xRM Update Manual

ADITO Software GmbH

Version 2024.2.0 | 2025-02-28

Version	Changes
2024.2.0	Update to 2024.2.0
2024.1.1	Update to 2024.1.1
2024.1.0	Update to 2024.1.0
2024.0.2	Update to 2024.0.2
2024.0.0	Update to 2024.0.0
2023.2.0	Update to 2023.2.0
2023.0.2	Update to 2023.0.2
2023.0.0	Update to 2023.0
2022.2.0	Update to 2022.2
2022.1.0	Update to 2022.1
2022.0.0	Update to 2022.0
2021.2.0	First Version

Character Formatting

The following signs will point you to specific sections:

ℹ	Hints and notes.
💡	Tips and tricks.
❗	This is important!
⚠	Warning! These actions are dangerous and can result in data loss!

The following font formatting applies:

Font type	Meaning
Mask	The mask, table or button to which the section refers
"Mask"	Terms that originate from the system and that need to be emphasized in the reading flow
`code();`	Code and program parts

Index

Character Formatting 2

1. Overview 6

2. How to Upgrade your Project 6

3. Update from 2024.1.1 to 2024.2.0 6

3.1. General 6

3.1.1. Changed structure of Liquibase files 6

3.2. xRM Sales (Module: 0.1.0 sales-basic) 7

3.2.1. Refactoring of Advertising (AdvertisingItem) 7

3.2.2. Maintain cash discount (Skonto) separately 7

4. Update from 2024.1.0 to 2024.1.1 8

4.1. xRM Plattform 8

4.1.1. Changes for Language-files in modular projects: Projectupgrade necessary 8

4.1.2. Blobhandling set to PROCESS by default 8

4.2. xRM Sales 8

4.2.1. Refactoring calculation of prices in offeritems and orderitems 8

5. Update from 2024.0.2 to 2024.1.0 9

5.1. xRM Basic 9

5.1.1. Libs have been split due to modularisation 9

6. Update from 2024.0.0 to 2024.0.2 9

6.1. xRM Sales 9

6.1.1. Refactoring of Orderitem 9

6.2. xRM Service 10

6.2.1. New property in _PREFERENCES_PROJECT for Resource planning 11

7. Update from 2023.2.2 to 2024.0.0 11

7.1. xRM Sales 11

7.1.1. Refactoring of OFFERITEM 11

7.1.2. Refactoring of Product/Productprice 12

8. Update from 2023.2.1 to 2023.2.2 13

8.1. xRM Plattform 13

8.1.1. Changes to columns of table AB_LOGHISTORY 13

8.2. xRM Service 13

8.2.1. Place of use added to email filters 13

9. Update from 2023.2.0 to 2023.2.1 14

9.1. xRM Collaboration 14

9.1.1. Controlling of Projects and booked times 14

9.2. xRM Sales 15

9.2.1. Header and footer texts classified by receipt type 15
9.2.2. Drag & Drop for offer items 15
10. Update from 2023.0.2 to 2023.2.0 16
10.1. xRM Sales 16
10.1.1. Offer_entity and Order_entity 16
10.2. xRM Contact Management 16
10.2.1. DocumentTemplate 16
10.3. xRM Sales 16
10.3.1. Topictree 16
10.3.2. Productprice_entity 17
10.4. xRM Service 17
10.4.1. TASK and TICKET got separated 17
11. Update from 2023.0.0 to 2023.0.2 19
11.1. xRM 19
11.1.1. Update Process for MariaDB Collation 19
11.2. xRM Sales 19
11.2.1. ClassificationUpdateUtils 19
12. Update from 2022.2.X to 2023.0.0 20
12.1. xRM Sales 20
12.1.1. TOPICTREEUSAGE table added 20
12.1.2. DISTRICTCONTACT changed 20
12.2. xRM Service 20
12.2.1. Unique and general views for knowledge articles 20
12.2.2. Changed Datatypes 21
12.2.3. Renaming of the Visitplanner 22
13. Update from 2022.1.X to 2022.2.0 22
13.1. xRM Contact Management 22
13.1.1. migrateCommunicationType_serverProcess 22
13.2. xRM Sales 22
13.2.1. TOPICTREE normalization 22
13.2.2. APPOINTMENT_ID dropped from VISITPLANENTRY 23
13.2.3. New territories table 23
13.2.4. New column PROGRESS in SALESPROJECT 24
13.3. xRM Plattform 24
13.3.1. Version table 24
14. Update from 2022.0.X to 2022.1.0 24
14.1. xRM Contact Management 24

14.1.1. OBJECT_TYPE and OBJECT_ROWID for ADDRESS 24
14.1.2. Observation_BG 25
14.1.3. OBJECT_TYPE and OBJECT_ROWID for TASKS in TASKATTENDEES 25
14.2. xRM Sales 25
14.2.1. VISITRECOMMENDATION to VISITPLANENTRY 25
14.2.2. insert_ab_currency.xml in release folder 25
15. Update from 2021.2.X to 2022.0.0 26
15.1. xRM Contact Management 26
15.1.1. Duplicate pattern 26
15.1.2. Renaming OrgUtils.getDSGVOAnonymisierungOrganisationId 26
15.1.3. Communication OBJECT_TYPE 26
15.1.4. Groupable subquery fields 26
15.2. xRM Sales 27
15.2.1. Multi-currency capability 27
15.2.2. DISCOUNTED_NET and DISCOUNTED_VAT dropped 27
15.2.3. Default Values and Constraints for OFFER/SALESORDER/SALESPROJECT 28
15.2.4. ADVERTISINGITEM changes 28
15.3. xRM Marketing 29
15.3.1. Advertising settings 29
15.4. xRM Plattform 29
15.4.1. New local development database 29
16. Update from 2021.1.X to 2021.2.0 29
16.1. xRM Contact Management 29
16.1.1. Duplicatescanner 29
16.1.2. Contact synchronization (EWS) 30
16.1.3. Export templates 30
16.1.4. Anonymization organisation 31
16.2. xRM Sales 31
16.2.1. Districts were expanded by visit frequency 31
16.2.2. Datatypes changes to default columns 31
16.2.3. Datatype change to PROBABILITY 32
16.2.4. Topic Tree 32
16.2.5. Microsoft Teams integration 32
16.3. xRM Plattform 33
16.3.1. $local.idvaluesExcluded 33

1. Overview

This document explains the following topics:

- How to Upgrade your xRM
- What Changes were done in each xRM Team that affects the usage of the xRM

2. How to Upgrade your Project

First and foremost you need to backup your system to be able to roll back your changes in case of issues. Pull the stable branch of the xRM Git project. Update and merge all remaining branches with a stable branch which you will use as your update branch. This prevents unnecessary merge conflicts later when merging your update into your working branches. Create a new branch which will be used to upgrade your project. Merge the xRM branch in your newly created project branch. Resolve any upcoming conflicts but always check if your changes are accepted as well. Test your merged branch and resolve any problems that may occur during testing. Commit und push the branch in your repository. Others can now use your created branch as their new master.

> ! This is a DEV-Version of this document. This is still in development. Breaking-Changes may not be listed in this version!

3. Update from 2024.1.1 to 2024.2.0

3.1. General

3.1.1. Changed structure of Liquibase files

As of version 2024.2.0, the structure of the Liquibase files has been changed and modularised.
This required the Liquibase scripts to be completely regenerated from the current database state.

> ! As a result, there are no longer any update scripts in the generated scripts.

If customer projects now want to update to a higher version, e.g. 2024.1.0 to 2024.2.3, they must first update to 2024.1.3.

Update scripts are still included in 2024.1.3. The 2024.2.0 is the same as the 2024.1.3 but without the update scripts.
You can then upgrade to 2024.2.3.

3.2. xRM Sales (Module: 0.1.0 sales-basic)

3.2.1. Refactoring of Advertising (AdvertisingItem)

The Advertising feature has been changed to work with the current Currency process. Unused and obsolete functions have been removed and other functions and libs have been marked as deprecated and will be removed in future updates.

This change reduces the complexity of the implemented functions, which will help maintain and update the Advertising feature in the future.

> ⚠ Please ensure that you do not use any deprecated functions or features in your projects, as they will be removed in future updates. Also make sure that you do not use any of the removed features.

Removed:

- **Advertising_lib:** Removed `AdvertisingItemUtils` and `AdvertisingUtils.getProductObject()`
- **AdvertisingItem_entity:** Removed `ProductObject_param`
- **OfferOrder_lib:** Removed all `ItemUtils.prototype.<functionName>()` functions

⍰ These libs and functions are no longer in use and have been removed.

Deprecated:

- **Item_lib**: Deprecated `MoveableItemUtils` and `MoveableItemOrderUtils`

⍰ These functions are no longer used. The tables now use Lexorank for drag and drop.

3.2.2. Maintain cash discount (Skonto) separately

1. **What changes in detail?**
   Cash discounts and payment terms are now stored in seperate data records. Previously, staggered discounts were not possible.
   e.g. payment term ‘30 days net, 7 days 3% cash discount’ is mapped to payment term ‘30 days net’ and a new cash discount entry is created in a new table with ‘7 days 3% cash discount’.
2. **How does it affect the change(s) to the previous logic?**
   The mapping of how existing payment terms are mapped to the new data structure must be maintained in the `migrate_cashDiscount_serverProcess` process. An example is stored in the process.
   The process must then be executed. Cash discount entries are created to match the mapping.

3. **Background to the change (Why was this changed)?**
   Only one cash discount entry could be maintained per data record, staggered cash discounts were not possible. It was technically cumbersome to read out the data because the payment term and cash discount were combined in one keyword.

4. Update from 2024.1.0 to 2024.1.1

4.1. xRM Plattform

4.1.1. Changes for Language-files in modular projects: Projectupgrade necessary

In modular projects (projects for modules) changes for the Language-files were organized by Modifications. In larger projects continous changes for the Language-files resulted in many Modification-files, thus making the transpiling time of these projects longer and rendering working in those projects unbearable.

Because of this, with plattform version 2024.1.1 these modifications will be summarized into distinct Language-files, which makes working with moduls faster and easier. Please check if all translations are present after upgrading the project.

> ℹ Please note that this will happen automatically once you upgrade your project within a 2024.1.1 ADITO Designer. This will not affect non-modular-projects as no sources will be affected in them. Aside from that, the projectupgrade is still necessary for non-modular-projects.

4.1.2. Blobhandling set to PROCESS by default

The property blobHandlingMethod is set to PROCESS within the Projekt Preferences: `<blobHandlingMethod>PROCESS</blobHandlingMethod>`

Also a new property called `custom.database.binarypath` has been introduced within the custom property. The configured path in this property will be used within the blobhandler-process.

In prior versions the blobHandlingMethod was set to DATABASE. Make sure that writing and reading binary files works in your project after upgrading.

4.2. xRM Sales

4.2.1. Refactoring calculation of prices in offeritems and orderitems

To ensure that the prices of super-items do not differ from the sum of the sub-items, super-items are now automatically calculated from the sub-items. As previously with free text items.

Check your code for any of the old methods listed below and change your code so it uses the new implementations.
Each function is documented within the library.

OrderItemtem_lib:

- **OrderItem.updateFreeTextItemPrices** renamed to **OrderItem.updateParentItemPrices**

Offer_lib:

- **OfferItemUtils.updateFreeTextItemPrices** renamed to **OfferItemUtils.updateParentItemPrices**

OfferOrder_lib:

- **ItemUtils.getItemSum**: The parameter **pOptional** has been removed. Optional items are calcualated exactly the same as non-optional items now. Optional items will now be displayed with the full price instead of 0.

5. Update from 2024.0.2 to 2024.1.0

5.1. xRM Basic

5.1.1. Libs have been split due to modularisation

Please note that starting with this release, libs in our basic project (called "xrm/basic" in Gitlab) have been split due to the restructuring of the basic project into multiple modules.

So please be aware of this change for future merges of this project into your projects.

For example, Attribute_lib contained the class/function `AttributeUtil'. This class/function is now contained in its own lib: AttributeUtil_lib.

Also common functions like `AttributeUtil.getPossibleUsageContexts` which refer to different contexts may have been adapted.

6. Update from 2024.0.0 to 2024.0.2

6.1. xRM Sales

6.1.1. Refactoring of Orderitem

To match the refactoring of Offeritem (see below), Orderitem has been refactored as well.

Check your code for any of the old methods listed below and change your code so it uses the new implementations.
Each function is documented within the library.

Order_lib:

- **OrderItemUtils.updateOrderNet** replaced by **OrderUtils.updateOrderSum**

OrderItem_lib:

- **OrderItemUtils.prototype.recalculatePrices** replaced with **OrderUtils.updateOrderSum()**
- **OrderItemUtils.prototype.getNetAndVat** replaced with **OrderItem.getTotalFromOrderItems()**
- **OrderItemUtils.prototype.initItemTree** removed
- **OrderItemUtils.prototype.getItemSum** replaced with **OrderItem.getTotalFromOrderItems()**
- **OrderItemUtils.prototype.getItemVAT** replaced with **OrderItem.getTotalFromOrderItems()**
- **OrderItemUtils.prototype.roundPrice** removed
- **OrderItemUtils.prototype.insertPartsList** replaced with **OrderItem.insertProductPartsList()**
- **OrderItemUtils.prototype.deletePartsList** replaced with **OrderItem.deleteOrderItemsAndTheirChildren()**
- **OrderItemUtils.prototype.getNextItemSort** replaced with **OrderItem.getMetaData()**
- **OrderItemUtils.prototype.getNextItemPosition** replaced with **OrderItem.getMetaData()**
- **OrderItemUtils.prototype.reOrgItems replaced** with **OrderItem.refreshItemPositions()**
- **OrderItemUtils.isEditable** replaced with **OrderItem.isEditable**

ProductPackage_lib:

function definition of ProductPackage.insertProducts as been changed.
Old: ProductPackage.insertProducts = function(pOfferId, pProductPackageId, pAssignedTo, pContactId, pSkipPartsList)
New: ProductPackage.insertProducts = function(pContext, pRowId, pProductPackageId, pAssignedTo, pSkipPartsList)

Context OfferAddProductPackage:

The context as well as all entities and views of **OfferAddProductPackage** have been renamed to **ProductPackageAddItems**.

6.2. xRM Service

6.2.1. New property in _PREFERENCES_PROJECT for Resource planning

In _PREFERENCES_PROJECT a new property named **showAppointmentsInEPK** is set to true by default. This property controls whether or not appointments will be displayed within the resource planning view (EPK).
If true, appointments will be displayed.

7. Update from 2023.2.2 to 2024.0.0

7.1. xRM Sales

7.1.1. Refactoring of OFFERITEM

To make it easier to extend the offer item construct, it got refactored. This entails the removal of the inheritance from OfferItemUtils to OfferItem. Functions of OfferItemUtils either got replaced by static methods or were removed.

> Check your code for any of the old methods listed below and change your code so it uses the new implementations.
> Each function is documented within the library.

- Replaced functions:

Old function	Replacement
OfferItemUtils.prototype.reOrgItems()	OfferItemUtils.refreshItemPositions()
OfferItemUtils.prototype.getNextItemPosition()	OfferItemUtils.getMetaData()
OfferItemUtils.prototype.getNextItemSort()	OfferItemUtils.getMetaData()
OfferItemUtils.prototype.insertPartsList()	OfferItemUtils.insertProductPartsList()
OfferItemUtils.prototype.getItemVAT()	OfferItemUtils.getOfferSum()
OfferItemUtils.prototype.getItemSum()	OfferItemUtils.getOfferSum()
OfferItemUtils.prototype.getNetAndVat()	OfferItemUtils.getOfferSum()

OfferItemUtils.prototype.recalculatePrices()	OfferItemUtils.updateOfferSum()

- Removed functions

Function
OfferItemUtils.prototype.traverse()
OfferItemUtils.prototype.initItemTree()
OfferItemUtils.prototype.roundPrice()

7.1.2. Refactoring of Product/Productprice

Simplification of the pricing logic, as the prices were converted several times between foreign and local currency. From now on only directly in offer or order.
Prices can now be calculated and updated automatically from the parts list.

Due to the changes usages of PRODUCTPRICE.CURRENCYRATE_ID / CURRENCY and TAKEPRICE need to be adapted.
The date of foreign currencies of the column/field PRODUCTPRICE.VALID_TO is updated via a liquibase script to date of yesterday, henceforth invalid for further usages.

New prices can only be added with the local currency.
Calls of the changed functions need to be adapted.

Changes in database tables

- PRODUCTPRICE.VALID_FROM changed datatype from Timestamp to Date
- PRODUCTPRICE.VALID_TO changed datatype from Timestamp to Date
- PRODUCTPRICE.CURRENCYRATE_ID removed
- PRODUCTPRICE.CURRENCY removed
- PRODUCTPRICE.TAKEPRICE removed

Changes in functions

- ProductUtils.getCurrentProductPrice(…) parameter pCurrency removed
- ProductUtils.presetPriceAndVat(…) removed

Changes in entities

- Offeritem_entity: Field itemInsertStatements removed

8. Update from 2023.2.1 to 2023.2.2

8.1. xRM Plattform

8.1.1. Changes to columns of table AB_LOGHISTORY

Columns `TABLENAMEID` and `SOURCE_TABLENAMEID` were changed to be varchar and their length got expanded to 63 characters. Update the tables via liquibase.

8.2. xRM Service

8.2.1. Place of use added to email filters

As the email bounce detection was also added to service emails, the email filters got expanded by a "place of use" functionality. For that the new table `EMAILFILTERPLACEOFUSE` was introduced. This table links a place of use to an email filter. So one filter can have N places of use. The column `PLACEOFUSE` contains the name of the context, where this filter should take effect. I.e. "BulkMail" or "SupportTicket".

To add this new table, execute the liquibase update.

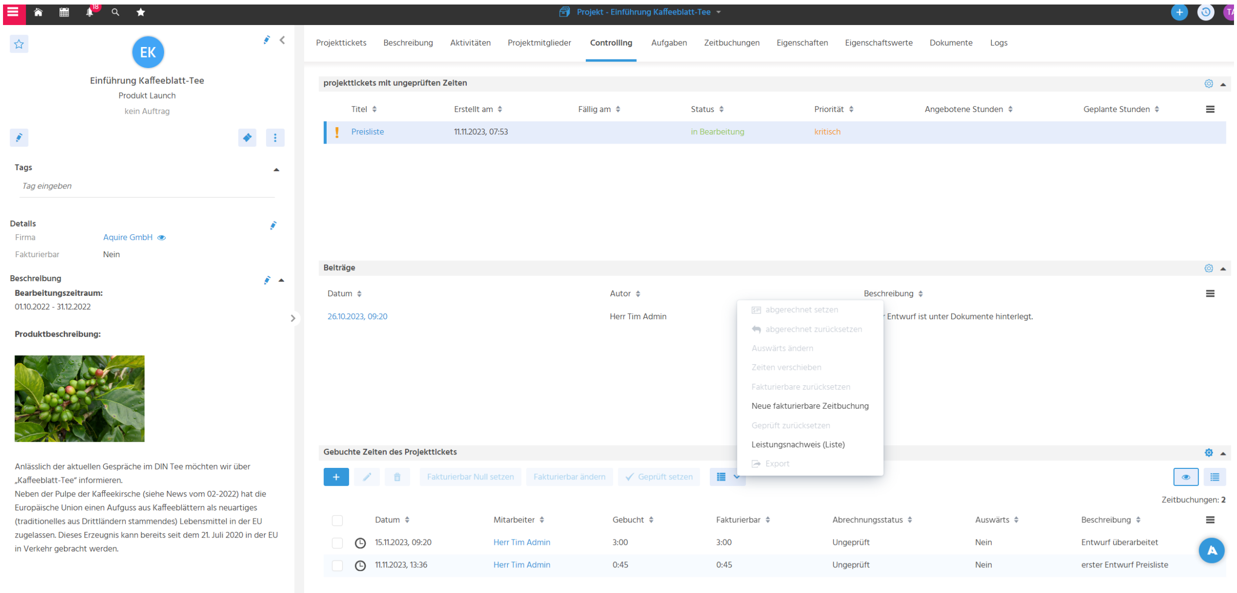
To load filters based on their place of use, a new parameter was added to the function `IncomingEmailFilterProcessor.prototype.loadFilters` of `EmailFilterHandling_lib`.

> Make sure to search your project for `.loadFilters(` to find every use of the function and add the applicable place of use. For example: "BulkMail" in the respective library or mailbridge processes.

9. Update from 2023.2.0 to 2023.2.1

9.1. xRM Collaboration

9.1.1. Controlling of Projects and booked times



Einführung Kaffeeblatt-Tee
Produkt Launch
kein Auftrag

Details

Firma	Aquire GmbH
Fakturierbar	Nein

Beschreibung

Bearbeitungszeitraum:
01.10.2022 - 31.12.2022

Produktbeschreibung:

Anlässlich der aktuellen Gespräche im DIN Tee möchten wir über „Kaffeeblatt-Tee“ informieren.
Neben der Pulpe der Kaffeekirsche (siehe News vom 02-2022) hat die Europäische Union einen Aufguss aus Kaffeeblättern als neuartiges (traditionelles aus Drittländern stammendes) Lebensmittel in der EU zugelassen. Dieses Erzeugnis kann bereits seit dem 21. Juli 2020 in der EU in Verkehr gebracht werden.

projekttickets mit ungeprüften Zeiten

	Titel	Erstellt am	Fällig am	Status	Priorität	Angebotene Stunden	Geplante Stunden
!	Preisliste	11.11.2023, 07:53		in Bearbeitung	kritisch		

Beiträge

Datum	Autor	Beschreibung
26.10.2023, 09:20	Herr Tim Admin	Entwurf ist unter Dokumente hinterlegt.

Gebuchte Zeiten des Projekttickets

Zeitbuchungen: **2**

Datum	Mitarbeiter	Gebucht	Fakturierbar	Abrechnungsstatus	Auswärts	Beschreibung
15.11.2023, 09:20	Herr Tim Admin	3:00	3:00	Ungeprüft	Nein	Entwurf überarbeitet
11.11.2023, 13:36	Herr Tim Admin	0:45	0:45	Ungeprüft	Nein	erster Entwurf Preisliste

The main view of the projects got expanded by a new tab, which lets you control and manage booked times. The first table shows all tickets with unchecked time bookings, the following two tables show the comments and booked times of the selected ticket. Within the third table you can select 1 to n booked times for executing an action on them.

These new actions are:

- set extern time null
  Sets the time to zero, so it won’t get billed.
- change extern time
  Changes the billable time to the chosen value.
- set checked
  Sets the checked flag to checked, which denotes a time as checked for billing.
- change away
  Sets the Away flag.
- move Ticket Time
  Moves the time to another ticket.
- reset extern time
  Sets the extern time back to the booked time.

- new extern time
  Creates a pseudo time whose booked time is zero, but the extern time can be set for billing.
- set unchecked
  Sets the checked flag to unchecked, so the time cannot be billed.
- Leistungsnachweis (Liste) (sic not yet translated)
  Creates a report which lists all times.
- Export
  Exports the times as a CSV file.

These actions are also available in the tab "booked times" in the project and project ticket contexts.

9.2. xRM Sales

9.2.1. Header and footer texts classified by receipt type

The classifications "Receipt header text" and "Receipt footer text" were replaced by a more granular classification. This makes it possible to find the right text blocks more easily.

The new classifications are:

- Order confirmation header text
- Order confirmation footer text
- Delivery note header text
- Delivery note footer text
- Invoice header text
- Invoice footer text
- Credit header text
- Credit footer text
- Cancellation header text
- Cancellation footer text

9.2.2. Drag & Drop for offer items

The column ITEMSORT was removed from the table OFFERITEM and was replaced by LEXORANK. The migration is done by the liquibase scripts. So execute the new liquibase scripts when updating to safely migrate your existing offer items.

> ! Check for any occurrences of the column ITEMSORT and customize accordingly.

> These occurrences either have to be removed or changed so they use LEXORANK and its functions to sort.

10. Update from 2023.0.2 to 2023.2.0

> Important Changes!

10.1. xRM Sales

10.1.1. Offer_entity and Order_entity

Field OFFER_CODE of Offer_entity and field ORDER_CODE of Order_entity got removed as their naming was confusing. Instead SALESORDER_ID is used in Order_entity and OFFER_ID is used in Offer_entity to derive order / offer codes.

> You have to check if ORDER_CODE and OFFER_CODE are used anywhere else and adjust those parts to SALESORDER_ID and OFFER_ID.

10.2. xRM Contact Management

10.2.1. DocumentTemplate

Document templates got changed to provide an easier way of creating templates with attachments. Attachments are now placed in the `Documents` tab of a template instead of creating its own template of type "attachment".

So the context, entity and database table `DocumentTemplateLink` aren’t necesarry anymore and got deleted. When updating your project, you have to run a liquibase update to remove the database table. The entity and context get deleted when merging your project with the most recent release.

> You have to check for any customizing which uses either the `DOCUMENTTEMPLATELINK` table or the `DocumentTemplateLink_entity` and adjust it accordingly, otherwise it won’t be working after updating!

10.3. xRM Sales

10.3.1. Topictree

The Topictree got expanded with drag & drop functionality, which replaces the move actions. For the drag & drop to work performantly the sorting was changed to using LexoRanks. The LexoRank algorithm (prominentally used by JIRA) allows for higher performance when changing the position of items in an ordered list or tree, because postions are based on the average between the two target nodes and only the moving nodes value has to be upadted instead of updating all values below the target postion. Sorting the lists or trees is also more effictient as this ranking uses simple strings as values for effincient alphabetical sorting.

To get this to work in the Topictree, the fields `SORTINGLAYER` and `SORTINGVALUE` from `TopicTree_entity` and the columns of the same name from the database table `TOPICTREE` got deleted. Instead of these the column and field `LEXORANK` got added.

In order to update, a liquibase script is provided, which changes the database structure and converts the old values of `SORTINGLAYER` and `SORTINGVALUE` into new, combined LexoRank values.

> ⚠ You have to check for any implementations outside of the Topictree which may use the deleted entity fields or table columns and change those parts to work with the LexoRanks instead! Leaving the old logic in place will result in error after updating.

10.3.2. Productprice_entity

The entity fields INFO, DELIVERYTERM and PAYMENTTERM were removed from the entity because they had no associated database fields or other functionality attached to them.

10.4. xRM Service

10.4.1. TASK and TICKET got separated

The service tickets have used database columns and keywords that were shared with tasks, which leads to unnecessary dependencies between them. This has been solved now.

To update you have to do the following steps:

1. **Execute Liquibase files to update the database structure.**
    - These columns are added to the table TICKET via Liquibase:
        - SUBJECT
        - STATUS
        - PRIORITY
        - REQUESTOR_CONTACT_ID
        - START_DATE
        - MATURITY_DATE

> ❗ If additional columns of TASK were used in your customizing of Serviceticket, you have to add those to the Liquibase file as well!

- Existing Ticket entries will then get deleted from TASK.

2. **Execute the process `migrate_taskTicket_serverProcess`**
   - This process updates existing filters which refer to the migrated columns. E.g. TASK.STATUS will be replacd by TICKET.STATUS.
   - The following filters will be updated automatically:
     - Ticket escalation filter
     - Employeegroup-Rulegroups
     - Checklist filter
     - Oberservation filter
     - Dashlet filters
     - Entity filter of SuppertTicket_entity
   - Important to know:

     > ❗
     > - If there are additional filters in your customizing which refer to the migrated columns, these filters have to be added to the process before executing it.
     > - The same is true for additional columns which got migrated from TASK to TICKET. E.g. TASK.PROGRESS → TICKET.PROGRESS
     > - It may happen that dashlets cannot be updated correctly, which may lead to errors being throw. In this case the user has to delete the dashlet and create it again.

3. **New keyword categories for Serviceticket**
   - In the past Serviceticket used keyword categories of tasks and activities. This got refactored and Serviceticket now uses its own keyword categories.
   - If you expanded the keyword categories of tasks or activities, you have to move those additions to the new keyword categories. Affected categories are:
     - TaskPriority → TicketPriority
     - ActivityCategory → TicketSource
     - TaskStatus → TicketStatus

- For each status an new icon has to be added via a keyword attribute.

11. Update from 2023.0.0 to 2023.0.2

Important Changes!

11.1. xRM

11.1.1. Update Process for MariaDB Collation

1. **What changes in detail?**

   For all columns of the xRM an update script was provided in 2023.0.2 which changes the character set to "utf8mb4". To also clean up all fields from customizing, there is the process updateMariaDBCollation_serverProcess. With this process you can either create a liquibase file or change the character set directly (Boolean EXPORT_TO_LIQUIBASE).

   For future Liquibase files, the following lines should be added to the changeset when creating new columns with type nclob or nvarchar in order to create the columns correctly:

   ```
   <modifySql dbms="mysql,mariadb">
               <regExpReplace replace="NVARCHAR\(([^\)]*)\)" with="VARCHAR($1) CHARACTER SET utf8mb4" />
               <replace replace="LONGTEXT CHARACTER SET utf8" with="LONGTEXT CHARACTER SET utf8mb4" />
   </modifySql>
   ```

2. **Background to the change (Why was this changed)?**

   On MariaDB and MySQL databases nvarchar interprets as "character set utf8" which is only a limited part of the utf8 character set and does not include emojis for example.

11.2. xRM Sales

11.2.1. ClassificationUpdateUtils

1. **What changes in detail?**

   The following methods have been moved to a new library

   ClassificationUtils.pagingSizeLoadEntity → ClassificationUpdateUtils.pagingSizeLoadEntity

   ClassificationUtils.executeUpdating → ClassificationUpdateUtils.executeUpdating

2. **How does it affect the change(s) to the previous logic?**

   Every place of use of these methods have to be updated.

3. **Background to the change (Why was this changed)?**

   The wrong declaration of the function led to an error when importing from ClassificationUtils of

Classification_lib as well

Merge Request

12. Update from 2022.2.X to 2023.0.0

12.1. xRM Sales

12.1.1. TOPICTREEUSAGE table added

1. **What changes in detail?**
   TOPICTREE has got a new table TOPICTREEUSAGE to assign usage locations for single topics.
2. **How does it affect the change(s) to the previous logic?**
   If you already use the topic tree, you have to maintain the usage location for all your existing topics - otherwise the lookup for the respective contexts is simply empty.
3. **Background to the change (Why was this changed)?**
   So that one can define and use various Topic trees in different contexts.

12.1.2. DISTRICTCONTACT changed

1. **What changes in detail?**
   DISTRICTCONTACT: RULE_ACTIVE has received a default value of 0 and a notNullConstraint.
2. **How does it affect the change(s) to the previous logic?**
   "null" inserts are therefore no longer possible.
3. **Background to the change (Why was this changed)?**
   The "boolean columns" should already have had all these constraints, but was forgotten here however at that time.

12.2. xRM Service

12.2.1. Unique and general views for knowledge articles

1. **What changes in detail?**
   With version 2023.0 it will be possible to count unique and general views of knowledge articles.
2. **How does it affect the change(s) to the previous logic?**
   Up to now, the views of knowledge entries in the NUMBEROFVIEWS column were incremented as simple integers. With version 2023. the column is moved to a new table UNIQUEVIEW. The outsourcing makes it possible to count unique (SELECT COUNT DISTINCT) and general (SELECT COUNT) views, since the CONTACT_ID of the user is stored each time the knowledge entry is called.

To migrate the existing view count, the process "migrate_views_serverProcess" must be executed once.

3. **Background to the change (Why was this changed)?**
   Both types of views should be counted. In addition, it opens the possibility to display the unique views in a filter view in individual projects, in order to be able to see which user opened the article and when.

12.2.2. Changed Datatypes

1. **What changes in detail?**
   In the tables DISTRICTCONTACT, DISTRICTCONTACTARCHIVE + VISITPLANENTRY the data type of some columns has changed from TIME to DATETIME:
   DISTRICTCONTACT:
   VISIT_BEGIN_TIME → VISIT_BEGINDATE
   VISIT_END_TIME → VISIT_ENDDATE
   (Time column + date from VISIT_BEGINDATE)
   DISTRICTCONTACTARCHIVE:
   VISIT_BEGIN_TIME → VISIT_BEGINDATE
   VISIT_END_TIME → VISIT_ENDDATE
   (Time column + date from VISIT_BEGINDATE)
   VISITPLANENTRY
   BEGIN_TIME → BEGINDATE
   END_TIME → ENDDATE
   (Time column + date from VISIT_DATE/STARTDATE - depending on whether the record is a visit proposal or a visit entry)
2. **How does it affect the change(s) to the previous logic?**
   All usages of the above columns must be adjusted; dates, on the other hand, are migrated by the respective Liquibase scripts by themselves:
   basic>2023.0.0>alter_visitplanAddBeginEndAsDatetimeAndMigrateData.xml
   basic>2023.0.0>alter_districtContactAddBeginEndAsDatetimeAndMigrateData.xml
   basic>2023.0.0>alter_districtContactArchiveAddBeginEndAsDatetimeAndMigrateData.xml
3. **Background to the change (Why was this changed)?**
   With pure TIME columns we can not apply the different time zones of the users in the system to display the times differently:
   If person A has a visit at 12 noon German time then a colleague from Chicago will also see this as 12 o’clock → this is confusing and contradicts the other times seen in the system.

12.2.3. Renaming of the Visitplanner

1. **What changes in detail?**
   Visitplanner → Operation Recommendation
   The functionality of the Operation Recommendation is still as known from the Visitplanner - but what has changed here is the functionality when it comes to create missions from a suggestion:
   -the actions have been replaced to create missions (both by serial action in the background and by preview action manually within the new component)
   -it is no longer possible to temporarily set From + To, instead this usecase can be mapped with the new action by creating a manual entry and simply specifying a different start/end time when creating a new entry.
2. **How does it affect the change(s) to the previous logic?**
   Functionality has changed - but from a technical point of view there is nothing to do
3. **Background to the change (Why was this changed)?**
   New and improved implementation with more features.

13. Update from 2022.1.X to 2022.2.0

13.1. xRM Contact Management

13.1.1. migrateCommunicationType_serverProcess

1. **What changes in detail?**
   The serverprocess migrateCommunicationType_serverProcess now migrates all communication type.
2. **How does it affect the change(s) to the previous logic?**
   In the xRM Update Manual 2021.2.X to 2022.0.0 a new serverprocess was added to migrate the existing communication records to have a standardized data structure. Due to the offpaging functionalities when updating large amount of records in custom projects there was a problem that only updated half of the records. Now it will update all records.
3. **Background to the change (Why was this changed)?**
   All records of the communication table will now be updated to the new standardized in order to keep data integrity.

13.2. xRM Sales

13.2.1. TOPICTREE normalization

1. **What changes in detail?**
   The TOPICTREE table and its data have been normalized.

2. **How does it affect the change(s) to the previous logic?**
   Until now there were 6 different columns to store the sortingValues, now there are only two: SORTINGLAYER and SORTINGVALUE.
   The migration of the data happens directly in the file where the new columns are added.
   Additional work is only required for projects that have added more levels than the 6 that were possible by default in xRM. In this case, a further update statement is necessary for each level.
   In all places where you have used the topic tree so far and have paid attention to the position and sorting so far, you now have to pay attention to the new structure. All functions in the TopicTree_lib have already been adapted and can be used for orientation.

3. **Background to the change (Why was this changed)?**
   With 2 columns normalized it allows us to create "unlimited number" of levels, there is no hard limit, extensibility is not a problem, performance is better as no more potentially huge caseWhens are needed to determine the current level, better readability of code.

13.2.2. APPOINTMENT_ID dropped from VISITPLANENTRY

1. **What changes in detail?**
   The APPOINTMENT_ID column was dropped from the VISITPLANENTRY table.

2. **How does it affect the change(s) to the previous logic?**
   The logic of the linked appointment of a visit proposal no longer goes via the APPOINTMENT_ID column but via the APPOINTMENTLINK record. All places in the basic have already been corrected to use the new logic (sync actions + display of the link).
   All non-basic places where the column was additionally used by customizing must now be adjusted accordingly and go the extra step via APPOINTMENTLINK.

3. **Background to the change (Why was this changed)?**
   By saving the APPOINTMENT_ID here, no calendar permissions could take effect and there were problems to determine when an appointment should be deleted and the reference thus clered.
   APPOINTMENTLINK on the other hand are deleted kernel-side when syncing the calendar

13.2.3. New territories table

1. **What changes in detail?**
   The assignment of companies to territories is now persisted (new table DISTRICTORGANISATION)

2. **How does it affect the change(s) to the previous logic?**
   The supervisor assignment now uses the persistent data records. In order for the assignment of maintainers to work, the company assignment must be triggered once. If you already execute the assignDistrict_serverProcess per interval you should consider in which rhythm you also want to execute the new company assignment process (assignOrganizationToDistrict_serverProcess) and set it accordingly.

3. **Background to the change (Why was this changed)?**
   Evaluating the filters to determine which company belongs to which area took a large part of the execution time of the process. This is despite the fact that there may be infrequent changes to company assignments depending on the project. By persisting now, the actual assignment of supervisors can be done much faster and per project, it can be considered how regularly the company assignment should be done.

13.2.4. New column PROGRESS in SALESPROJECT

1. **What changes in detail?**
   SALESPROJECT has got a new column PROGRESS. PROGRESS persistently stores how far this opportunity is with the checklist entries of the respective phase. To fill this initial "updateChecklistProgress_serverProcess" must be executed once (this calculates and sets the progress for all salesprojects in the system).
2. **How does it affect the change(s) to the previous logic?**
   It does not change anything existing, but allows to see at a glance how far which opportunity is in the respective phase. The value is persisted as said; at each check of the checklist + change phase it is automatically recalculated and set.
3. **Background to the change (Why was this changed)?**
   New feature to see at a glance how far an opportunity is in its phase with the checklist entries.

13.3. xRM Plattform

13.3.1. Version table

This Knowledge Contribution shows all currently active versions of ADITO.

14. Update from 2022.0.X to 2022.1.0

14.1. xRM Contact Management

14.1.1. OBJECT_TYPE and OBJECT_ROWID for ADDRESS

1. **What changes in detail?**
   The CONTACT_ID used to be stored in the addresses. In order to be able to save addresses for non-contacts as well, we have removed the column and replaced it with OBJECT_TYPE and OBJECT_ROWID. The existing data is already adjusted by a LiquibaseScript.
2. **How does it affect the change(s) to the previous logic?**
   Liquibase must be executed. And all non basic set adjust to OBJECT_TYPE and OBJECT_ROWID.
3. **Background to the change (Why was this changed)?**

So that also addresses can be deposited with e.g. building objects.

14.1.2. Observation_BG

1. **What changes in detail?**
   Changes are now collected in the Observation_BG table and processed using a server process that runs every few minutes.
2. **How does it affect the change(s) to the previous logic?**
   The server process must be set an interval by the administartor.
3. **Background to the change (Why was this changed)?**
   It has been moved to the background server due to performance

14.1.3. OBJECT_TYPE and OBJECT_ROWID for TASKS in TASKATTENDEES

1. **What changes in detail?**
   The CONTACT_ID and DEPARTMENT_ID used to be stored in the tasks. Changed to OBJECT_ROWID and OBJECT_TYPE in the TASKATTENDEES table.
2. **How does it affect the change(s) to the previous logic?**
   Liquibase must be executed. And all non basic set adjust to OBJECT_TYPE and OBJECT_ROWID
3. **Background to the change (Why was this changed)?**
   Continuity in the system and problems with the resolution of IDs.

14.2. xRM Sales

14.2.1. VISITRECOMMENDATION to VISITPLANENTRY

1. **What changes in detail?**
   The VISITRECOMMENDATION table is dropped and the data before it is migrated to VISITPLANENTRY.
2. **How does it affect the change(s) to the previous logic?**
   All calls and usages of the VISITRECOMMENDATION table must be adapted.
   Data is migrated automatically via this liquibase script, if structural changes have been made to VISITRECOMMENDATION via customizing, this changelog must be adapted if necessary.
3. **Background to the change (Why was this changed)?**

14.2.2. insert_ab_currency.xml in release folder

1. **What changes in detail?**
   The insert_ab_currency.xml changelog was previously in the demo data folder, but makes more sense in a normal release folder (same logic as with keywords).

2. **How does it affect the change(s) to the previous logic?**
   If you already have entries in the table, you may not want to run the script.
3. **Background to the change (Why was this changed)?**
   Makes more sense this way, since the data is relevant to system operation.
   Diff

15. Update from 2021.2.X to 2022.0.0

15.1. xRM Contact Management

15.1.1. Duplicate pattern

1. **What changes in detail?**
   Duplicate pattern (added by {technical: true})
2. **How does it affect the change(s) to the previous logic?**
   When opening a person, the duplicates are not loaded immediately, although nothing has been entered yet.
3. **Background to the change (Why was this changed)?**
   Performance when opening the person_editview.

15.1.2. Renaming OrgUtils.getDSGVOAnonymisierungOrganisationId

1. **What changes in detail?**
   Renamed the function OrgUtils.getDSGVOAnonymisierungOrganisationId to OrgUtils.getDSGVOAnonymizeOrganisationId.
2. **Background to the change (Why was this changed)?**
   The function name does not comply with the code conventions (AID001).

15.1.3. Communication OBJECT_TYPE

1. **What changes in detail?**
   Communication data had OBJECT_TYPE 'Contact' for company and person. Now the company has 'Organisation' and the persons addresses 'Person' as type.
2. **How does it affect the change(s) to the previous logic?**
   Run migrateCommunicationType_serverProcess. And adjust all the non-basic settings to the two types.
3. **Background to the change (Why was this changed)?**
   System continuity and distinction between companies and individuals Addresses.

15.1.4. Groupable subquery fields

1. **What changes in detail?**
   Groupable fields with subqueries were included in the FromClause of the record container.
2. **How does it affect the change(s) to the previous logic?**
   The fields can be grouped again.
3. **How do I test the change(s) as a user?**
   Group by a field if a Derby/Oracle/MSSQL DB is connected in the background.
4. **Background to the change (Why was this changed)?**
   Fields whose display value is determined by subqueries cannot be grouped in Derby/Oracle/MSSQL.

15.2. xRM Sales

15.2.1. Multi-currency capability

⚠ Major Change!

1. **What changes in detail?**
   The multi-currency capability has been implemented, and immediately implemented in all sales contexts:
   Offer (Offer_entity & Offeritem_entity), Document (Order_entity & Orderitem_entity), Salesproject (Salesproject_entity, Forecast_entity), Advertising_entity, Advertisingitem_entity, Price(Productprice).
2. **How does it affect the change(s) to the previous logic?**
   Before anything else is done, the local currency must be stored in the project preferences. String property "local.currency" → simply write the ISO4 of the currency ("EUR").
   Check the content within the entitys documentation property for further information!
3. **Background to the change (Why was this changed)?**
   Check the content within the entitys documentation property for further information!

15.2.2. DISCOUNTED_NET and DISCOUNTED_VAT dropped

1. **What changes in detail?**
   The columns DISCOUNTED_NET and DISCOUNTED_VAT are dropped from OFFER and SALESORDER.
2. **How does it affect the change(s) to the previous logic?**
   All calls and uses of the columns must be adjusted.
   Value is calculated as follows: SALESORDER.NET - SALESORDER.NET * SALESORDER.DISCOUNT /

100.

3. **Background to the change (Why was this changed)?**
   It did not make sense to cache these values in the first place.

15.2.3. Default Values and Constraints for OFFER/SALESORDER/SALESPROJECT

1. **What changes in detail?**
   The columns DISCOUNT at OFFERITEM and SALESORDERITEM got a defaultValue (0) and a notnullconstraint. (in this turn, all items without discount get a discount of 0).
   The same goes for PROBABILITY of OFFER/SALESORDER/SALESPROJECT.
2. **How does it affect the change(s) to the previous logic?**
   All quotes and voucher items now always start with 0 discount.
   All quotes, documents and sales projects now always start with 0 probability.

15.2.4. ADVERTISINGITEM changes

1. **What changes in detail?**
   Default constraint on SENT of ADVERTISINGITEM.

15.3. xRM Marketing

15.3.1. Advertising settings

1. **What changes in detail?**
   The advertising settings have been revised. An advertising permission or ban now always refers to a specific address and is no longer possible globally. In addition, the COMMRESTRICTION table, which is no longer used, was removed from the xrm.
2. **How does it affect the change(s) to the previous logic?**
   The existing global locks/permits are updated to individual locks/permits on all channels via a Liquibase script.
3. **Background to the change (Why was this changed)?**
   In the current implementation, complete traceability is not given. The communication address of a consent, for example, is not stored.

15.4. xRM Plattform

15.4.1. New local development database

For local use only!

1. **What changes in detail?**
   Version 2022.0.0 includes a new script for MariaDB that can be used for local development. This script runs over NodeJS and downloads a MariaDB locally (currently only for Windows), creates the required tables and runs Liquibase. The user just has to right click on the file "package.json" in the project tree and run it under "NodeJS" → "npm install". For this the plugin of NodeJS is needed, which must be completely installed. Afterwards the script "StartMariaDB" can be executed via selection in the ADITO Designer. The required serverconfig.xml is also supplied/generated by the script.
2. **Background to the change (Why was this changed)?**
   So that in the future you can directly start a local development system without having to make further changes.

16. Update from 2021.1.X to 2021.2.0

16.1. xRM Contact Management

16.1.1. Duplicatescanner

1. **What changes in detail?**
   The duplicate pattern has changed as well as the corresponding provider and consumer for

organisation and person.

2. **How does it affect the change(s) to the previous logic?**
   Every project that has changed the default value of the duplicatescanner has to manually adjust the change **in** their project (table "DUPLICATESCANNER", column "SCAN_PATTERN"). If you did not change any settings you can simply upgrade your project and the provided liquibase file will update your duplicatescanner. The provider "Dublicates" in organisation and person was changed to "Duplicates". If you implemented the dupblicate scanner in other entities you need to adjust the name there as well manually. The consumer "Duplicates" has to be changed to "DuplicatesPerson" and "DuplicatesOrganisation" in their respective entity.
3. **Background to the change (Why was this changed)?**
   The typo had to be fixed.

16.1.2. Contact synchronization (EWS)

1. **What changes in detail?**
   The former logic to add contacts to the contact synchronization (EWS) got refactored in order to be more efficient and suitable for mass data.
2. **How does it affect the change(s) to the previous logic?**
   A new dashboard and context (see "User Documentation" / userhelp of the context "Contact synchronisation") let’s you add filter in which the resulting data sets will be synchronized. If you are using the database Postgres you need to add a seperate modul in your database called "uuid-ossp" in order to use this function. The former logic will be removed with the next major release (2022.0.0)!
3. **How do I test the change(s) as a user?**
   See document "User Documentation" or the userhelp of the context "Contact synchronisation".
4. **Background to the change (Why was this changed)?**
   Suitable for mass data and be dynamic via the filters.

16.1.3. Export templates

1. **What changes in detail?**
   The placeholder logic for export templates was removed and replaced with the entity field logic. You now use entity fields to use as a mapping in your exports.
2. **How does it affect the change(s) to the previous logic?**
   Every export template has to be manually adjusted to the new logic. You cannot use placeholders anymore! All your existing export templates will remain and be visible but cannot be used after the update before you manually corrected them.
3. **How do I test the change(s) as a user?**

The administrator has a new tab to use the entity field logic and maintain the fields (see "Releaseletter 2021.2").

4. **Background to the change (Why was this changed)?**
   So you now don’t have to involve a contact in order to export.

16.1.4. Anonymization organisation

1. **What changes in detail?**
   The liquibase file changed to insert anonymized organisations to another data set.
2. **How does it affect the change(s) to the previous logic?**
   Anonymized organisations will no longer be stored in "meine Firma".

16.2. xRM Sales

16.2.1. Districts were expanded by visit frequency

1. **What changes in detail?**
   The visit frequency is now stored with the contacts of the area and taken from the visitPlanrecommendation_entity accordingly.
2. **How does it affect the change(s) to the previous logic?**
   Previously, the automatic visit suggestions were generated based on the visit frequency that was stored for the companies. Now, the visit frequency is entered via the areas in the corresponding caregiver records.
3. **How do I test the change(s) as a user?**
   Add a responsible person to a territory, add this person as a supervisor for a company (optionally also automatically by the assign area action). → if the supervisor record has a visit frequency set, a new suggestion appears at the visit suggestions.
4. **Background to the change (Why was this changed)?**
   To fully utilize the district functionality.

16.2.2. Datatypes changes to default columns

1. **What changes in detail?**
   The datatype of the columns have been changed (Table: Column → New Datatype, …):
   - CHECKLIST: DATE_NEW → DATETIME, DATE_EDIT → DATETIME
   - DISTRICTRESPONSIBLE: DATE_NEW → DATETIME, DATE_EDIT → DATETIME
   - KNOWLEDGETAGLINK: VALID_FROM → DATETIME, VALID_UNTIL → DATETIME
   - SALESPROJECTPHASEDEFINITION: DATE_NEW → DATETIME, DATE_EDIT → DATETIME

2. **How does it affect the change(s) to the previous logic?**
   If these columns have already been accessed outside the xRM project via sql, it would be advisable to look at these places again, since the change of the data types can lead to errors, especially in the date columns.

3. **Background to the change (Why was this changed)?**
   Consistency throughout the xRM project.

16.2.3. Datatype change to PROBABILITY

1. **What changes in detail?**
   The data type of the PROBABILITY column of the OFFER column has been changed from CHAR to INT.

2. **How does it affect the change(s) to the previous logic?**
   If these columns have already been accessed outside of the xRM project via sql, it would be advisable to look at these places again, since the change of the data types can lead to errors.

3. **Background to the change (Why was this changed)?**
   Consistency throughout the xRM project + better filtering options in the client (greater than, less than, etc…).

16.2.4. Topic Tree

1. **What changes in detail?**
   The topic tree is back and it is possible to migrate the topic tree to the new structure thanks to the new "migrate_themesAsTopics_serverProcess" (as long as the old structure was the same as in the old xRM project!)

2. **How does it affect the change(s) to the previous logic?**
   The status is not adjusted. CAUTION! There are only 6 layers possible and not infinitely many as before! if more exist either the tree must be adapted or the number of possible layers must be expanded. (If a topic is active, all subtopics are also active, if a topic is inactive, all subtopics are inactive → data is not falsified, but can be easily adjusted after the import logically in the administration).

3. **Background to the change (Why was this changed)?**
   To support an easy way to migrate existing topic trees into the new structure.

16.2.5. Microsoft Teams integration

1. **What changes in detail?**
   The Microsoft Teams integration is now detached from the sales project and modularly usable everywhere (Object_type, Object_rowid).

2. **How does it affect the change(s) to the previous logic?**
   If you already used the Microsoft Teams integration you may need to adapt to the newcomming changes
3. **Background to the change (Why was this changed)?**
   Consistency throughout the xRM project.

16.3. xRM Plattform

16.3.1. $local.idvaluesExcluded

1. **What changes in detail?**
   The variable "$local.idvaluesExcluded" is now also observed in the basic library for building the RecordContainerCache keys (CachedRecordContainer_lib).
2. **How does it affect the change(s) to the previous logic?**
   If you don’t use this library you might have to add it. It may be necessary to exclude entries that would be returned by the RecordContainer. For this purpose the variable $local.idvaluesExcluded is used which contains the excluded UIDs. This must of course also be taken into account in the RecordContainerCache, since it affects the delivered records that are contained there.
3. **Background to the change (Why was this changed)?**
   With version 2021.2.0 comes the so called RecordsRecipe (view the Update Manual for more information).

-