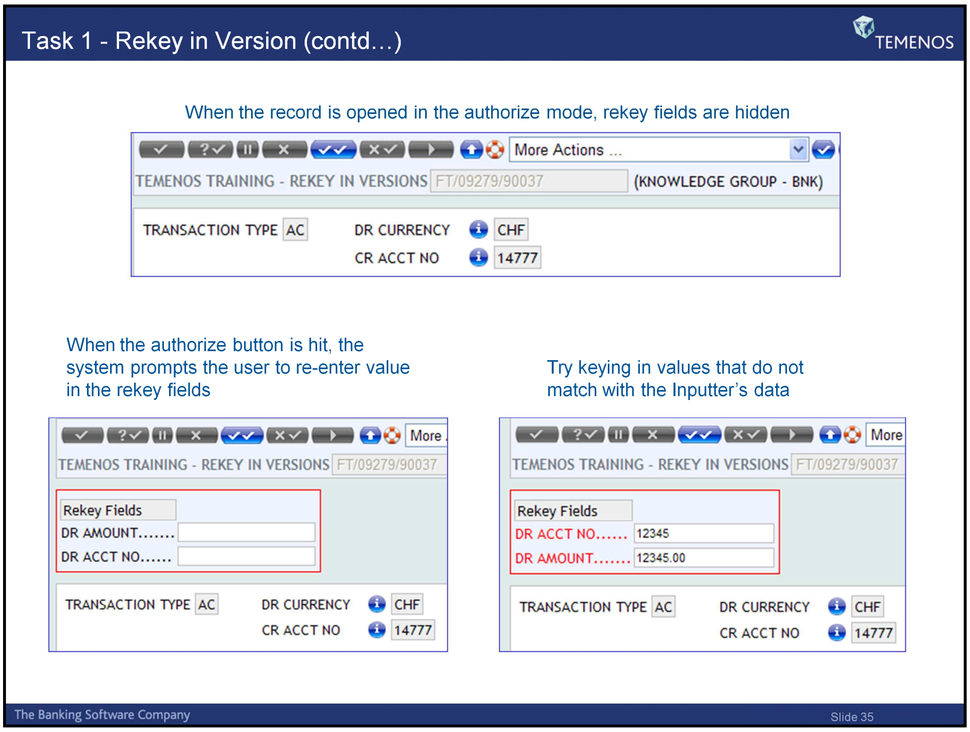
2. A new funds transfer application using the version is opened.

3. Click on new deal icon and the fields as required in the version application alone is displayed.



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

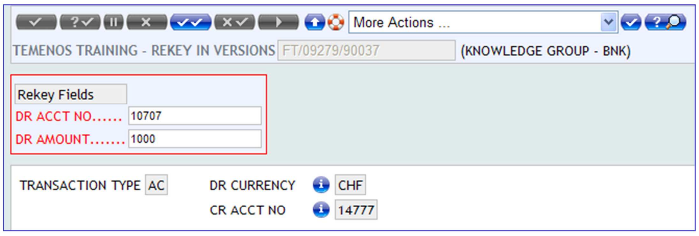
2. Note the record key (ID) of the Funds Transfer that was created.



1. When the record is opened in authorize mode the rekey fields are hidden, although these fields are visible when the record is opened in the SEE mode.
2. Hit the authorize button and the system will prompt the user to re-enter value in the rekey fields.
3. If the data entered by the Authoriser does not match with the Inputter data, the record does not get authorised. The rekey fields are marked in red to denote incorrect data. No error message are displayed at this point.

Task 1 - Rekey in Version (contd…)

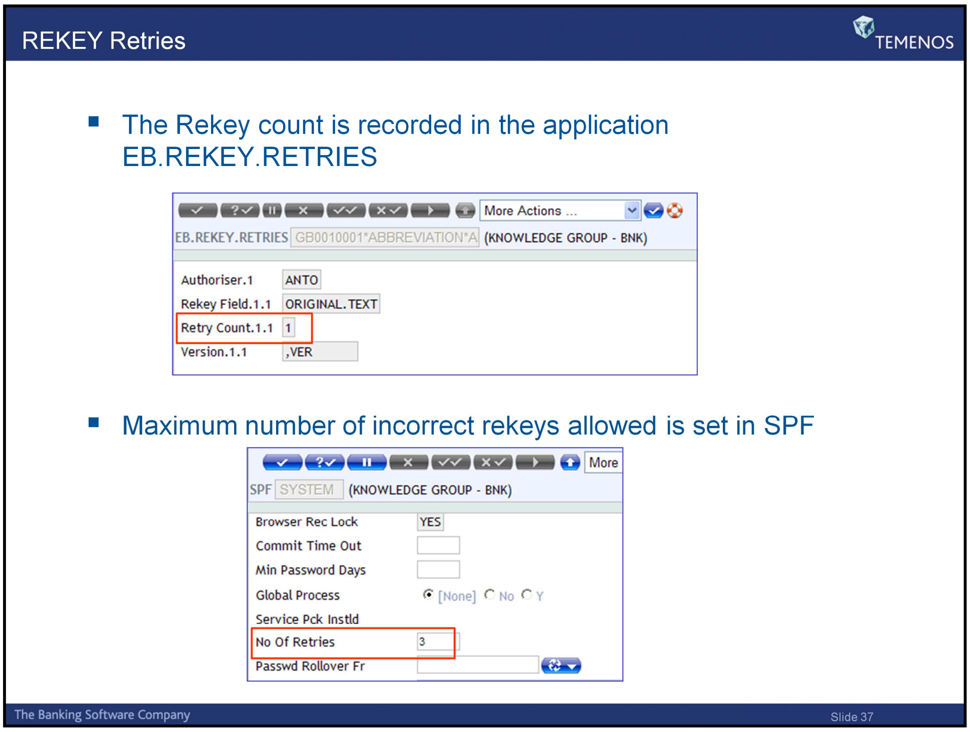
- Login as another user to authorise the record.
- Click on the Authorise link and the rekey fields are displayed



The Banking Software Company
Slide 36

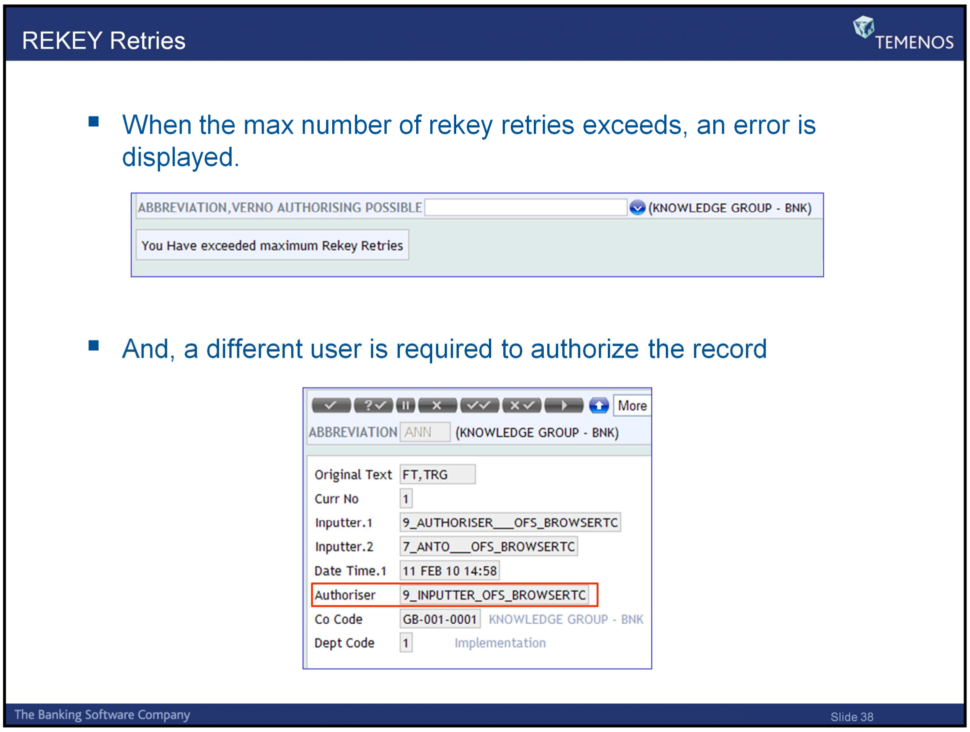
1. Login as a different user to authorise the funds transfer record.

2. On Authorisation the system prompts the user to rekey the data for the fields that were set to be rekeyed. The fields that need to be rekey are actually hidden from the Version screen. Only if the data reentered matches with the one that was already input by the inputter the record get authorised.

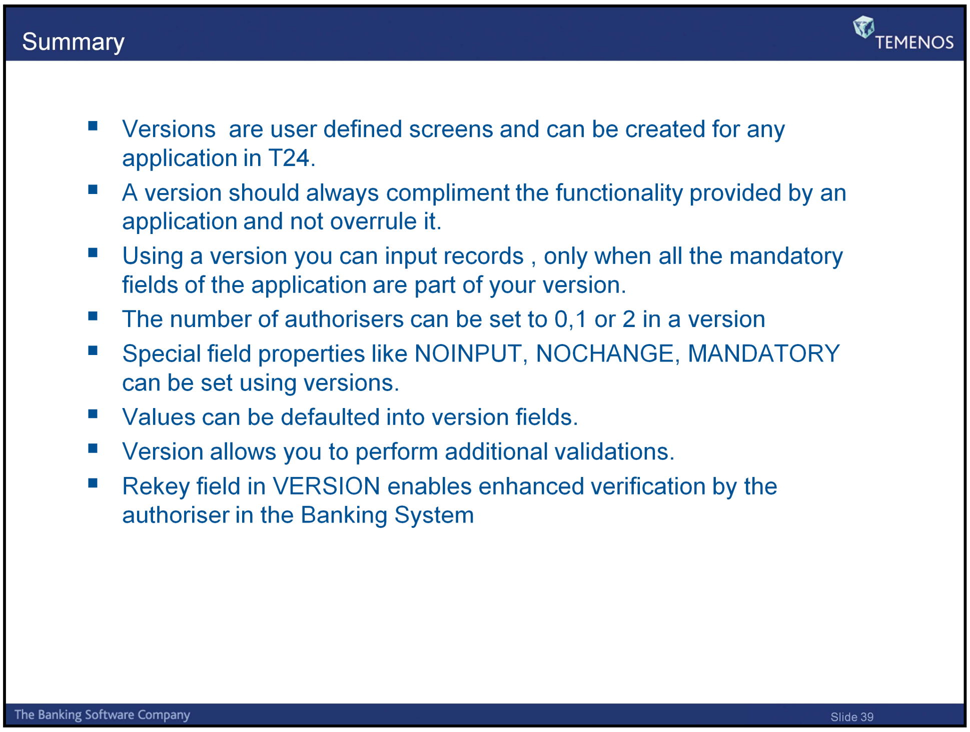


1. What happens when a USER who is trying to authorize the record enters incorrect rekey values multiple times ? The system records the number of incorrect rekey attempts by a same authorizer, in a file called EB.REKEY.RETRIES. A record is created in this file whenever a version that contains a rekey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

2. 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. 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.
2. Hence, a different authorizer is required to authorize the same record.



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

2. A version should always compliment 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.

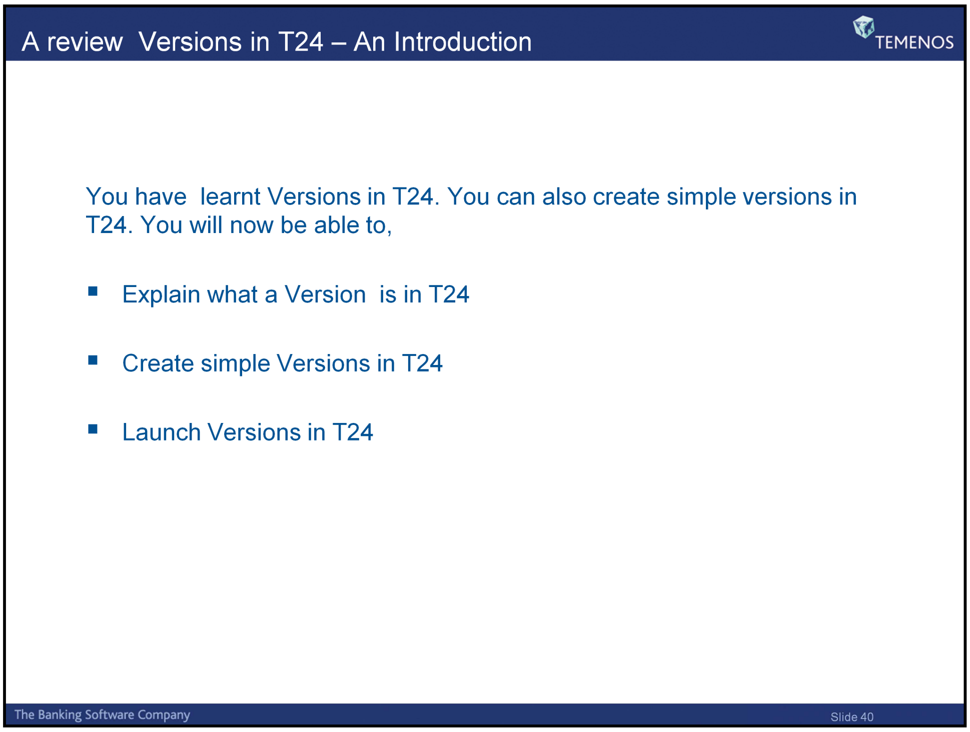
4. The number of authorisers can be set to 0,1 or 2 in a version.

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

Version History

TEMENOS

Version	Date	Author	Reviewer
1.0	2004	Alagammai Palaniyappan	
1.2	Dec 2006/Jun 2007	Alagammai Palaniyappan	
1.3	03rd Feb 2008	Nasim Kazi S	
1.4	June 2008	Assis Priya	Sara Cleur
1.5	Aug 2009	Vimal Menon	Kripesh Krishnadas

The Banking Software Company Slide 41

Thank You

TEMENOS EDUCATION CENTRE

TEMENOS
The Banking Software Company

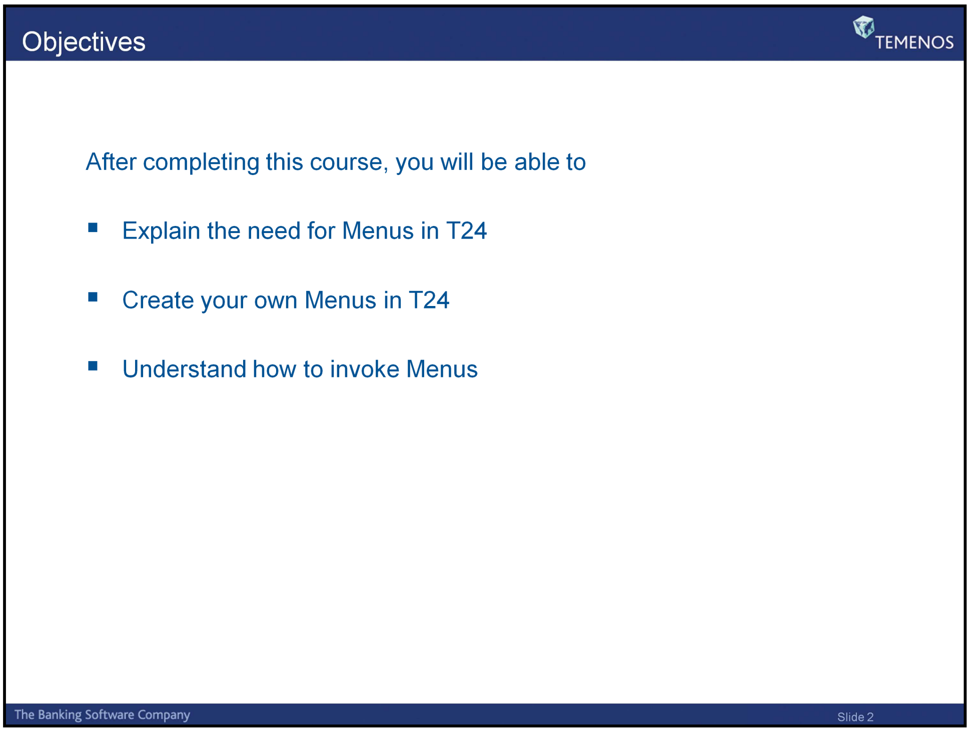
Introduction To Menus

TEMENOS EDUCATION CENTRE

TEMENOS

The Banking Software Company

Welcome to the Menus in T24 learning unit. In this learning unit, you will learn about how to create and customize T24 menus.



After completing this course, you will be able to

1. Explain the need for Menus in T24
2. Create your own Menus in T24
3. Understand how to invoke Menus

Need for Menus in T24

TEMENOS

- Easy navigation
- Assign different menu structure for different users
- Group similar tasks for ease of use.

The Banking Software Company

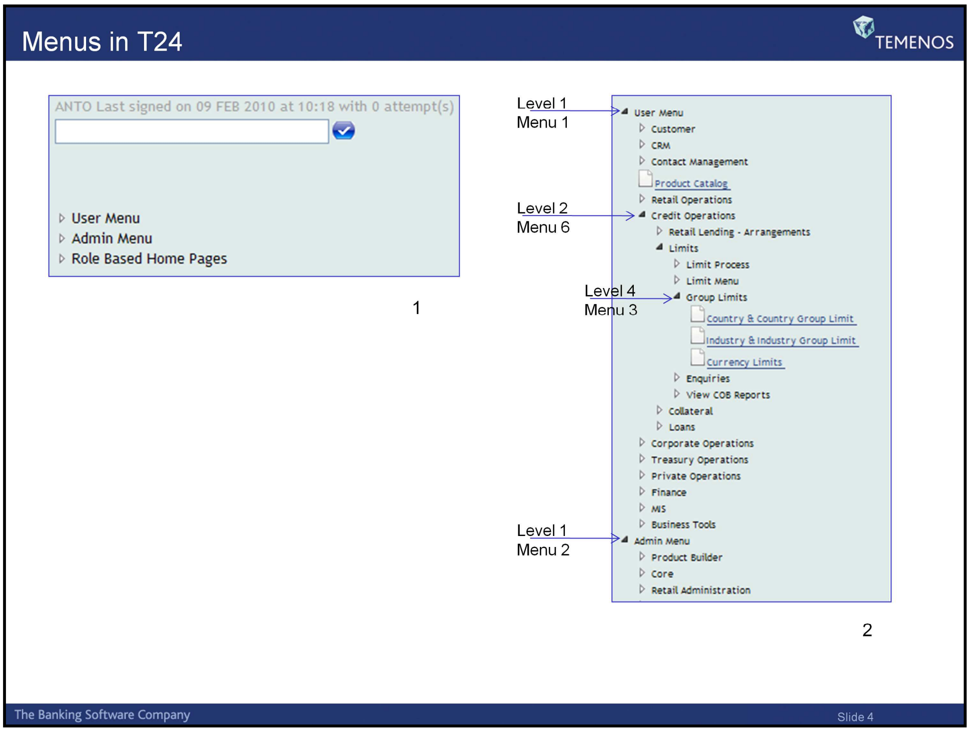
Slide 3

Assume you had an enquiry with a long name, say TRG.ENQ.CUSTOMER.DETAILS. To launch it, you would have to remember the name correctly and type it all in the command line. There could be several such enquiries, versions and even application names like **LD.LOANS.AND.DEPOSITS**, **FUNDS.TRANSFER** in T24. It would be easier for an end user if they did not have to remember all these application names. As soon as you logged in to T24, if all you need to access is readily available on a menu, that would help immensely. This is what T24 menus are all about.

1. Novice users don't need to remember application names and can easily navigate using menus to accomplish their tasks.

2. T24 allows you to create multiple menus according to each users work profile

3. Related tasks can be grouped together using the menus and submenus.



1. The illustration shows how a typical menu appears in T24.

   The highest level in this menu structure has three submenus namely **User Menu**, **Admin Menu** and **Role Based Home Pages**. The menus can be expanded by clicking on the menu.

   All user related tasks are grouped under the user menu and administration tasks are grouped under admin menu. This helps the user to navigate easily.

2. When the user expands a menu item by clicking on the menu, the submenus are listed for each of the top level menus.

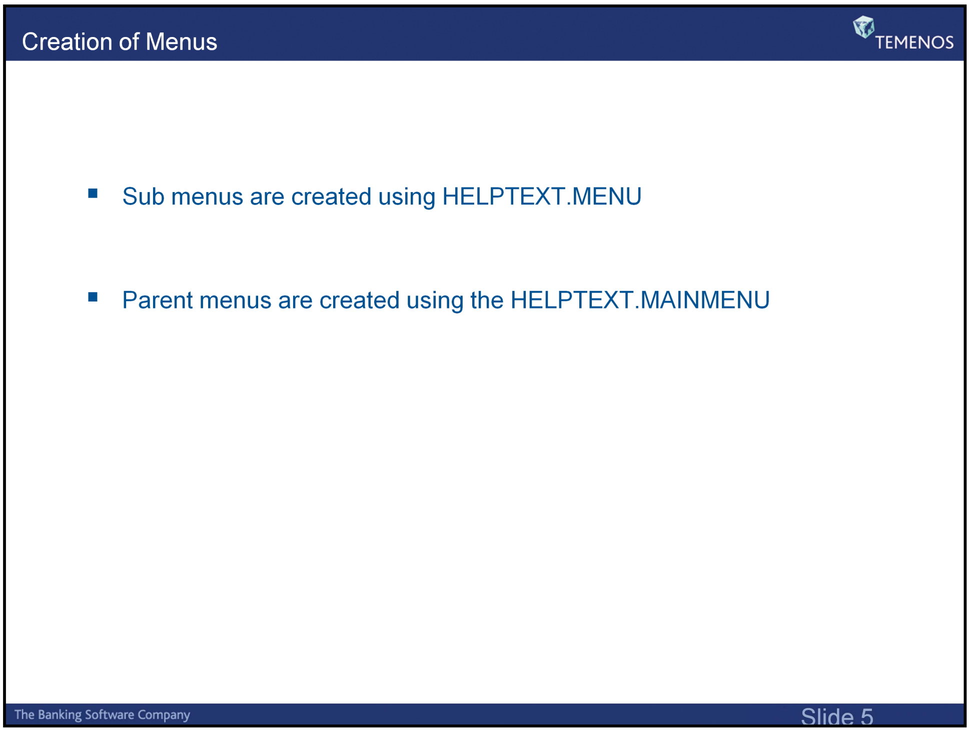
   User Menu, Admin Menu and Role Based Home Pages form the first level of menus.

   Customer, CRM, Contact Management, Product Catalog, Retail Operations, Credit Operations and others form the second level.

   Under Credit operations, Limits, Collateral and Loans form the third level.

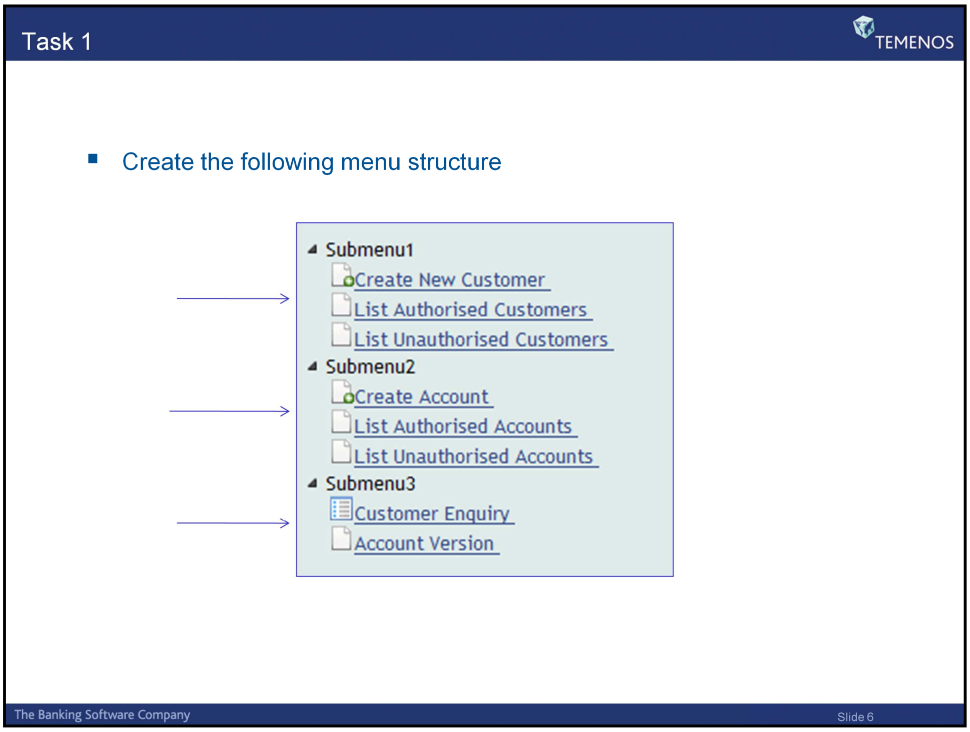
   In Limits, Limit Menu, Group Limits , Enquiries and 'View COB Reports' form the fourth level.

   In this illustration, some T24 applications are listed under Group limits. These are also referred as menu items.



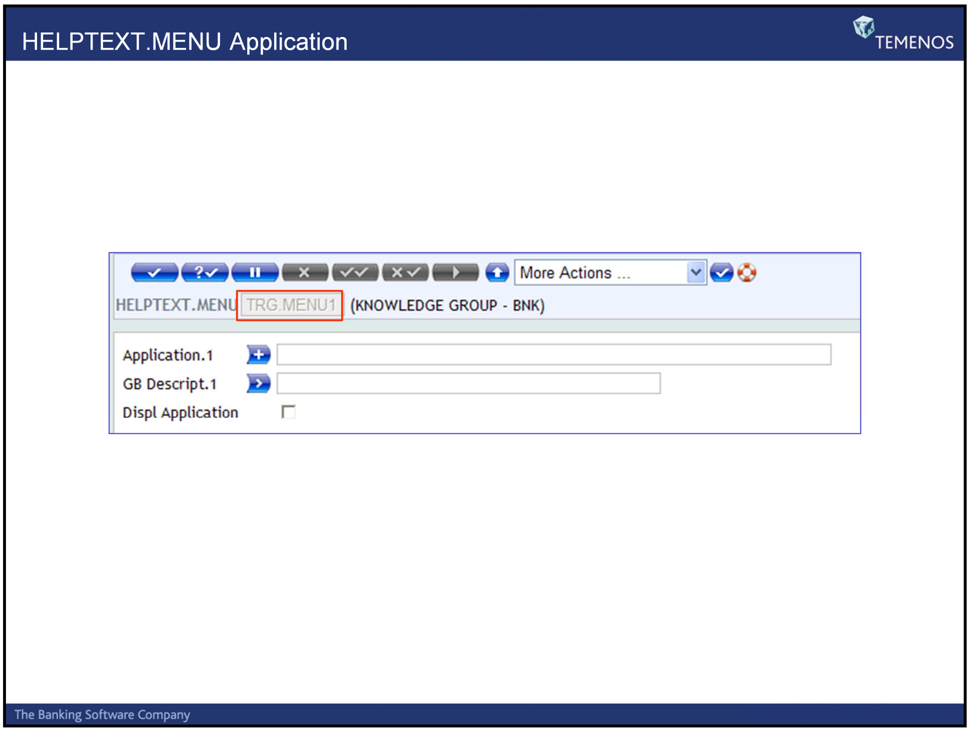
Lets understand how to create these submenus and parent menus.

1. Sub menus are created using **HELPTEXT.MENU** application in T24
2. Parent menus are created using **HELPTEXT.MAINMENU** application and is used to combine the submenus into a main entity.



To learn to create menus in T24, this learning unit demonstrates how you would create a simple menu like the one in the illustration.

The menu that you can see above is one main menu, which has three smaller menus attached as indicated by the arrows. Each of the these submenus have numerous menu items. You have already seen that there are two different applications in T24 that you will use to create menus. You will now learn how to use them.



To create a submenu in T24, you will use the **HELPTEXT.MENU** application. There are only 3 fields in this application.

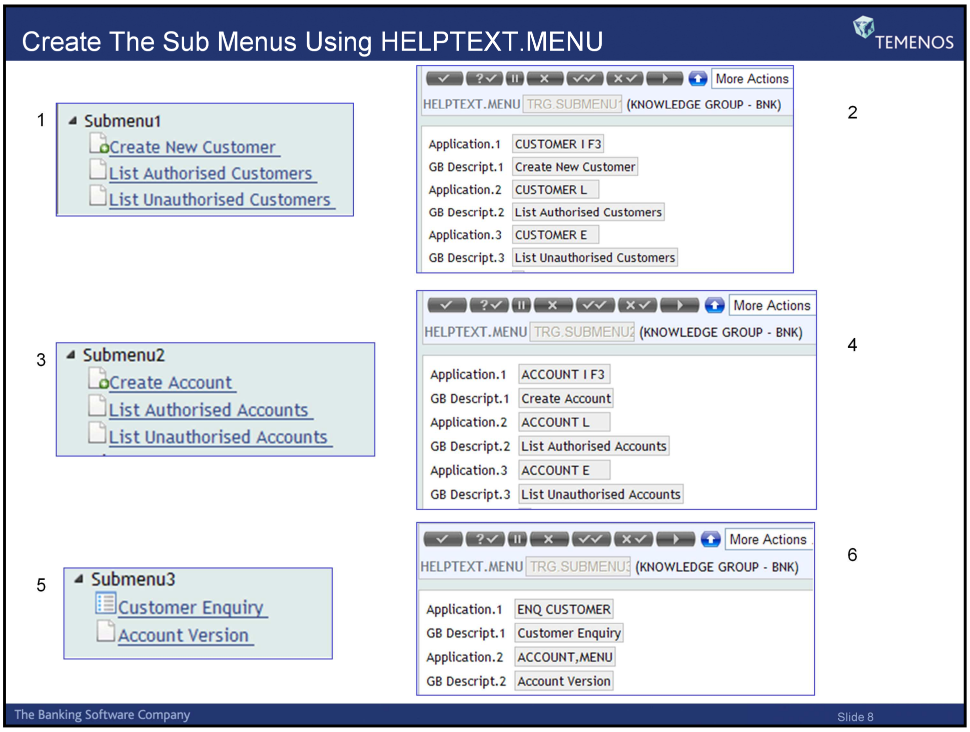
ID: Any user defined text. First character has to be a letter.

APPLICATION : This is where you define the menu items. A menu item can be a valid T24 application name, enquiry name or version name.

The **ACCOUNT** application has been chosen for illustration. You can also add the T24 function behind the application / version name

GB DESCRIPTION : Description of the menu item that will be displayed for the user to click on.

DISPL APPLICATION : This field can be used to display the application name in the menu.



So now that you know how to create a record in **HELPTEXT.MENU**, to accomplish what we started out to create, you will need 3 records in **HELPTEXT.MENU**.

1. Submenu1 has three menu items to create a customer record, list authorized and unauthorized customer records.

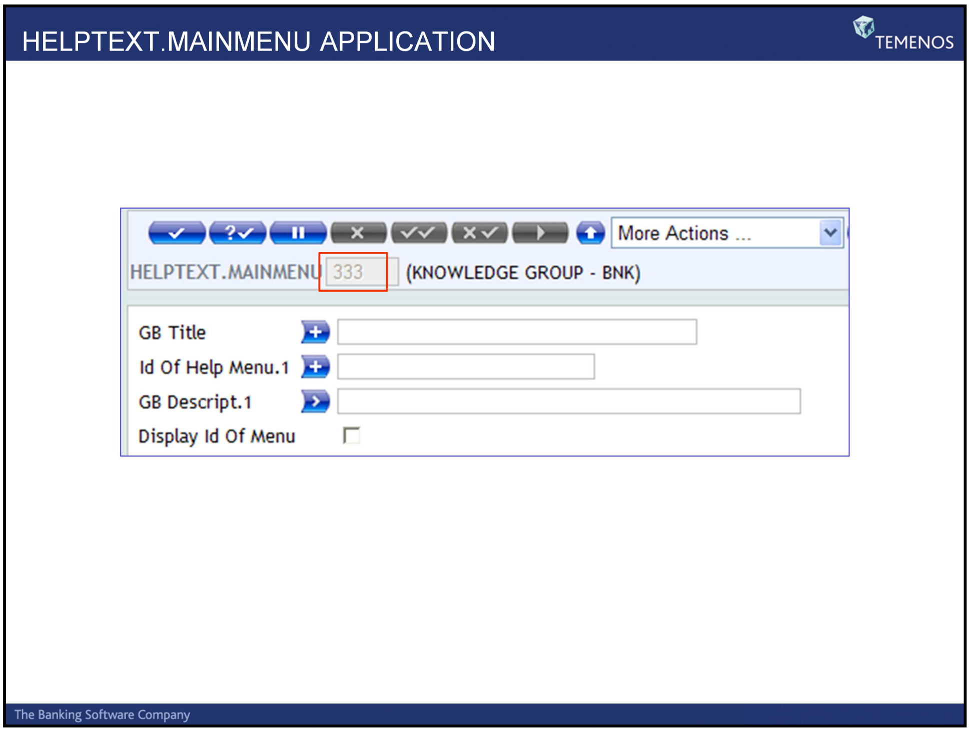
2. The record TRG.SUBMENU1 is created for this purpose.

3. Submenu2 has three menu items to create an account and list unauthorised and authorised accounts.

4. The record TRG.SUBMENU2 is created for this purpose

5. Submenu3 has two menu items to launch an account version and an enquiry to list the customer details.

6. The record TRG.SUBMENU3 is created for this purpose



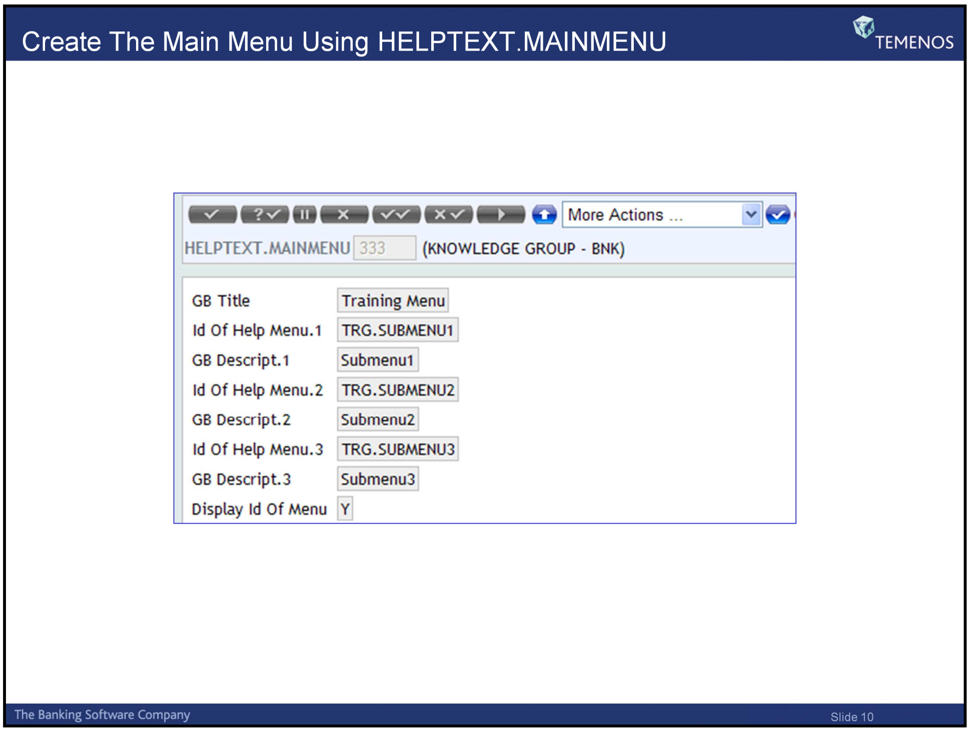
Now that the 3 submenus are ready, how can you use them? All submenus must be attached to create a main menu. This is done using the **HELPTEXT.MAINMENU** application.

ID : Has to be numeric

GB Title : Description for the main menu

Id Of Help Menu : The whole idea of using this application is to bring together existing submenus. This field must hold an the ID of an existing HELPTEXT.MENU

GB Description.3 : Description that will appear for the group of submenus

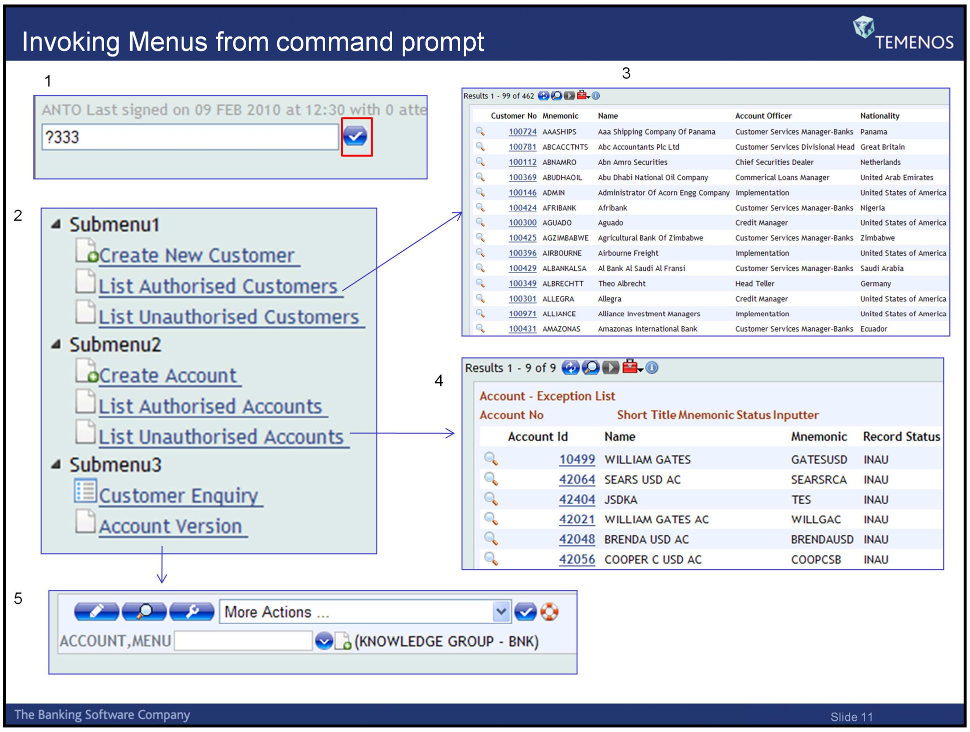


This is the record that must be created in order to group the three sub menus

The ID of the **HELPTEXT.MAINMENU** in this example is 333.

The submenus TRG.SUBMENU1, TRG.SUBMENU2 and TRG.SUBMENU1 are attached to this record.

Do not forget to commit and authorize the record.



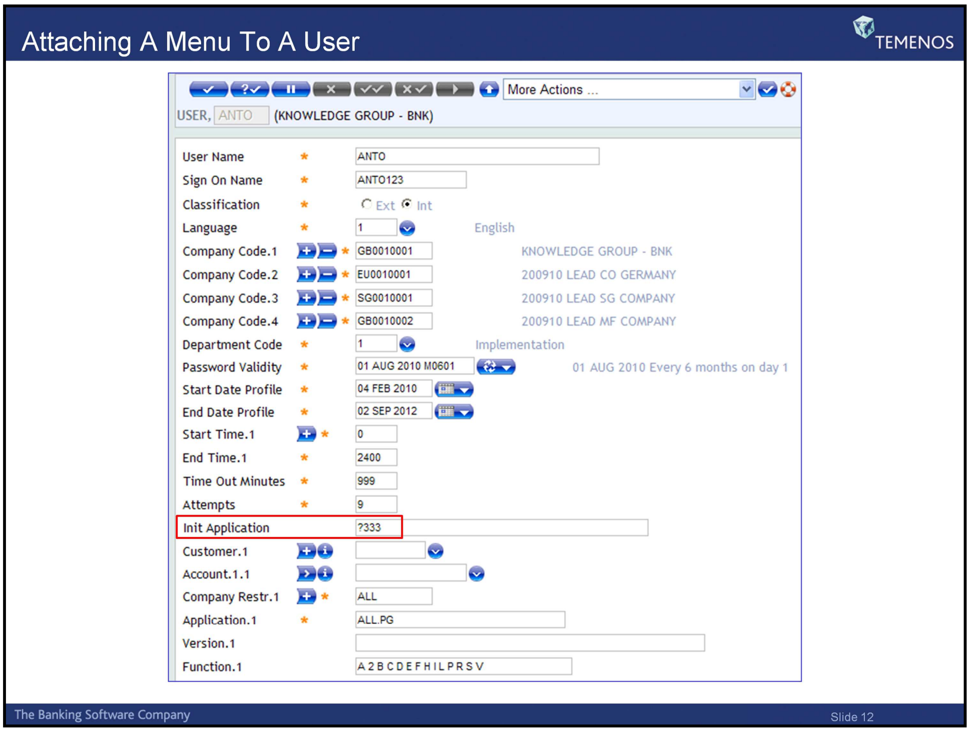
1. Now lets us understand how to invoke the menus that you created. The first way is using the command line. The syntax you must use is a question mark followed by **HELPTEXT.MAINMENU** id (in this example 333)

2. This is how the main menu will look with all the sub menu's expanded.

3. The result of clicking on the menu item to list the authorised customers is shown.

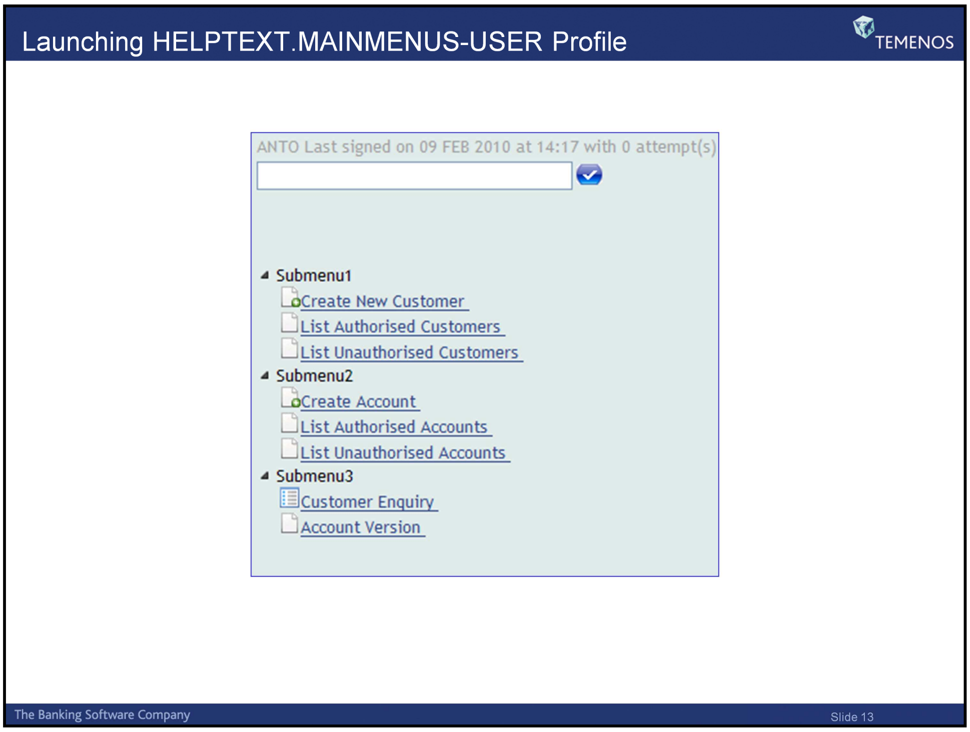
4. This depicts the screen when the menu item to list unauthorised accounts is chosen.

5. This shows the account version being launched from the third sub menu.

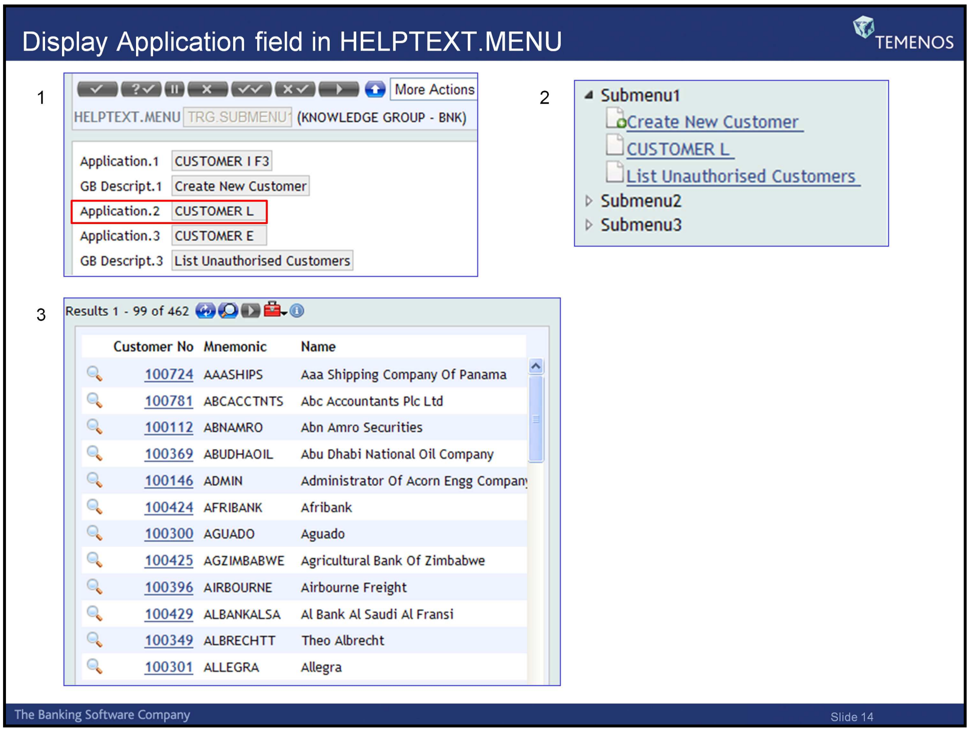


The second way of launching a menu is by attaching it a **USER** profile. By doing this, this menu is displayed as soon as the user logs in.

The **HELPTEXT.MAINMENU** id is attached to the field ‘Init Application’ prefixed with a question mark. In this example, the main menu id is 333 and this is attached to the field as illustrated.



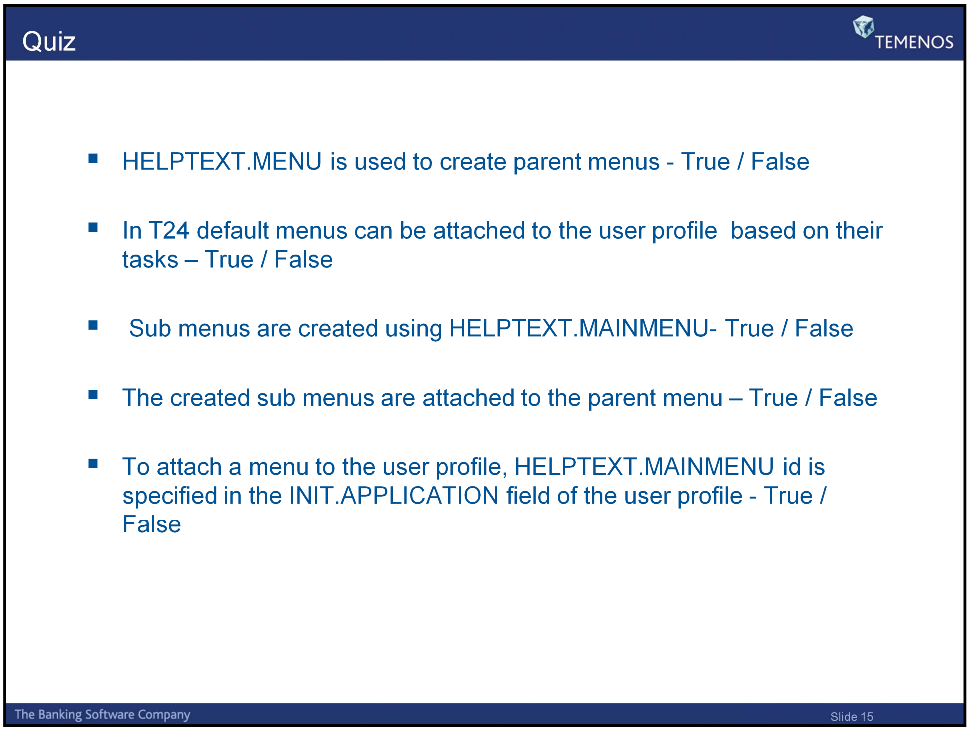
When the user signs on, the menu item attached to the user profile gets loaded for that user.



1. Lets modify the submenu TRG.SUBMENU1 and remove the GB Description for listing the authorised customer records. The 'Display Application' field is set to 'Y'. Don't forget to commit and authorise the record.

2. The application name is displayed when Submenu1 is expanded as illustrated. So it shows CUSTOMER L entered in the field Application.

3. On clicking the link, the enquiry gets executed and lists the authorised records for the customer.



1. HELPTEXT.MENU is used to create parent menus - True / False

   Answer: False

2. In T24 default menus can be attached to the user profile based on their tasks – True / False

   Answer: True

3. Sub menus are created using HELPTEXT.MAINMENU- True / False

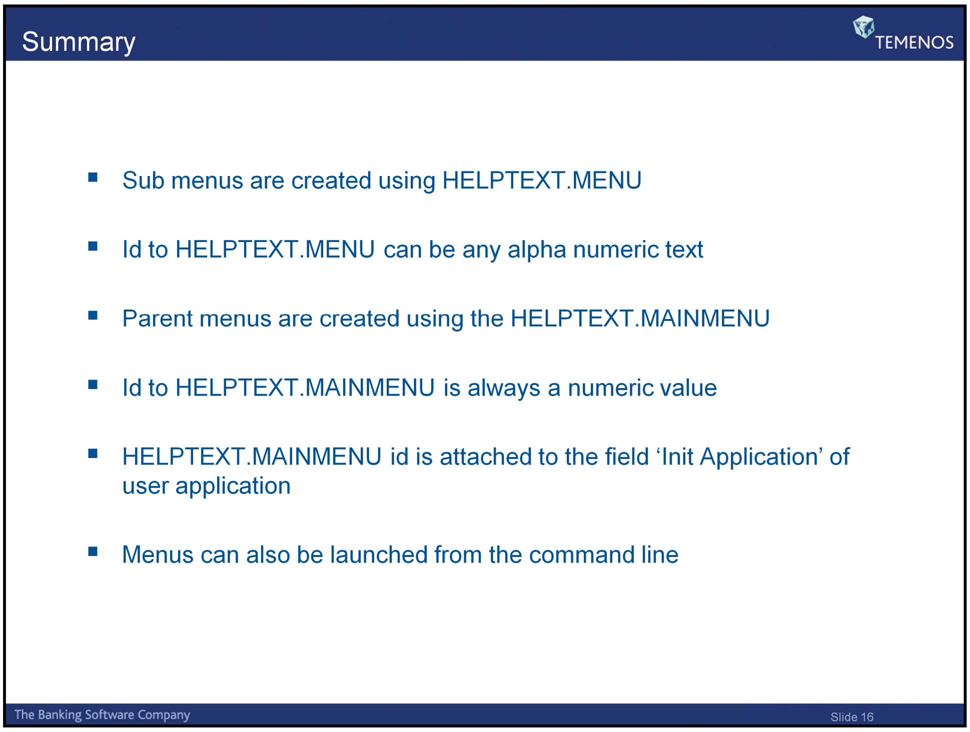
   Answer: False

4. The created sub menus are attached to the parent menu – True / False

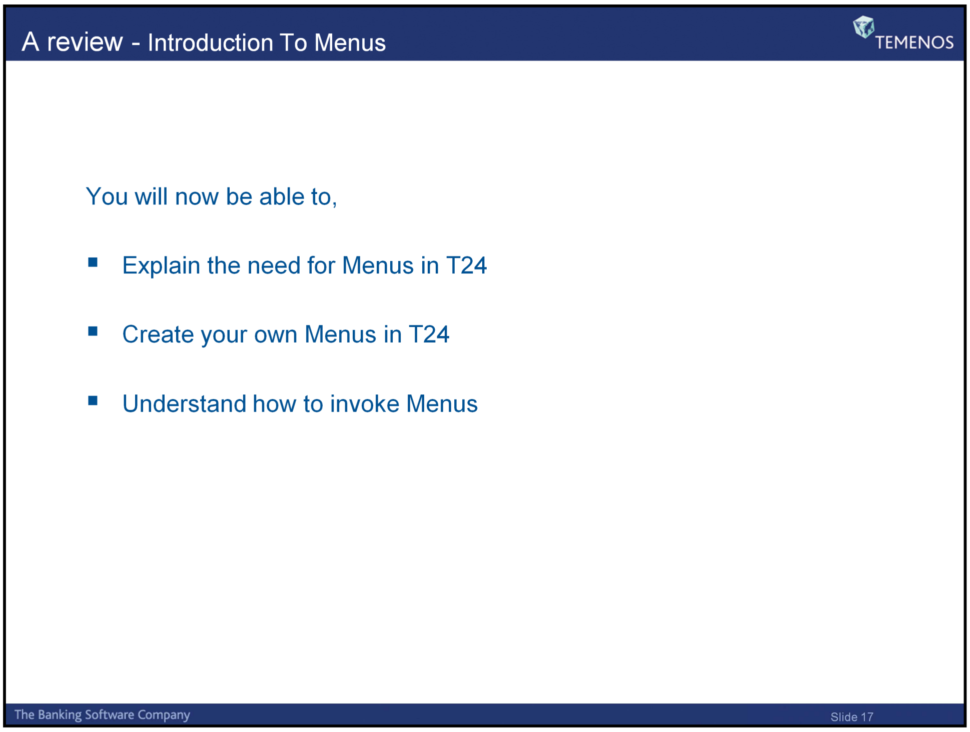
   Answer: True

5. To attach a menu to the user profile, HELPTEXT.MAINMENU id is specified in the INIT.APPLICATION field of the user profile - True / False

   Answer: True



1. Sub menus are created using HELPTEXT.MENU
2. Id to HELPTEXT.MENU can be any alpha numeric text
3. Parent menus are created using the HELPTEXT.MAINMENU
4. Id to HELPTEXT.MAINMENU is always a numeric value
5. HELPTEXT.MAINMENU id is attached to the field ‘Init Application’ of user application
6. Menus can also be launched from the command line



In this learning unit you have learnt about T24 menus. You will now be able to,

1. Explain the need for Menus in T24

2. Create your own Menus in T24

3. Understand how to invoke Menus

TEMENOS

Authoring Contributions:

Alagammai Palaniappan(First Edition, 2004) – Temenos India Private Ltd.
Sara Cleur(Second Edition, 2005) – Temenos India Private Ltd.
Alagammai Palaniappan(Third Edition, 2006) – Temenos India Private Ltd.
B. Nanda Kumar(Fourth Edition, 2007) – Temenos India Private Ltd.
Nasim Kazi S(FifthEdition, Dec 2007) – Temenos India Private Ltd.
Sudarkodi S(FifthEdition, 2008) – Temenos India Private Ltd.
Anto William (Sixth Edition, 2010) – Temenos India Private Ltd.

Thankful Acknowledgements:

Temenos Corporate Training Team

The Banking Software Company

Slide 18

Thank You

TEMENOS EDUCATION CENTRE

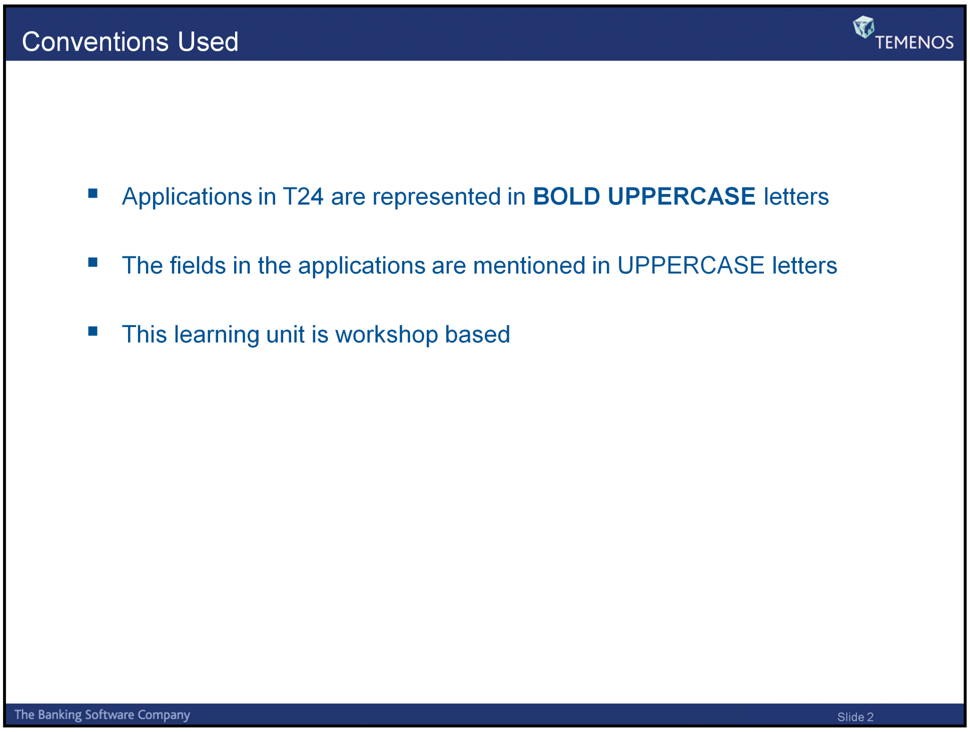
TEMENOS
The Banking Software Company

Manipulating Field Properties in Versions

TEMENOS EDUCATION CENTRE

TEMENOS
The Banking Software Company

Welcome to the learning unit on 'Manipulating Field Properties in Versions' . In this course you will learn the enhanced features that you could add to the fields of a version.

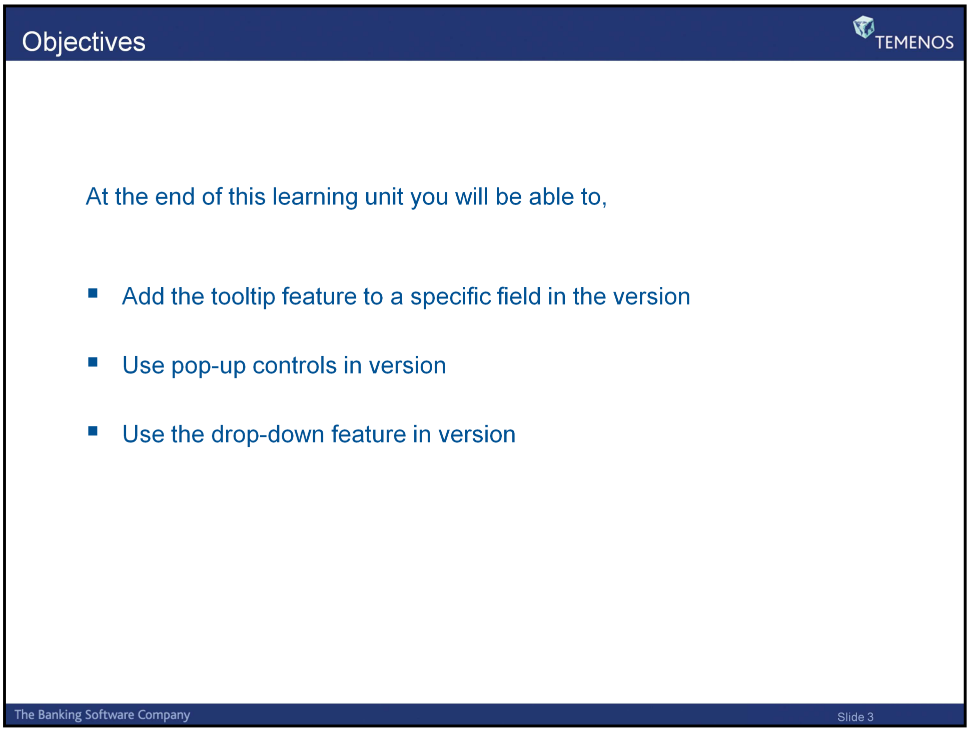


The conventions used in this learning unit are,

1.Applications in T24 are represented in bold uppercase letters.

2.The fields in the applications are mentioned in uppercase letters.

3. This learning unit is example based. First an example output will be shown then you will be taught the steps to create the Version. At the end of each section a workshop is given to test your level of understanding.



At the end of this learning unit you will be able to,

1. Add the tooltip feature to a specific field in the version
2. Use pop-up controls in version
3. Use the drop-down feature in version

Task 1

- Amend the version CUSTOMER,DETAIL . Add the following fields - INTRODUCER and CONTACT.DATE in the version record
- Make the following changes in the version record,
  - The field INTRODUCER should be made as a drop down field. In the dropdown menu specify the valid customer ids from the **CUSTOMER** application
  - Add the tooltip to this field as ‘Introducer’
  - Display a line break after the field NAME.1
  - To the field CONTACT.DATE add a pop-up to display the calendar

The Banking Software Company Slide 4

The given task teaches you to add extra features like tooltip and pop-up controls to the fields in the version, and you will also learn to introduce a line break between the fields. The requirement is as follows,

1. Amend the version CUSTOMER,DETAIL . Add the following fields - INTRODUCER and CONTACT.DATE in the version

In the field INTRODUCER of the version you need a valid customer id to be specified. Can you attach a menu to this field to display the available customer ids? Can an ordinary field be amended as a dropdown field in the version?

Yes you can attach a dropdown menu to the normal fields of your version.

2. Make the following changes in the version record,

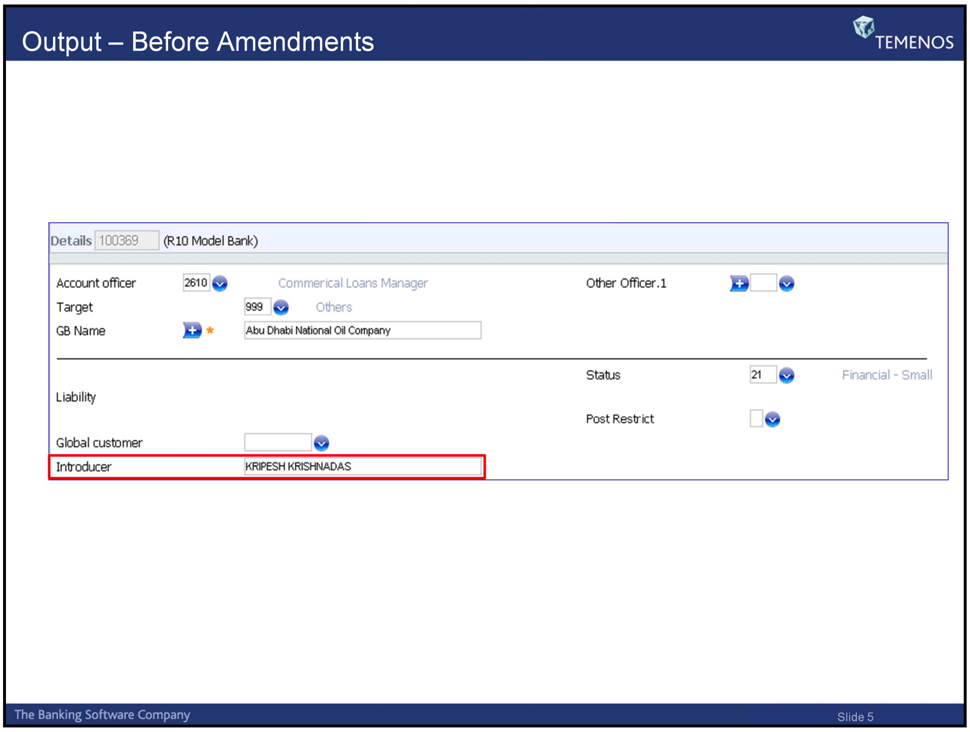
    2.1 The field INTRODUCER should be made as a drop down field. In the dropdown menu specify the valid customer ids from the **CUSTOMER** application

    2.2 Add the tooltip to this field as ‘Introducer’

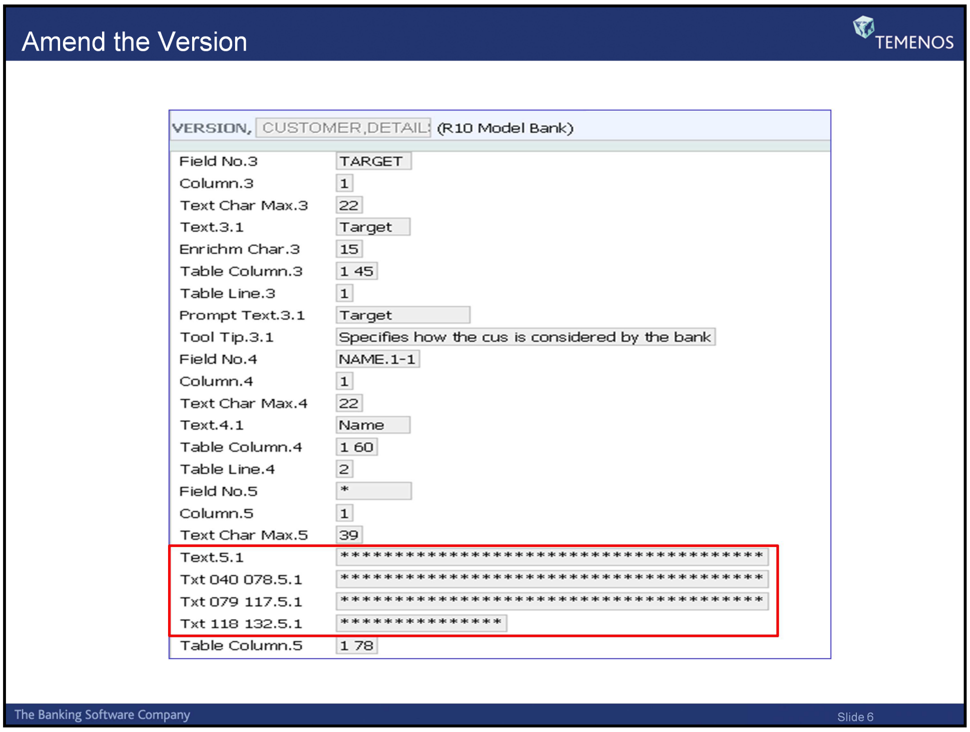
The user’s job could be made simpler using the various features in version . To the field CONTACT.DATE if a calendar is attached as a pop-up menu it will be easy to choose the date.

    2.3 Display a line break after the field NAME.1

    2.4 Add a pop-up to display the calendar to the field CONTACT.DATE



You can see the output of the version before all the amendments were made. Note that the field INTRODUCER is an ordinary field in the version.



Now you will learn about the other customization features that you can add to the fields of your version.

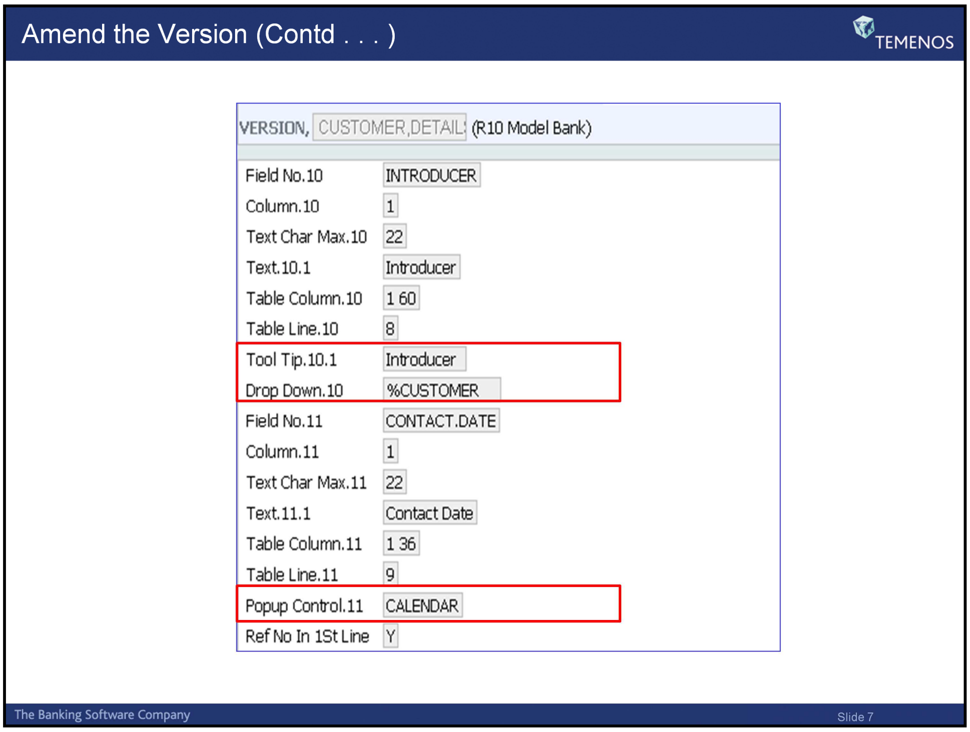
FIELD.NO – In the multi-value field of FIELD.NO.5 a line break is introduced by specifying an asterisk in this field.

TEXT Char – You may enter a field label here. The maximum length of the characters specified is 39 unless you specify an asterisk in the FIELD.NO field. This is a sub-value field. In order to view the line break you specify a maximum of 39 asterisks in this field. The only permitted character here is asterisk and it is not a mandatory field.

TXT 040 078- This field is only used when a line break ('*' in FIELD NO) across the screen is required.

This field is part of the sub-value which holds value from column 40 to 78. Here you specify the asterisks to get the line break. Similarly for column 79 to 117 the value is specified in the next following field. The following field holds the values for column position from 118 to 132.

When you execute the version a line break will be displayed after the field NAME.1



The field INTRODUCER should display the tooltip as ‘Introducer’ .

TOOL.TIP – Specify the tooltip as Introducer in the field TOOL.TIP.

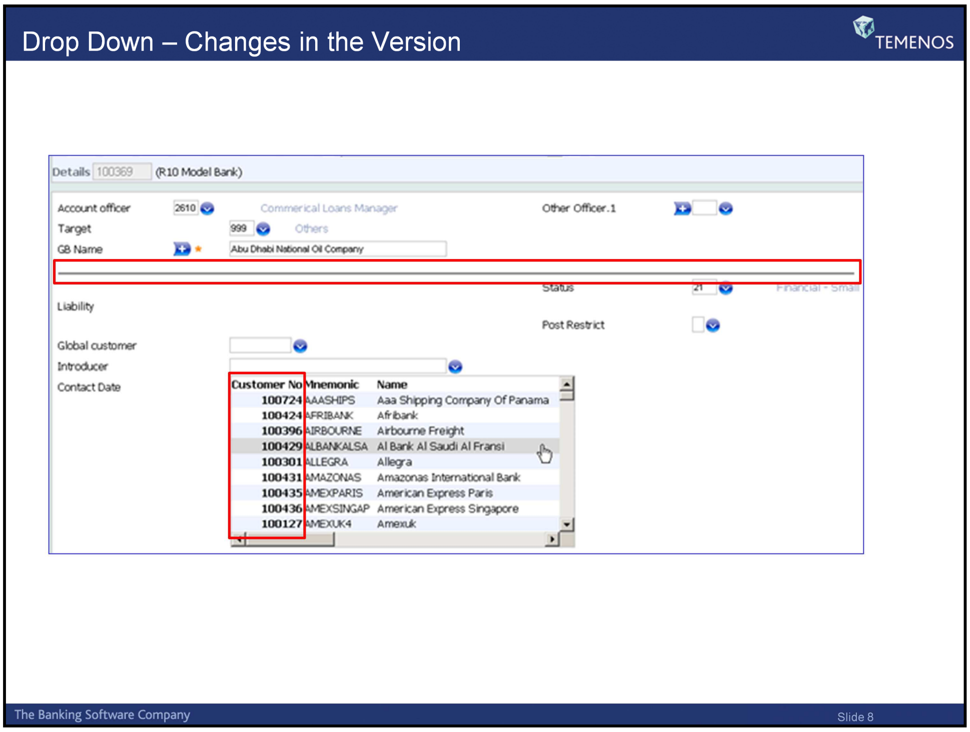
Now add the dropdown feature to the field INTRODUCER. In order to get the customer ids in the menu, you will make use of the default enquiry (%CUSTOMER) for the **CUSTOMER** application in T24.

DROP.DOWN – In the DROP.DOWN field enter a valid enquiry id. This enquiry will be launched to fetch the customer ids. You can also choose the default customer enquiry from the dropdown menu available.

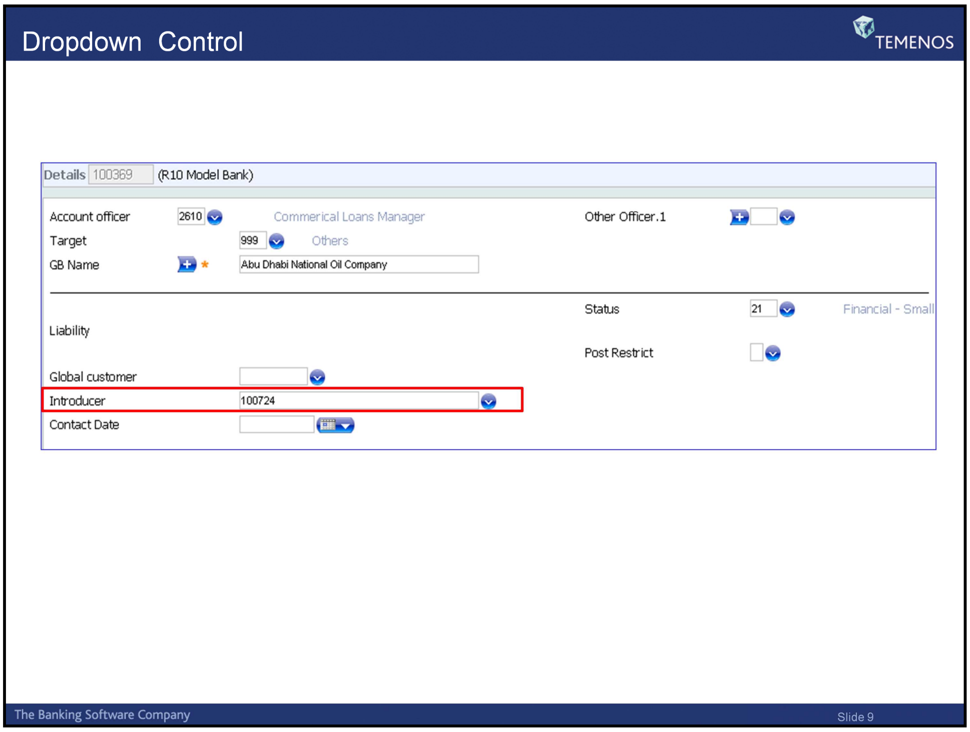
In T24 four different popup controls are available namely calculator, calendar, rate control & recurrence. The last feature to be added to your version is the popup control for the field CONTACT.DATE .

POPUP.CONTROL – Choose the calendar option from the dropdown menu to enable calendar as a popup display for the field CONTACT.DATE

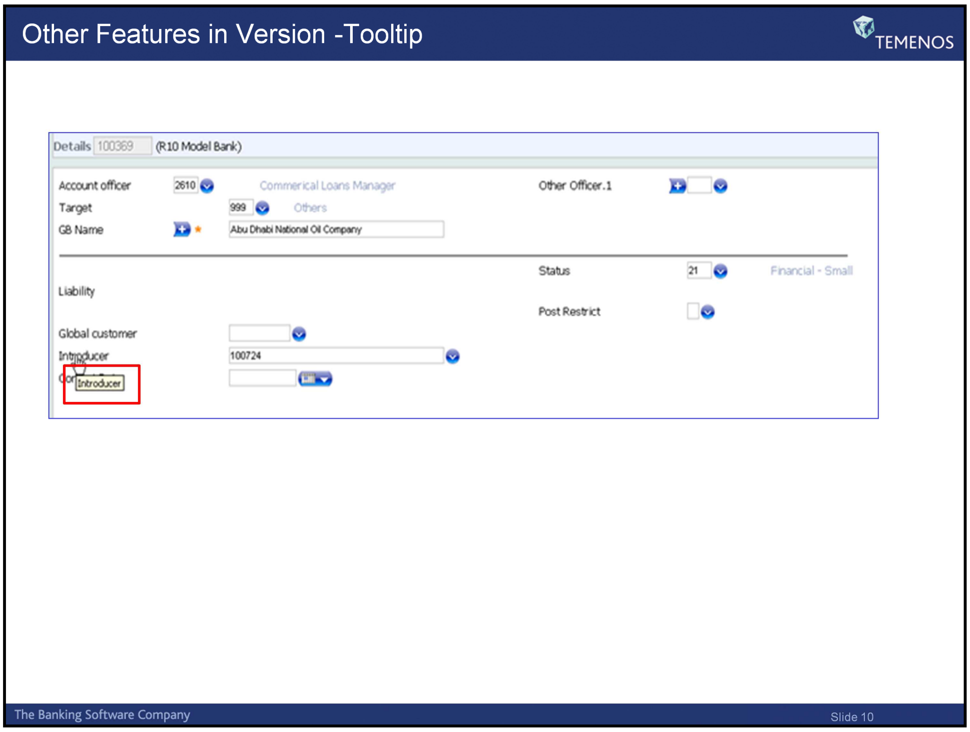
Note : Only the calendar option works in the browser



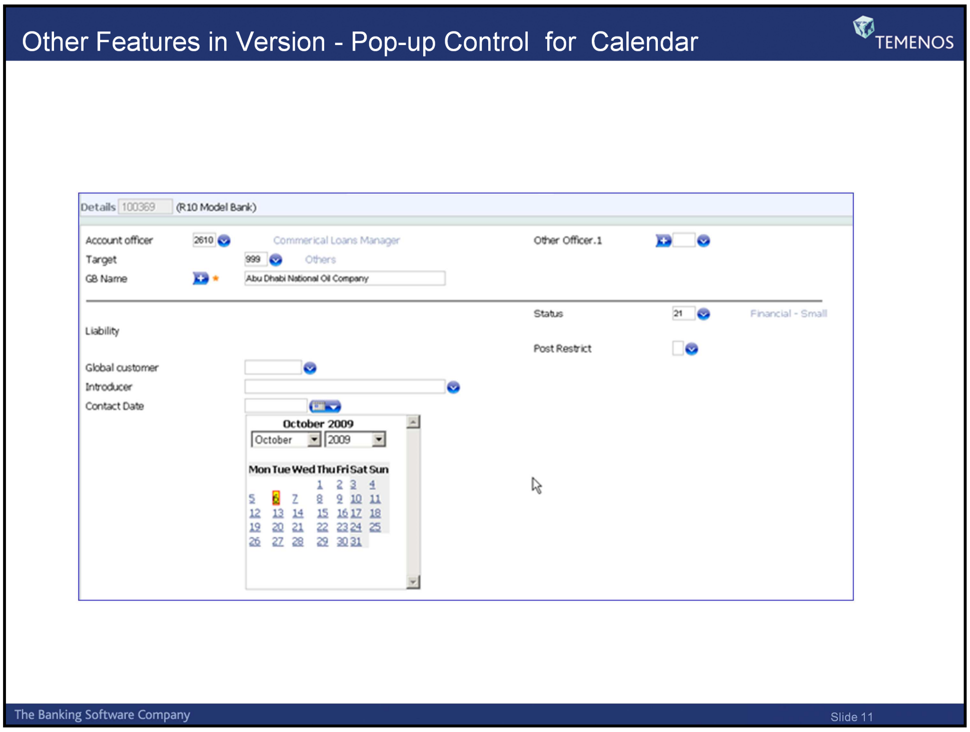
Launch the version. You can see the line break which is displayed after the field NAME.1. The field INTRODUCER has the dropdown feature added to it. Once you click on the dropdown menu of the INTRODUCER field the default customer enquiry will be launched. Now you can select a customer record from the dropdown .



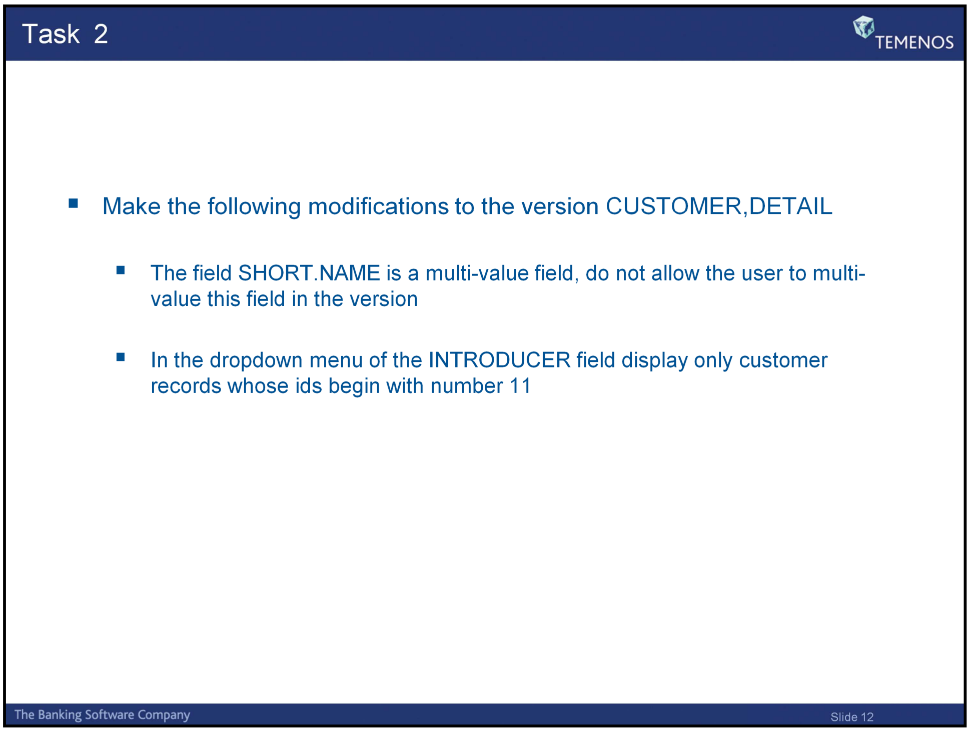
Note: Only the customer id is displayed in the field INTRODUCER. Always the customer id will be returned when you choose a customer record from the dropdown menu.



You can view the tooltip that is attached to the INTRODUCER field of the version.



The field CONTACT.DATE provides an additional feature for displaying the calendar as the popup control.



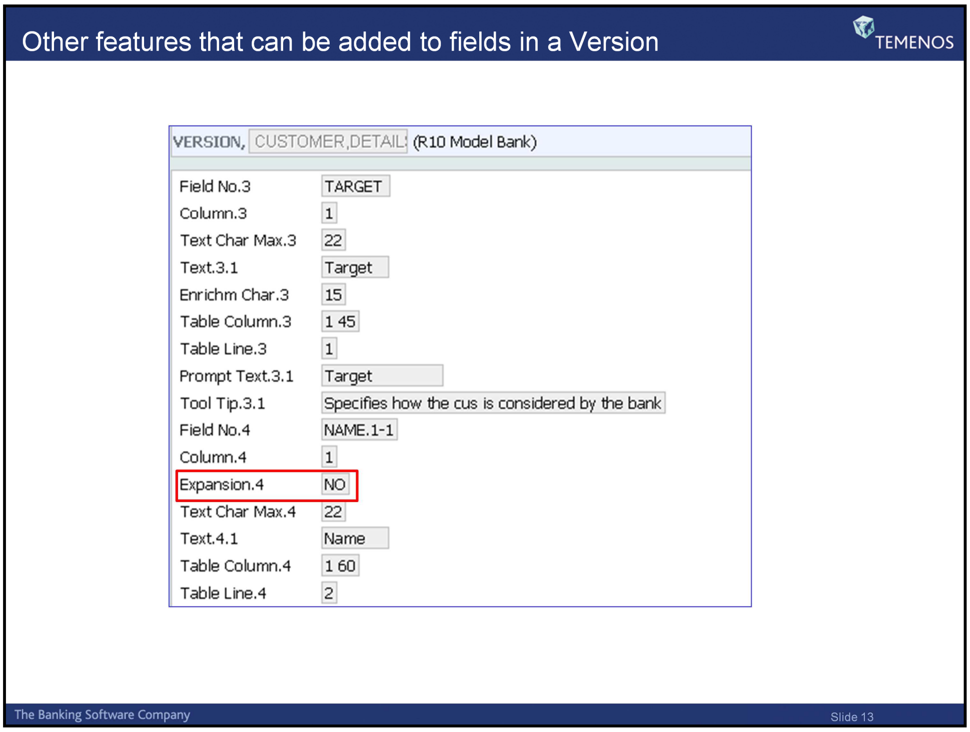
The stated task teaches you to attach a drop-down menu to a specific field in the version. The requirement for the given task is as follows,

When a user creates customer records you may want the user to specify the introducer as an existing customer. To do so you may add a dropdown feature to the introducer field.

1. Make the following modifications to the version CUSTOMER,DETAIL

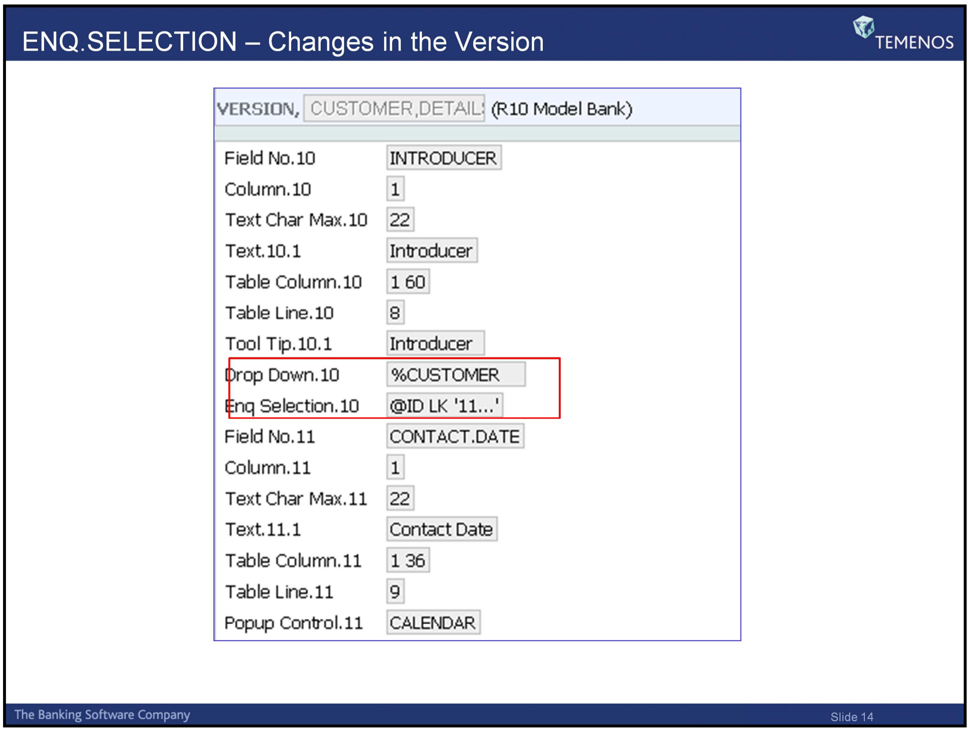
1.1 The field SHORT.NAME is a multi-value field, do not allow the user to multi-value this field in the version

1.2 In the dropdown menu of the INTRODUCER field display only customer records whose ids begin with number 11.



The field SHORT.NAME is a multi-value field in the **CUSTOMER** application. The user should not be allowed to expand this field in the version.

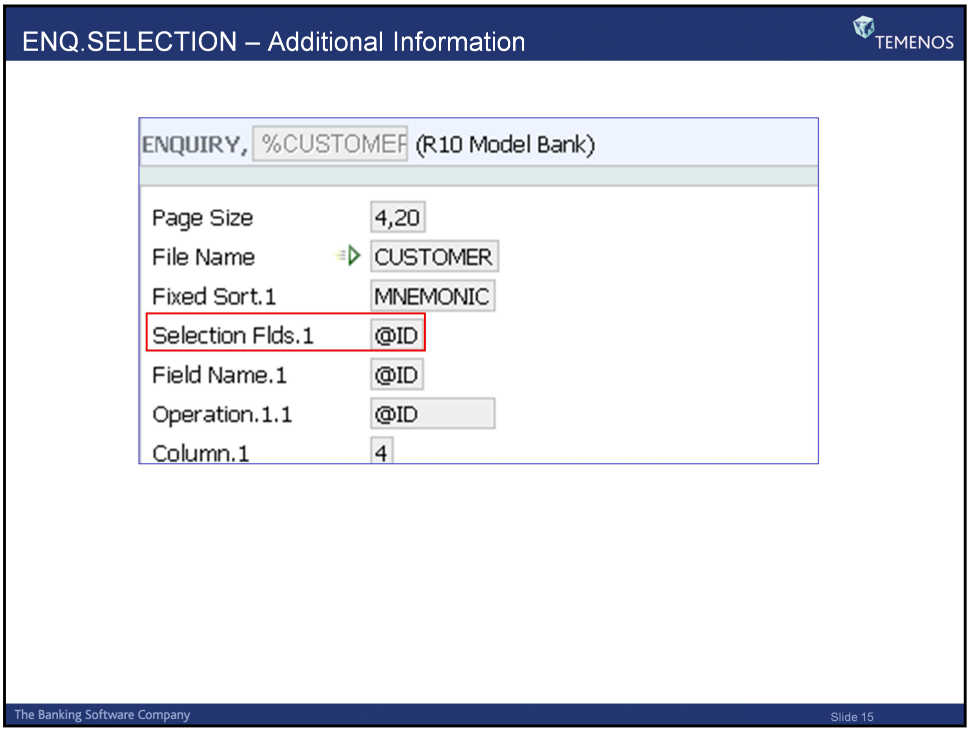
EXPANSION – In the version the EXPANSION field indicates whether a field can be expanded or not . Input to this field is used only when FIELDS.PER.LINE is set multi. By default this field is left blank, the multi-value or sub-value field can then be expanded on-line when using this version . Therefore in your version record, for the field SHORT.NAME set the field EXPANSION to 'NO' . This makes the multi-value icon to disappear in the version.



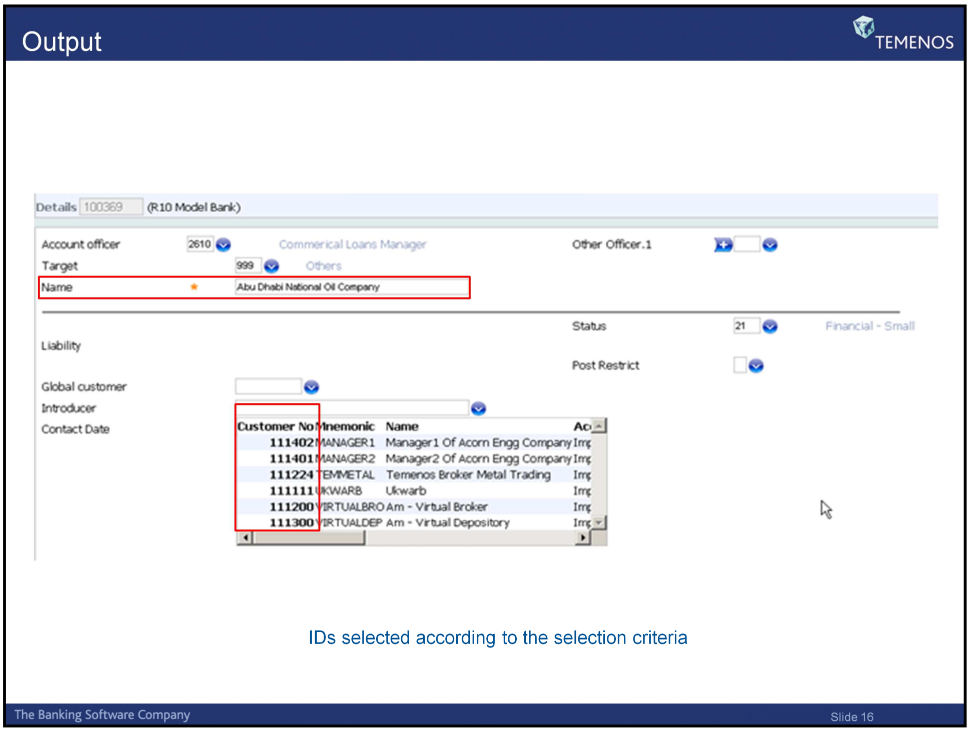
User can enter a selection criteria that must be used by the ENQUIRY used in the Drop Down Field.

Now for the field INTRODUCER , the dropdown menu should display only those set of customer records that match the selection criteria. Here the selection criteria is to fetch only those customer records whose id's start with the number 11.

ENQ.SELECTION – The selection criteria is specified in the field ENQ.SELECTION as @ID LK "11…" Where LK is the operand for LIKE. The field ENQ.SELECTION works with the DROP.DOWN Field. The given condition will fetch only those customer records whose id's start with 11.



Note : @ID is a valid selection field defined in the default ENQUIRY %CUSTOMER



Launch the version. The field NAME.1 cannot be multi-valued in this version. In the dropdown option for the field INTRODUCER you can see that the customer ids matches the selection criteria.

Quiz

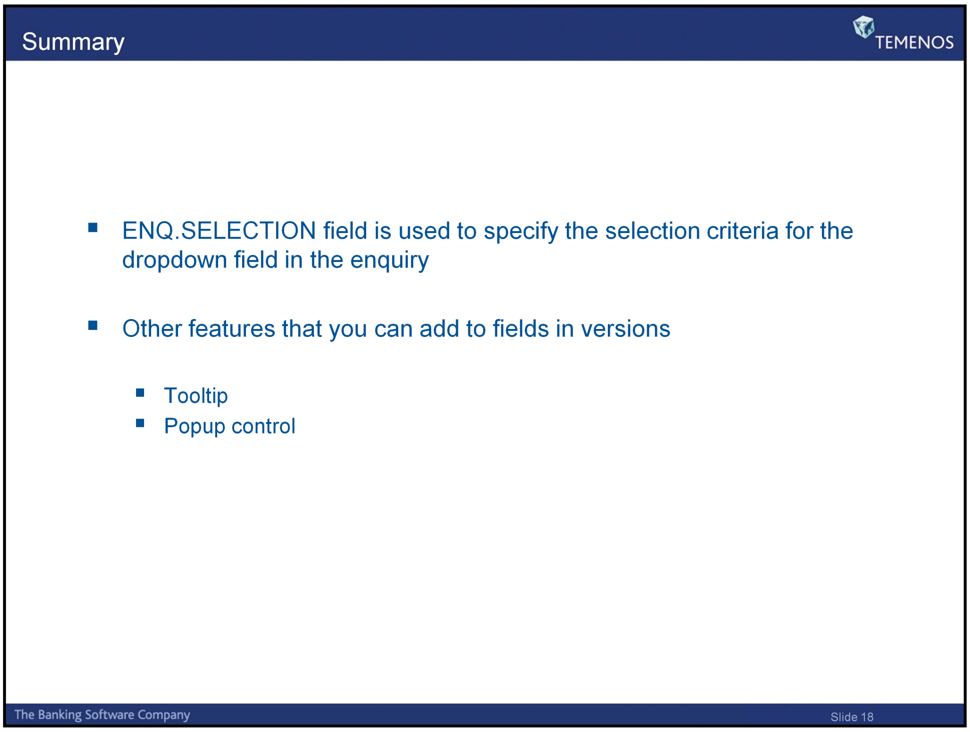
TEMENOS

- Drop-down menus can be attached to fields in version - True / False
- Tool tip can be attached to fields in versions using the field TOOL.TIP - True / False
- The EXPANSION field is used only when FIELDS.PER.LINE is set to multi - True / False
- To specify selection criteria for drop down field , the specific field need not be a valid selection field of the enquiry - True / False

The Banking Software Company

Slide 17

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

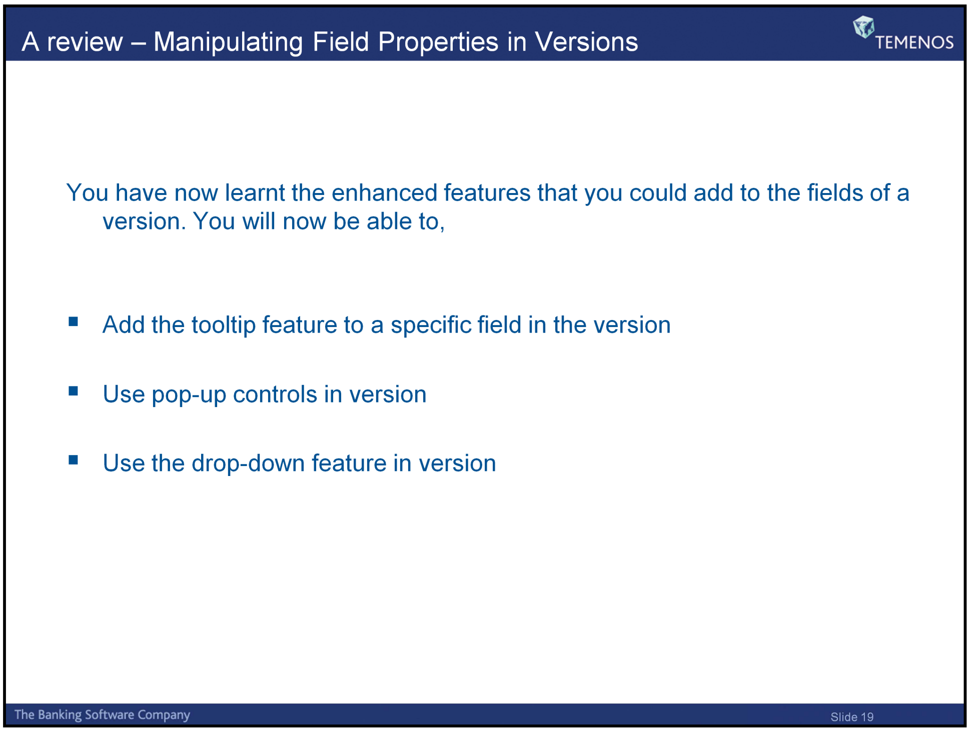


1. ENQ.SELECTION field is used to specify the selection criteria for the dropdown field in the enquiry

2. Other features that you can add to fields in versions

   2.1 Tooltip

   2.2 Popup control



You have now learnt the enhanced features that you could add to the fields of a version. You will now be able to,

1. Add the tooltip feature to a specific field in the version
2. Use pop-up controls in version
3. Use the drop-down feature in version

Version History

TEMENOS

Version	Date	Author	Reviewer
1.0			
1.1	13th Feb 2008	Nasim Kazi S	
1.2		Assis Priya	Sara Cleur
1.3	15th Dec 2009	Kripesh Krishnadas	Kripesh Krishnadas

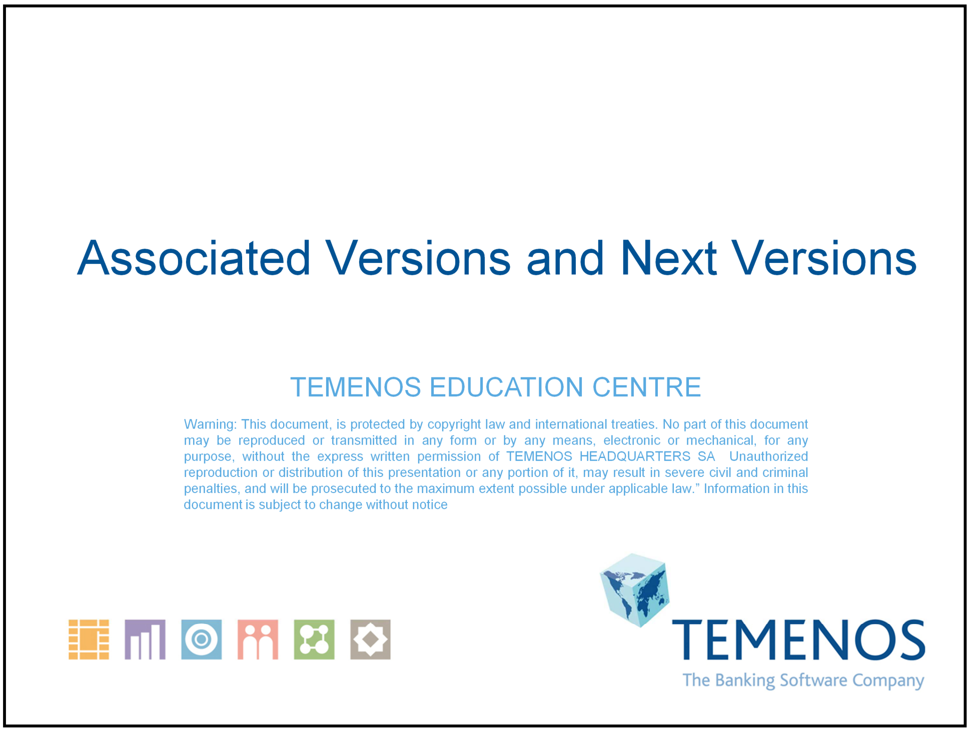
The Banking Software Company

Slide 20

Thank You

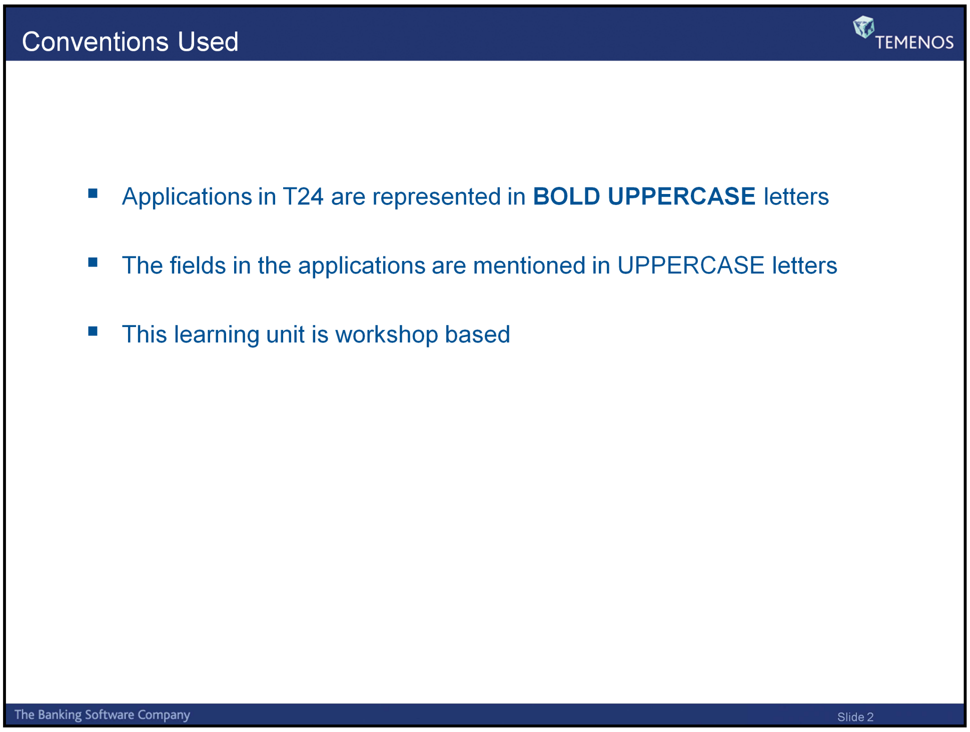
TEMENOS EDUCATION CENTRE

TEMENOS
The Banking Software Company



CUS24-Associated Versions and Next Versions-R09.01

Welcome to the learning unit on 'Associated Versions and Next Versions' . In this course you will learn associated versions and next versions in T24.



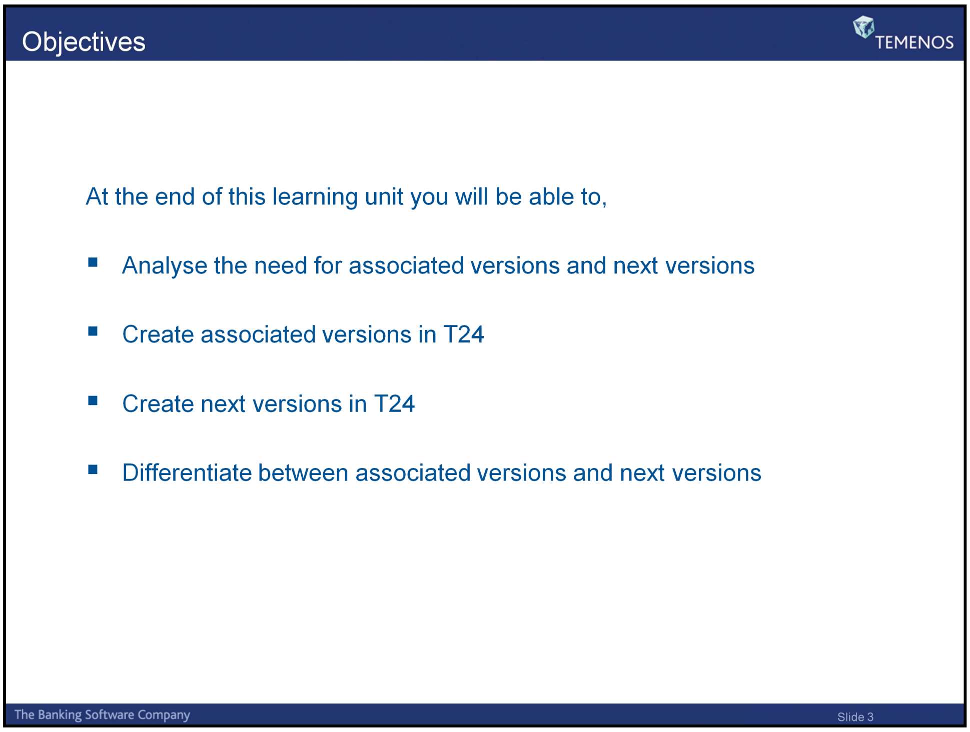
CUS24-Associated Versions and Next Versions-R09.01

The conventions used in this learning unit are,

1.Applications in T24 are represented in bold uppercase letters.

2.The fields in the applications are mentioned in uppercase letters.

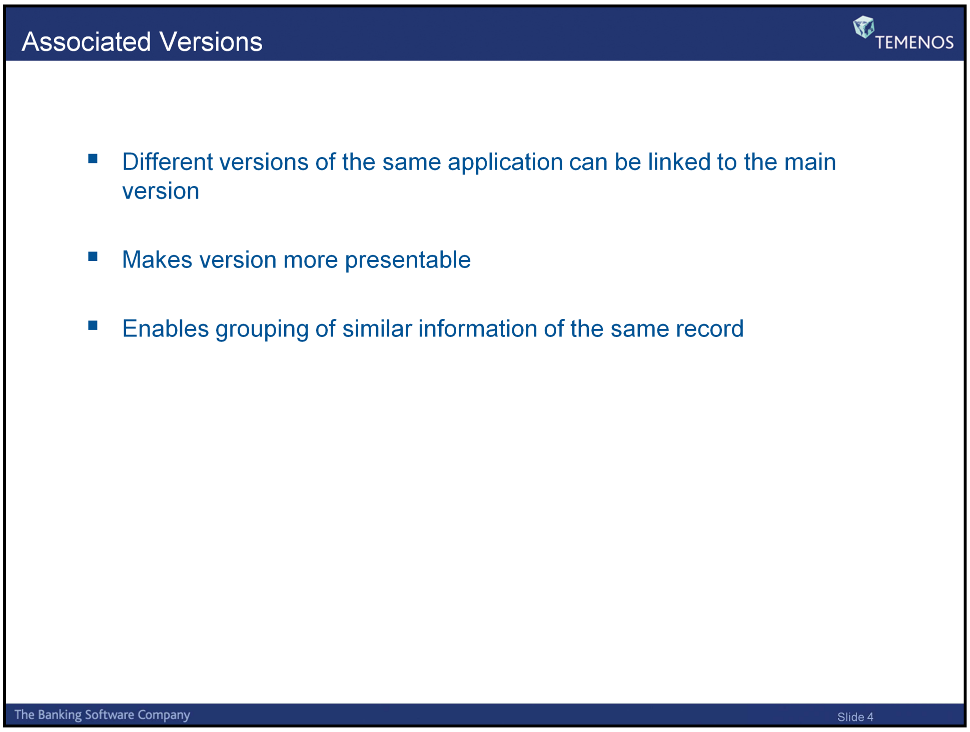
3. This learning unit is example based. First an example output will be shown then you will be taught the steps to create the Version. At the end of each section a workshop is given to test your level of understanding.



CUS24-Associated Versions and Next Versions-R09.01

At the end of this learning unit you will be able to,

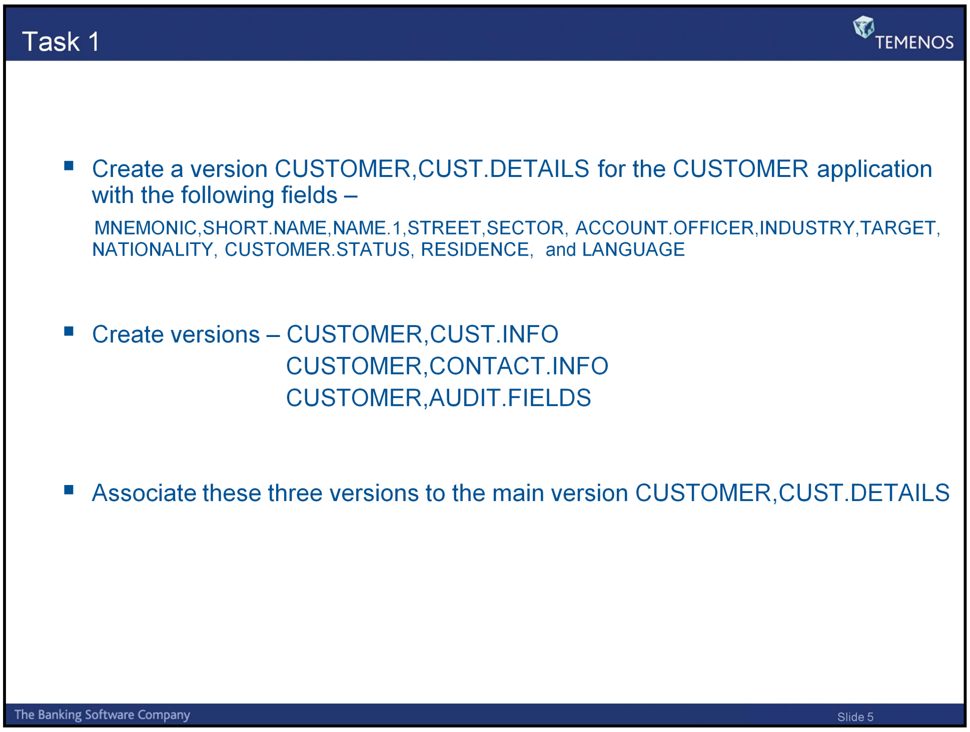
1. Analyse the need for associated versions and next versions
2. Create associated versions in T24
3. Create next versions in T24
4. Differentiate between associated versions and next versions



CUS24-Associated Versions and Next Versions-R09.01

You know to create versions in T24. Imagine that you have created two versions for the **CUSTOMER** application . One to view the customer info and the other to view the contact details of the customer. Is there a way to link these two versions in T24 ?

Yes, Using associated versions you can link versions that are based on the same application, here the versions are based on the **CUSTOMER** application. These versions are attached to the main version. Using associated versions you can group similar information of the data required.



The stated task teaches you to create associated versions in T24. The requirement is stated here as follows,

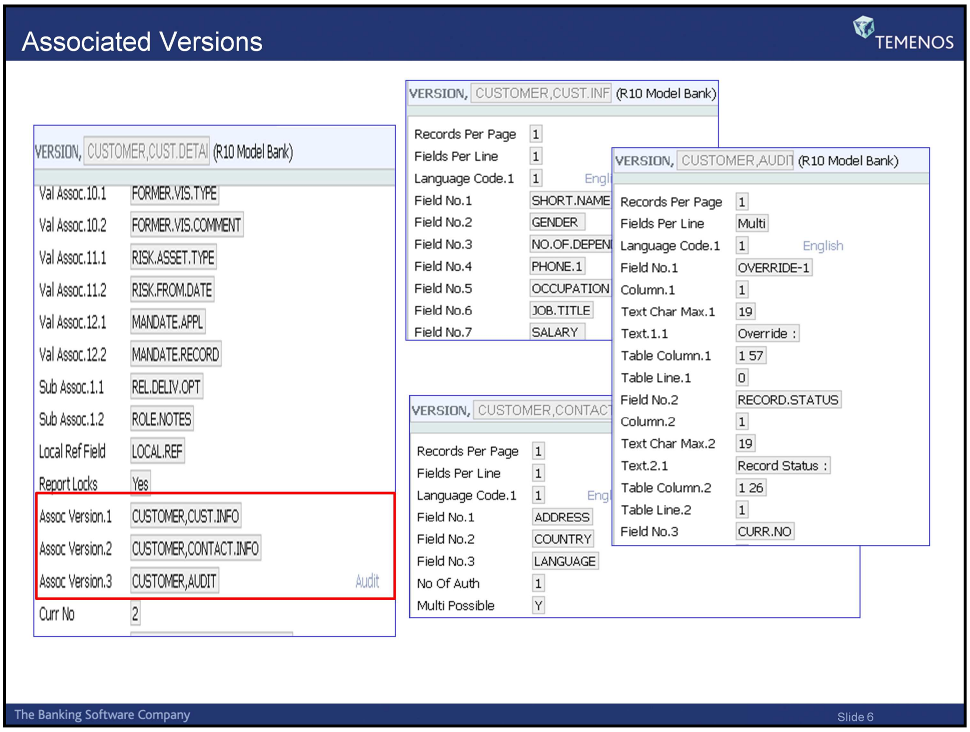
1. Create a version CUSTOMER,CUST.DETAILS for the CUSTOMER application with the following fields –

MNEMONIC,SHORT.NAME,NAME.1,STREET,SECTOR,ACCOUNT.OFFICER,INDUSTRY,TARGET,NATIONALITY, CUSTOMER.STATUS, RESIDENCE and LANGUAGE

2. Create versions - CUSTOMER,CUST.INFO / CUSTOMER,CONTACT.INFO / CUSTOMER,AUDIT.FIELDS

The version CUSTOMER,CUST.INFO will display the general customer information and CUSTOMER,CONTACT.INFO displays contact information of the customer and the version CUSTOMER,AUDIT.FIELDS displays the audit fields of the customer record.

3. Associate the 3 versions to the main version CUSTOMER,CUST.DETAILS



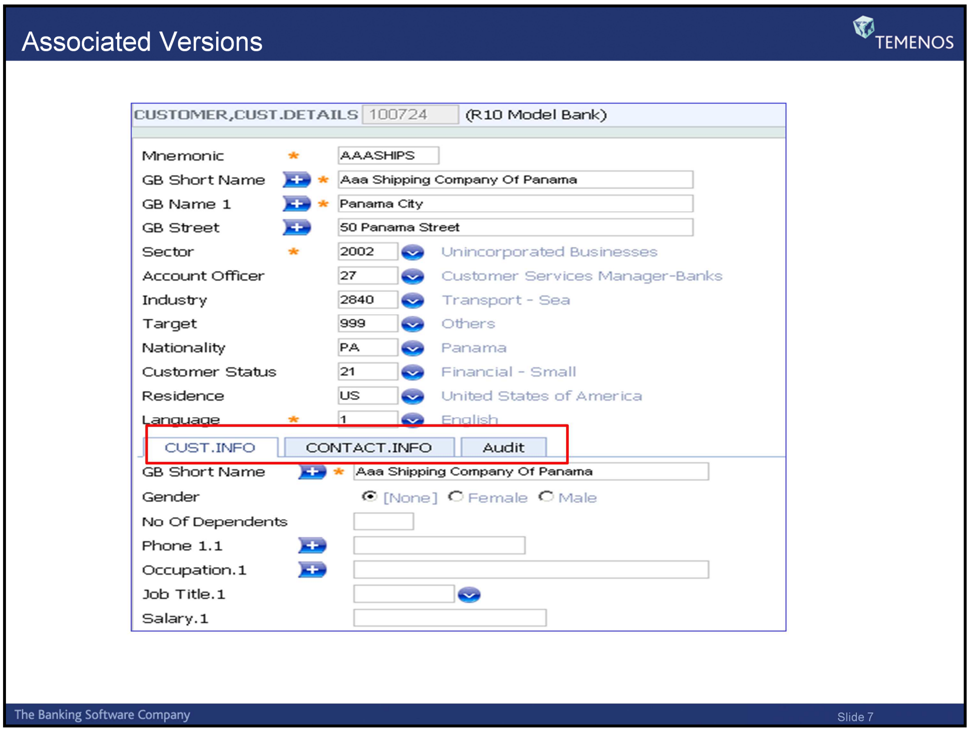
CUS24-Associated Versions and Next Versions-R09.01

Create the main version and also the other 3 versions that are to be associated.

1. The versions that are to be linked should be created before you attach it to the main version

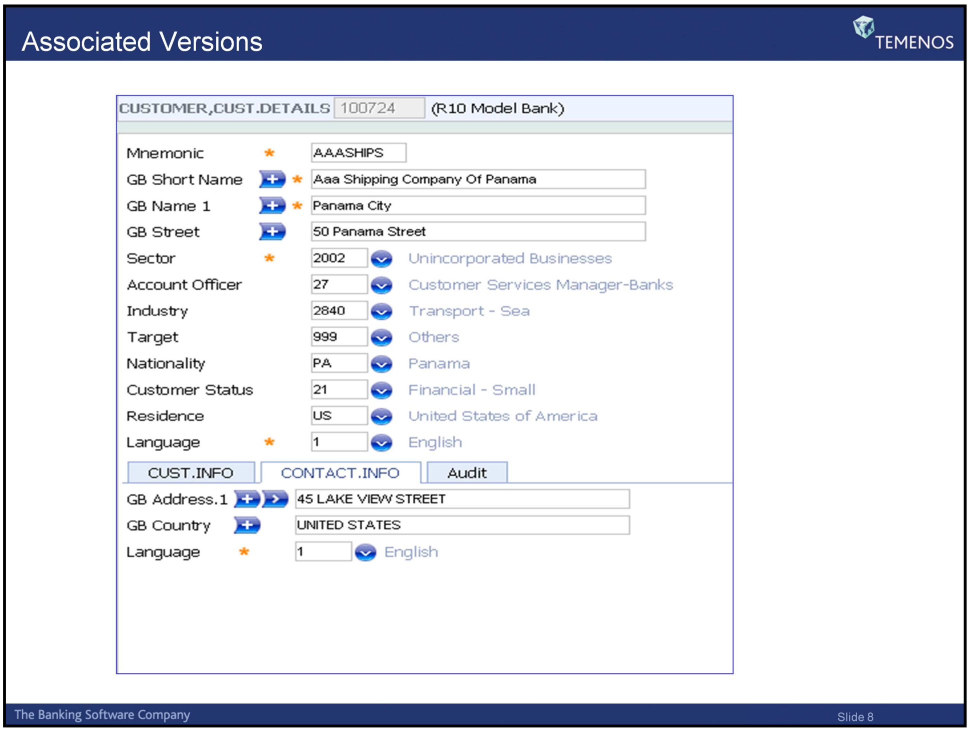
2. The associated versions are attached to the main version by specifying the version name in the field ASSOC.VERSION of the main version. You have to specify a valid version name here. It is a multi-value field.

3. The associated versions appear as tabs in the main version they are attached to.



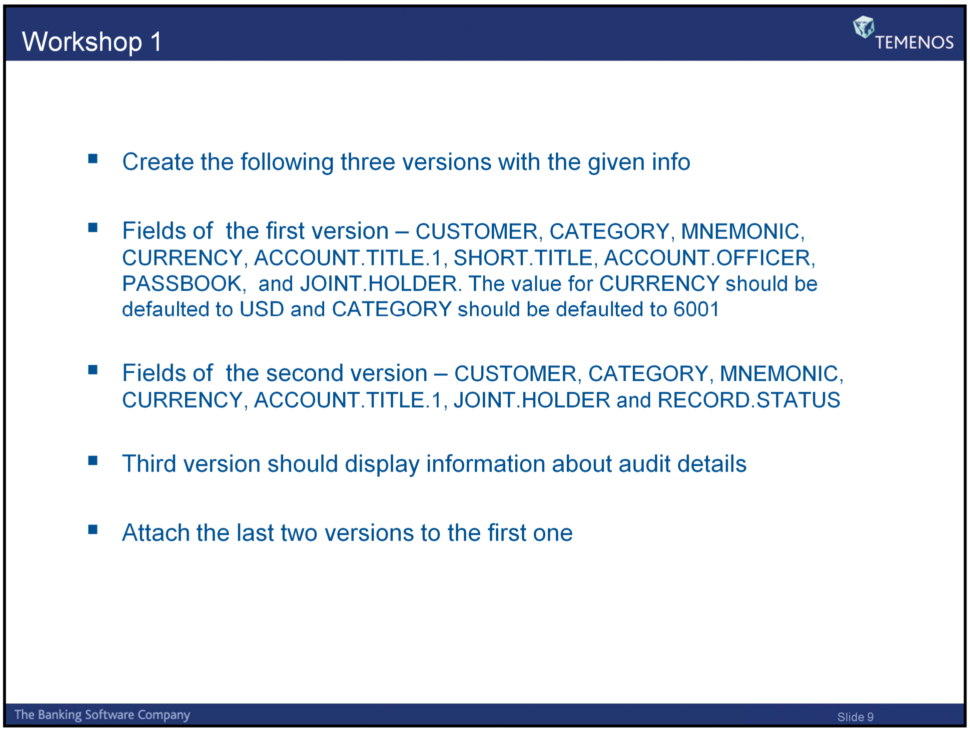
CUS24-Associated Versions and Next Versions-R09.01

Launch the main version. You can see the associated versions linked to the main version. The tabs shown are CUST.INFO,CONTACT.INFO and AUDIT.FIELDS.



You can click on the tabs of each associated version and view the contact information and the audit details in their respective tabs.

CUS24-Associated Versions and Next Versions-R09.01



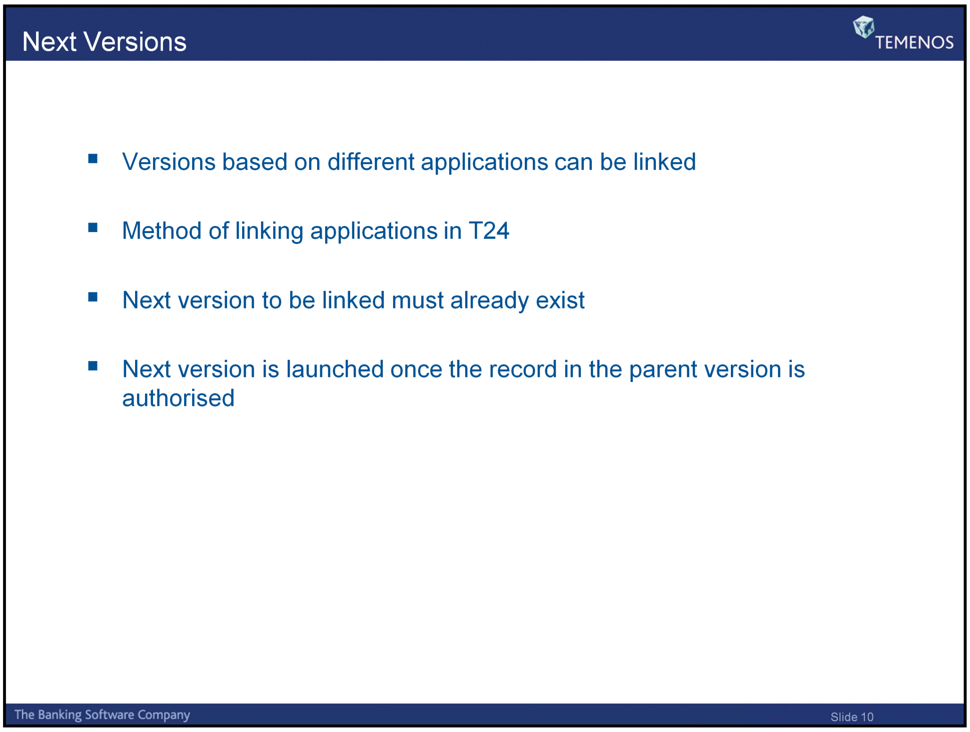
1. Create the following three versions with the given info

2. Fields of the first version are CUSTOMER, CATEGORY, MNEMONIC, CURRENCY, ACCOUNT.TITLE.1, SHORT.TITLE, ACCOUNT.OFFICER, PASSBOOK, and JOINT.HOLDER. The value for CURRENCY should be defaulted to USD and CATEGORY should be defaulted to 6001

3. Fields of the second version are CUSTOMER, CATEGORY, MNEMONIC, CURRENCY, ACCOUNT.TITLE.1, JOINT.HOLDER and RECORD.STATUS

4. The third version should display information about audit details

5. Attach the last two versions to the first one

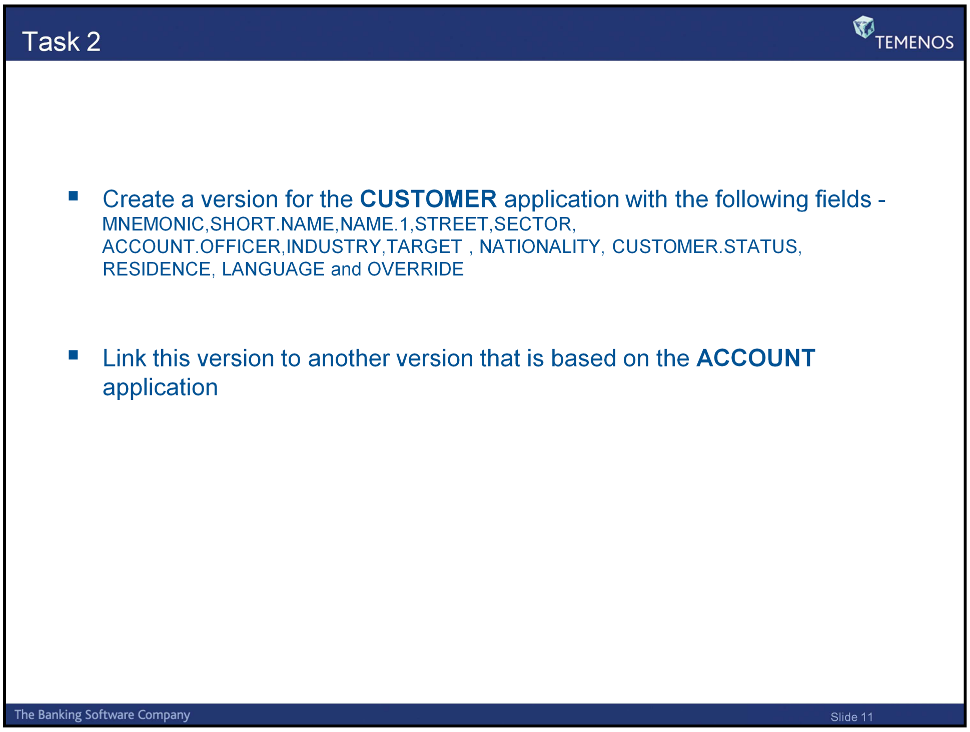


CUS24-Associated Versions and Next Versions-R09.01

1,2 . In T24 you can link versions that are based on different T24 application. This is done using Next Versions.

3. The versions need to be created before you link them to the parent version.

4. The next version is launched once you authorise the record in the parent version.



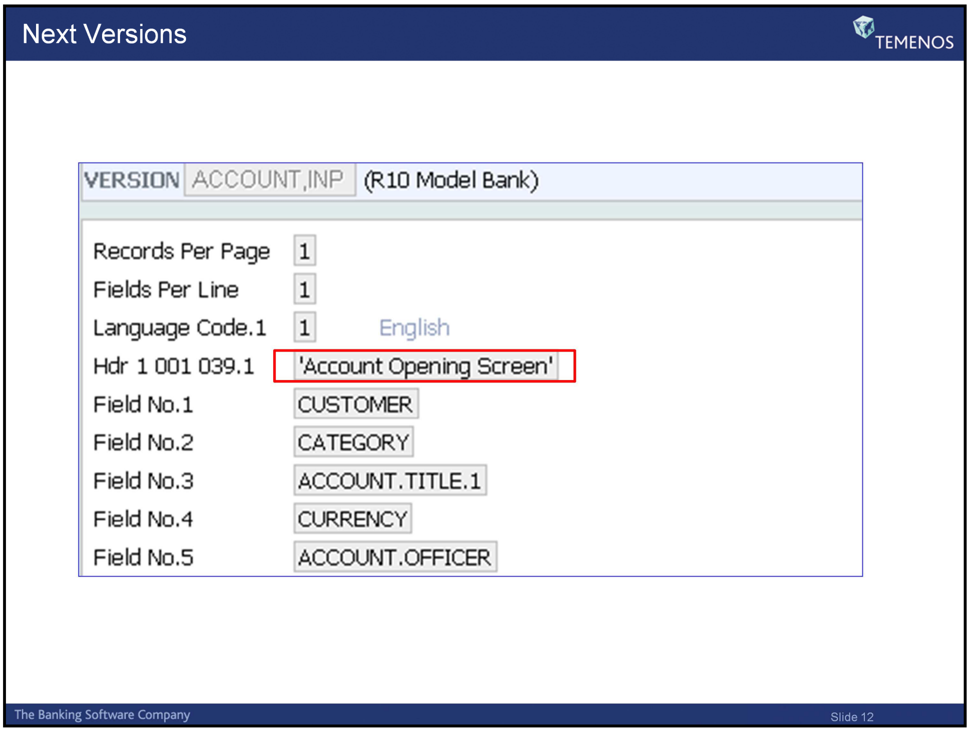
The stated task teaches you to create next versions in T24. The requirement is stated here as follows,

Can you link versions that are based on different applications in T24?

Yes. You can link Versions based on different applications

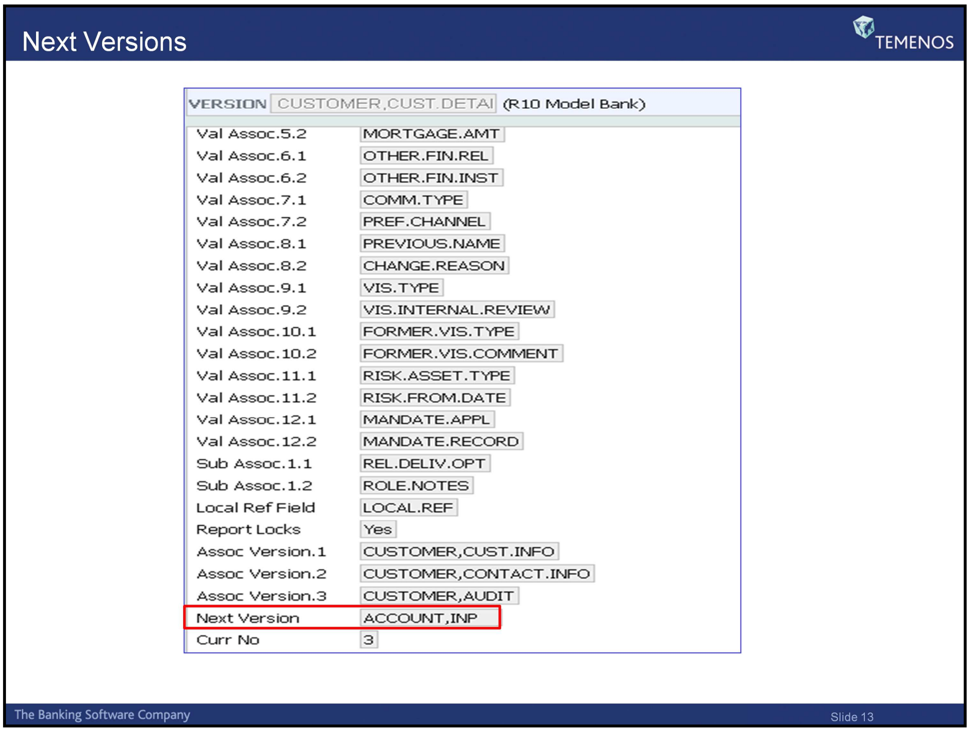
1.Create a version for the **CUSTOMER** application with the following fields - MNEMONIC,SHORT.NAME,NAME.1,STREET,SECTOR,ACCOUNT.OFFICER, INDUSTRY,TARGET,NATIONALITY,CUSTOMER.STATUS,RESIDENCE, LANGUAGE and OVERRIDE

2. This parent version will be linked to another version that is based on the **ACCOUNT** application



CUS24-Associated Versions and Next Versions-R09.01

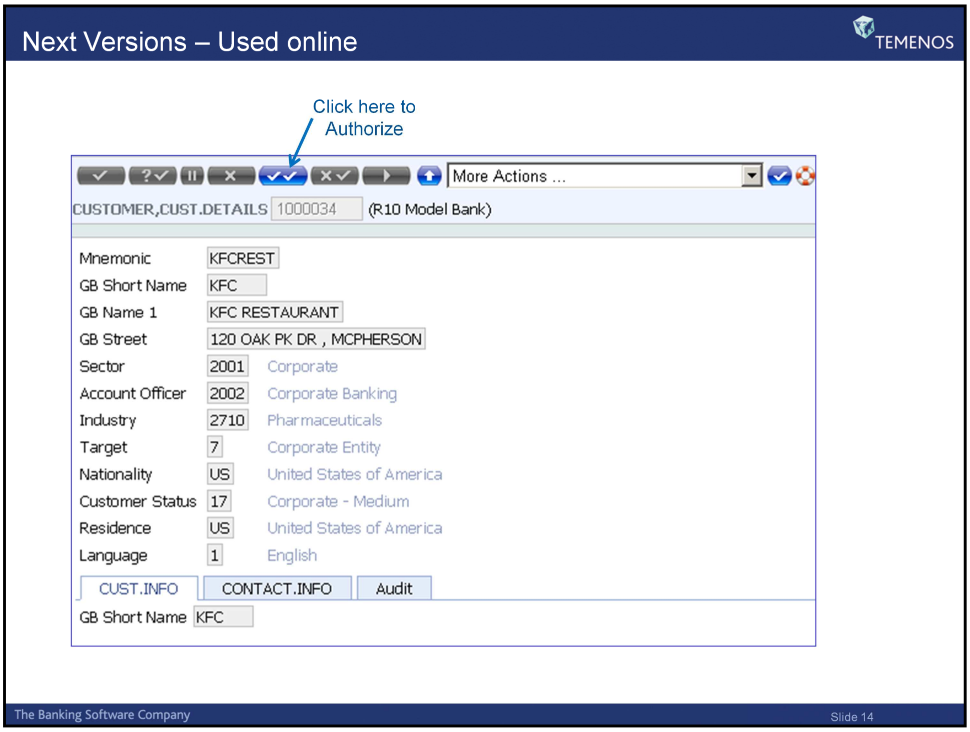
Create the version which is to be linked to the parent version. This version is based on the **ACCOUNT** application. You can take a look at the header of this **ACCOUNT** version.



CUS24-Associated Versions and Next Versions-R09.01

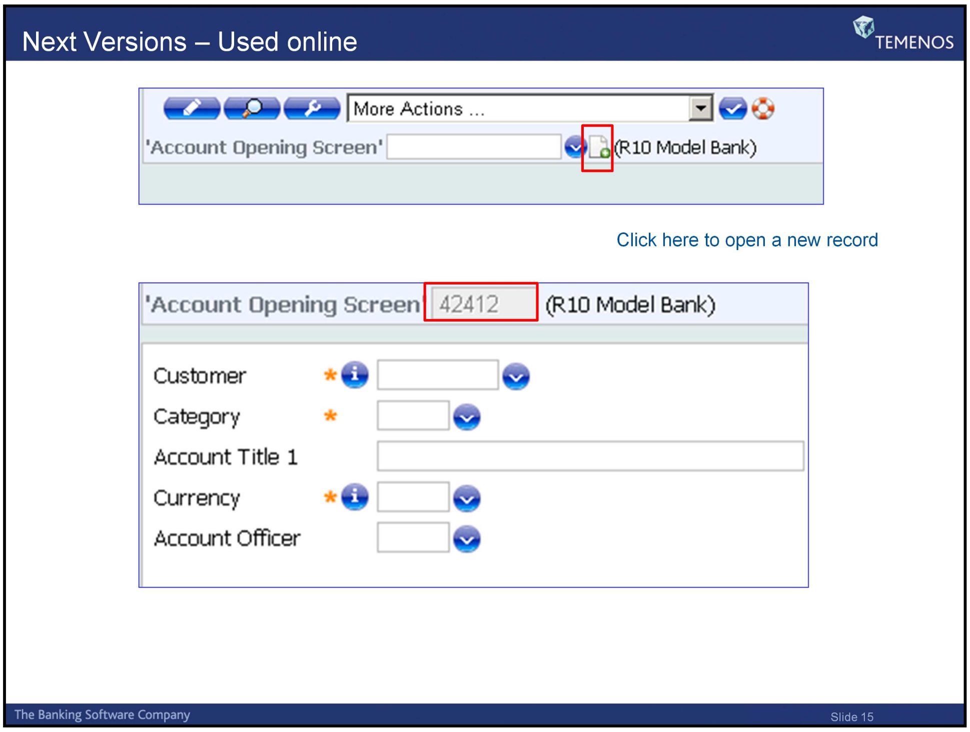
Create the parent version which is based on the **CUSTOMER** application. In this example, an ACCOUNT version is attached as a NEXT version in a CUSTOMER version.

NEXT.VERSION – In this field specify a valid version name. This version will be launched once you authorise the record in the parent version.



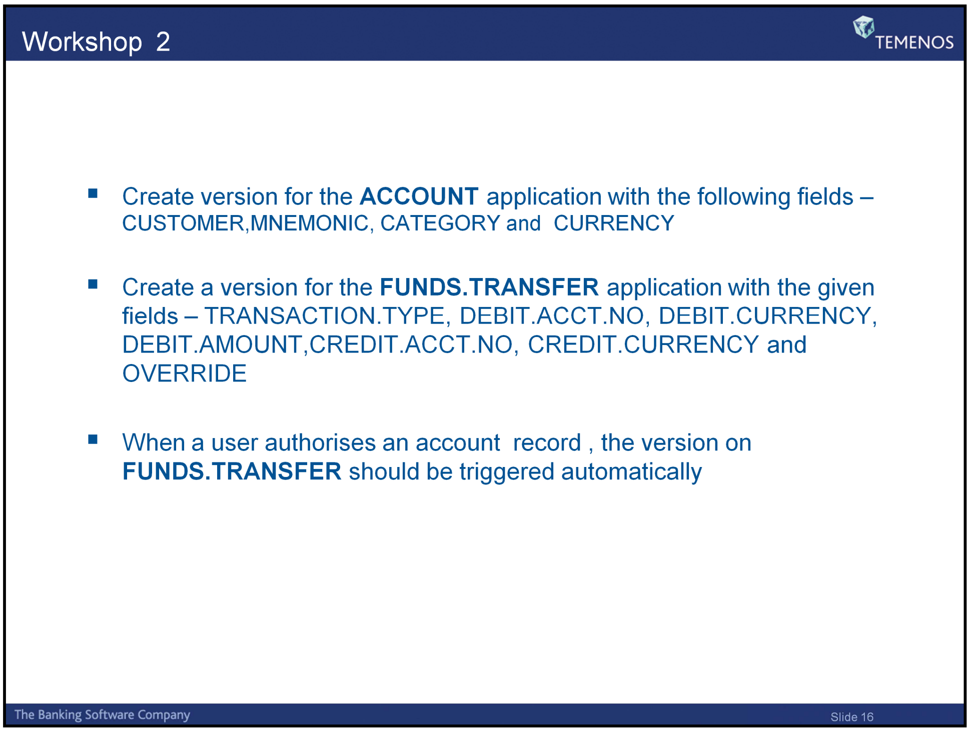
CUS24-Associated Versions and Next Versions-R09.01

Once you authorise the customer record in the parent version, your next version will be launched automatically.



CUS24-Associated Versions and Next Versions-R09.01

Your next version is launched. Click on the new deal icon to open a new account. Note the header of the **ACCOUNT** version.

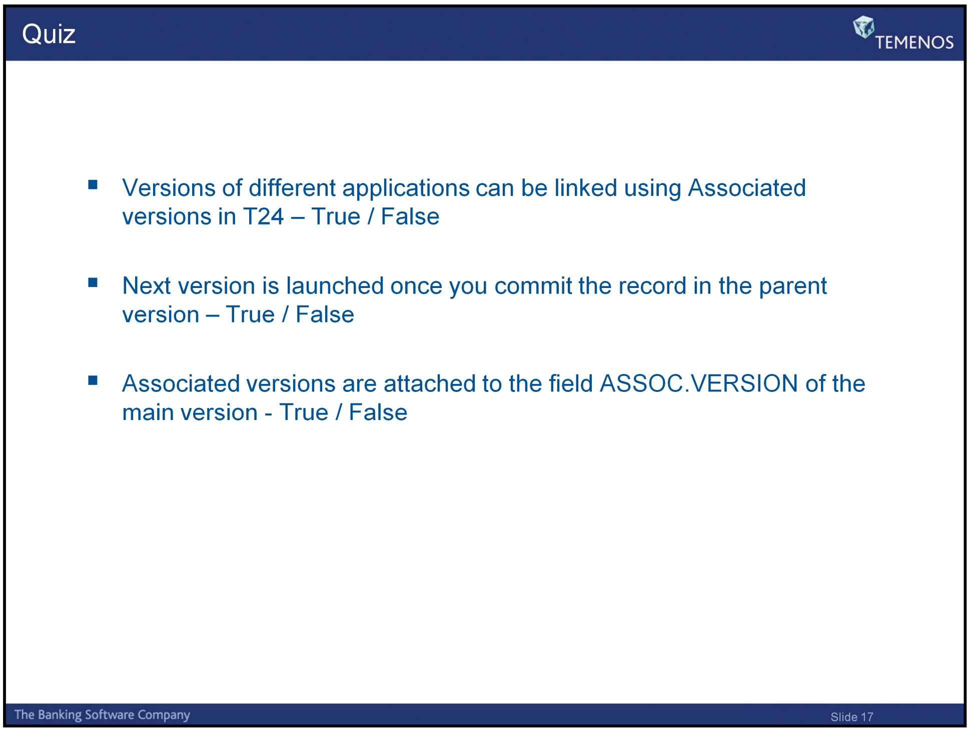


CUS24-Associated Versions and Next Versions-R09.01

1. Create version for the **ACCOUNT** application with the following fields – CUSTOMER,MNEMONIC, CATEGORY and CURRENCY

2. Create a version for the **FUNDS.TRANSFER** application with the given fields – TRANSACTION.TYPE, DEBIT.ACCT.NO, DEBIT.CURRENCY, DEBIT.AMOUNT,CREDIT.ACCT.NO, CREDIT.CURRENCY and OVERRIDE

3. When a user authorises an account record , the version on **FUNDS.TRANSFER** should be triggered automatically

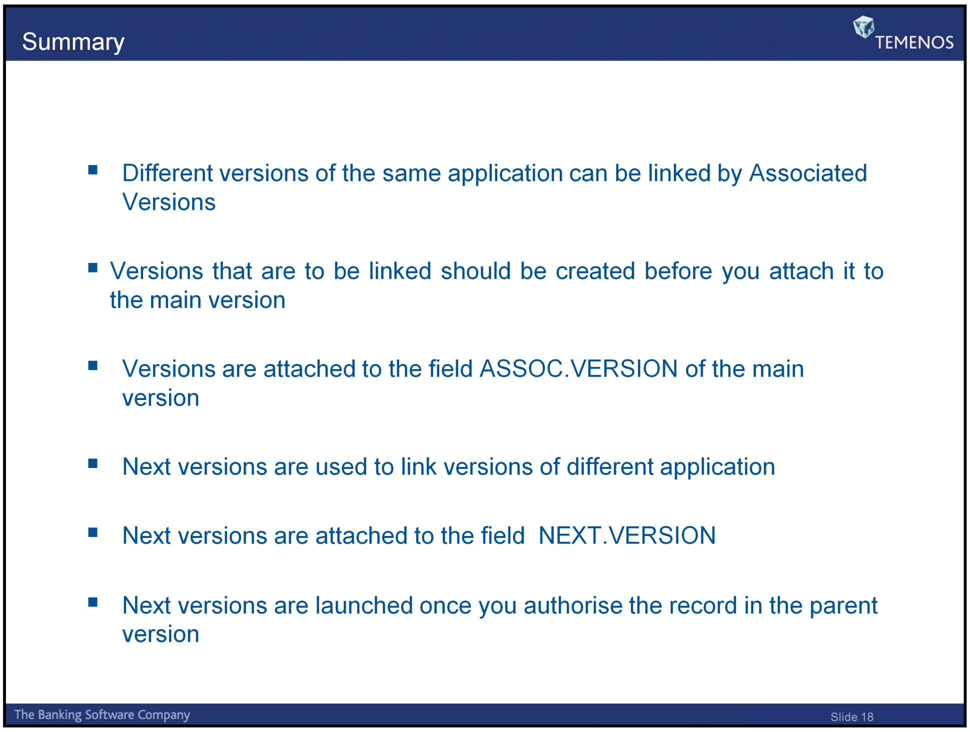


CUS24-Associated Versions and Next Versions-R09.01

1. False

2. False

3. True



CUS24-Associated Versions and Next Versions-R09.01

1. Different versions of the same application can be linked by associated versions

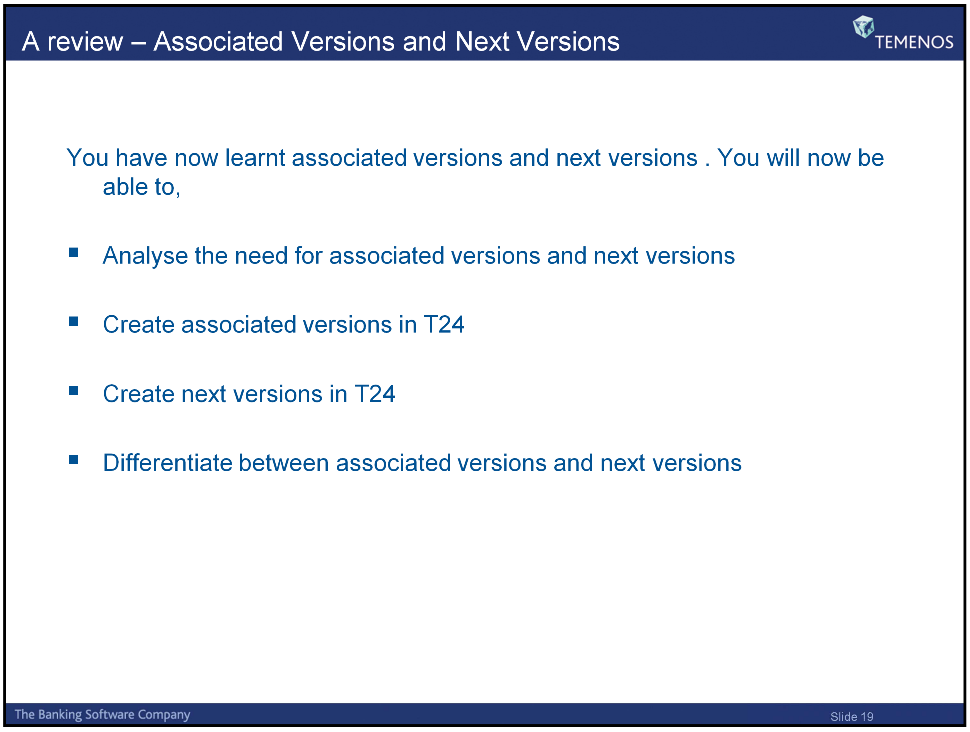
2. Versions that are to be linked should be created before you attach it to the main version

3. Versions are attached to the field ASSOC.VERSION of the main version

4. Next versions are used to link versions of different application

5. Next versions are attached to the field NEXT.VERSION

6. Next versions are launched once you authorise the record in the parent version



CUS24-Associated Versions and Next Versions-R09.01

You have now learnt associated versions and next versions . You will now be able to,

1. Analyse the need for associated versions and next versions

2. Create associated versions in T24

3. Create next versions in T24

4. Differentiate between associated versions and next versions

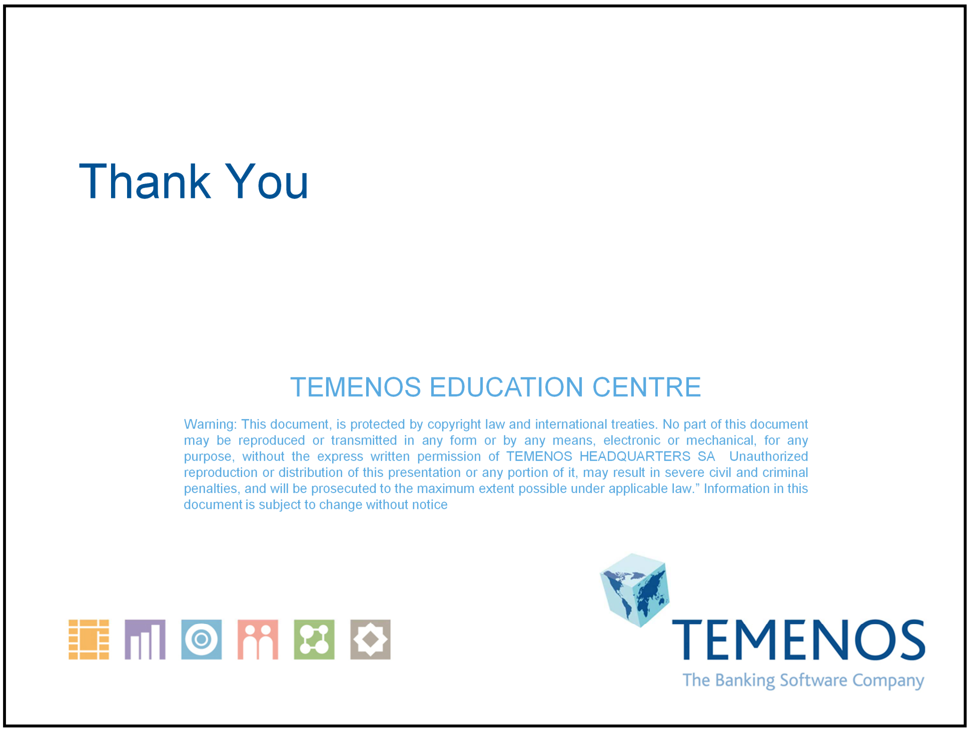
Version History

TEMENOS

Version	Date	Author	Reviewer
1.0			
1.1	13th Feb 2008	Nasim Kazi S	
1.2		Assis Priya	Sara Cleur
1.3	17th Dec 2009	Kripesh Krishnadas	

The Banking Software Company Slide 20

CUS24-Associated Versions and Next Versions-R09.01



CUS24-Associated Versions and Next Versions-R09.01