Integrated Accounting Tools

Documentation & User manual

Content

[Overview 2](#_Toc100676395)

[Requirements 2](#_Toc100676396)

[Installation and use 2](#_Toc100676397)

[Directory and file structure 10](#_Toc100676398)

[Application design and workflow 10](#_Toc100676399)

[Testing and error handling 12](#_Toc100676400)

[Revision 12](#_Toc100676401)

# Overview

The “Integrated Accounting Tools” represents a VBA-Based Excel client application, created to automate diverse set of simple, but time-consuming accounting activities performed manually by QtC team in SAP transactions. As of version 2.2.20210923, the following activities are automated:

* mass attachment messaging (1 file per message) to a specific address using MS Outlook email client,
* open items transfer from customer account to customer/GL accounts using F-30 transaction,
* file upload to DMS using UDM\_DISPUTE transaction,
* uploading of credit management documents to customer accounts using FD32 transaction,
* creating a list of accounting document attachments using FB03 transaction,
* retrieving international location numbers (ILN) for customer accounts using XD03 transaction,
* determining the type and fiscal year of credit notes (external/internal) using VA03 transaction.

The application exposes an excel form-based graphical user interface (further referenced as GUI) to interact with the user during runtime. Application input data is entered by the user on an operation-specific built-in sheet containing data table created for that purpose. The operation output data (where applicable) is then recorded in the appropriate field(s) of the table.

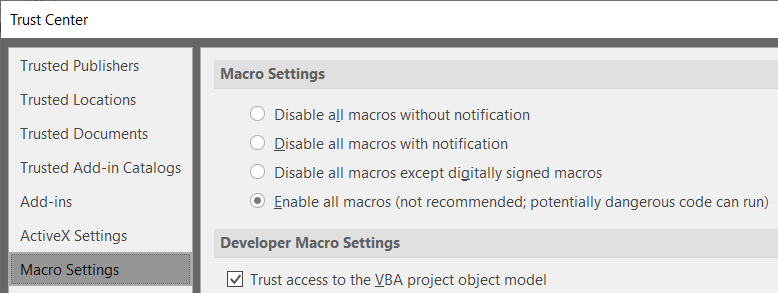
# Requirements

The application requires SAP GUI 750, MS Excel 2016 and MS Outlook 2016 installed on the client computer. Access to SAP P25 system and the appropriate user roles are mandatory for successful data processing in the SAP transactions.

# Installation and use

Copy the application folder to a location of your choice. Double-click the file: “IAT.xlsb” to open the application-containing workbook. Make sure macros are enabled on your computer before launching the application GUI (Navigate to: “File” -> “Options” -> “Trust Center” -> “Trust Center Settings…” and tick all the options as per the Pic. 1)

**Pic. 1:** Excel Trust Center settings required to run the application.

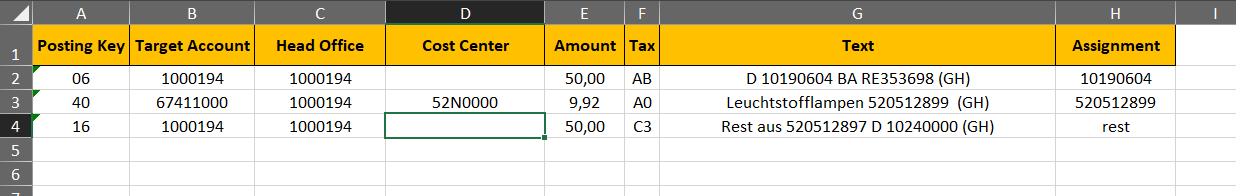


1. **Data input**

The user is mandated to fill in all the required fields with valid data. ***Warning***: Entering an invalid value will result in skipping the entire record during processing or an unexpected program behavior!

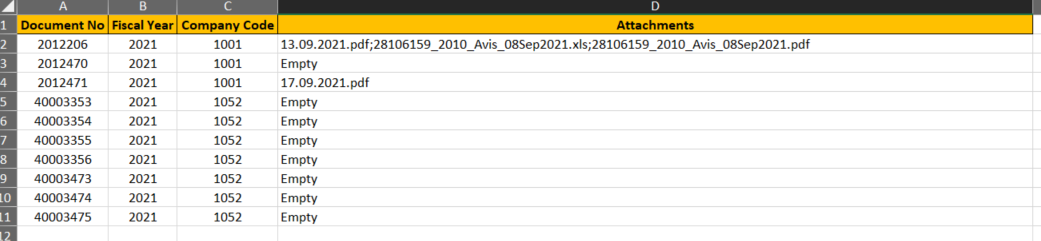
**1.1 Sheet F30 data fields**

* **Posting Key**: Defines the way items get posted to the target account. A list of available posting keys is displayed by clicking the drop-down list (chosen by user).
* **Target Account**: Destination account to which items are transferred (entered by user).
* **Head Office**: Head office account to which the item is assigned, regardless of customer account location (entered by user). Valid only for GL accounts requiring a profit center for item transfer.
* **Cost Center**: Cost center used for item transfer (entered by user).
* **Amount**: Amount in document currency to transfer, must be always a positive value (entered by user).
* **Tax**: Tax code applied to the item transferred (entered by user).
* **Text**: Description text applied to the item transferred (entered by user).
* **Assignment**: A unique item identifier applied to the item transferred (entered by user).

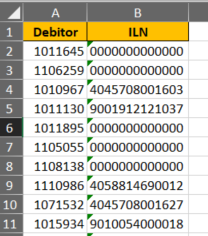


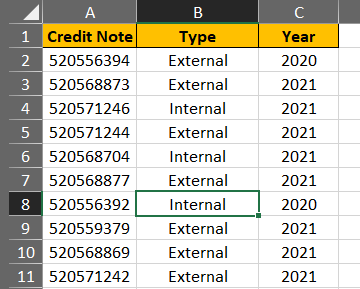
**1.2 Sheet FB03data fields**

* **Document No:** The number of the document generated upon payment booking (entered by user).
* **Fiscal Year:** Represents the document posting fiscal year (entered by user).
* **Company Code:** Unique country accounting identifier (entered by user).
* **Attachments:** List of files attached to the document (program output)



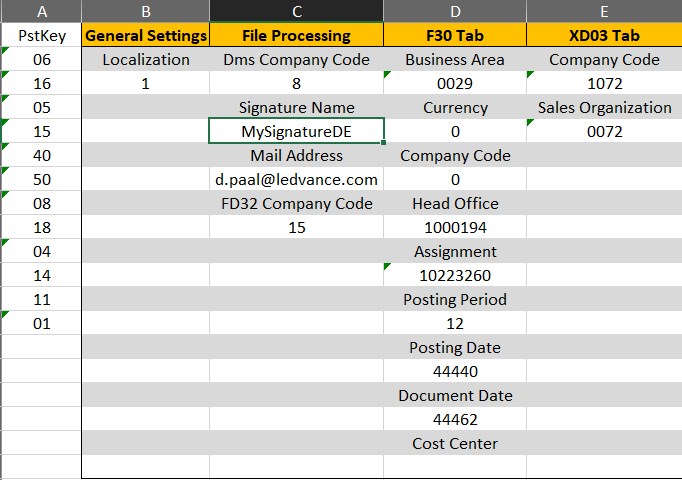
**1.3 Sheet XD03 data fields**

* **Debitor:** SAPcustomer account for which the ILN will be retrieved (entered by user).
* **ILN:** International location number of the customer/subsidiary (program output).

**1.4 Sheet VA03 data fields**

* **Credit Note:** Credit note number to check (entered by user).
* **Type:** Type (external/internal) of the credit note (program output).
* **Year:** Fiscal year in which the document was created (program output)

**1.5 Sheet Settings data fields**

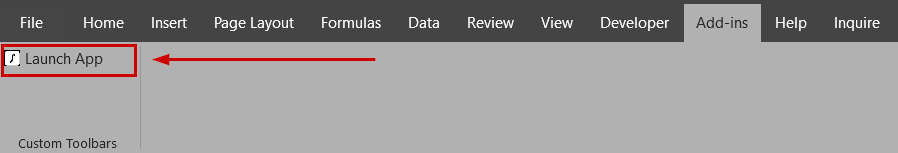
For developing purpose only! Normally, this sheet is hidden in the production version. Stores all user-entered values and settings in GUI fields and drop-down menus.

* **PstKey**: posting keys used in “F30” sheet data table
* **General settings**: stores general application settings
* **File Processing**: stores settings for DMS, FD32 and mailing operations performed on the “Files” tab of GUI
* **F30 Tab**: stores settings for the “F30” tab of GUI
* **XD3 Tab**: stores settings for the “XD03” tab of GUI

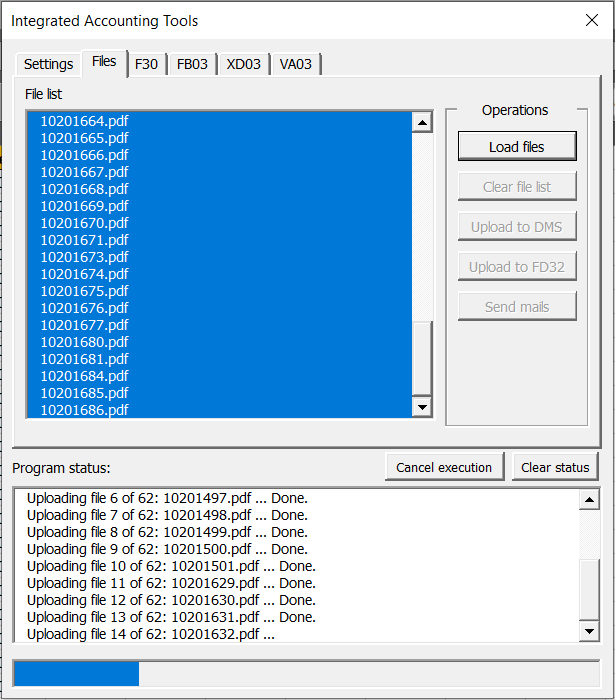
1. **Data processing**

Upon entering the data, user needs to open the application graphical user interface (further referenced as GUI). To start the GUI, navigate to the tab “Add-ins” and clicks the “Launch App” button.

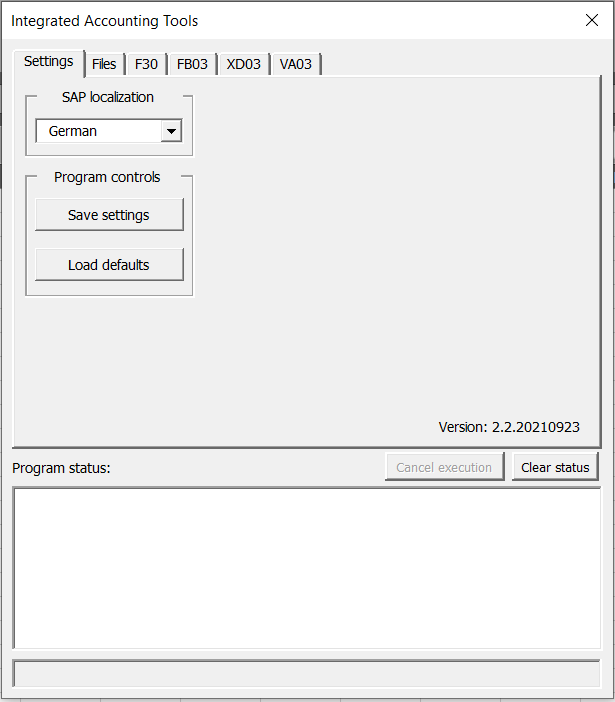
**Pic. 2:** Launching of the application GUI



An excel-form based GUI will open. The user interface contains application controls organized into tabs, each of which is designed to perform process-specific tasks, except for the “Settings” tab serving solely to configure application functioning during runtime. All user-relevant program events occurred at runtime are recorded in the “Program status” window in the lower part of the GUI. In case of any error, this is printed with a detailed description, allowing the user to have a full control over the processed data and settings. The progress of data processing is displayed via a progress bar placed below the “Program status” window. The purpose of the “Clear status” button is clearing the status window content and resetting the progress bar. Any running operation can be cancelled by clicking the “Cancel operation” button. All GUI text fields support value checking, that means, if a potentially incorrect value is entered, the field changes its color to red and an alerting message box appears.

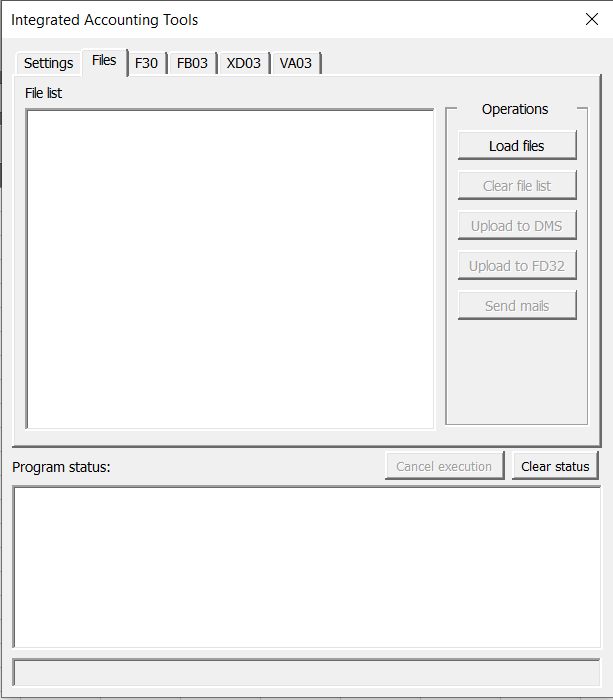
**Pic. 3**: Example of a running application with the progress indicated by the blue stripe at the bottom of the window.

**2.1 The Settings tab**

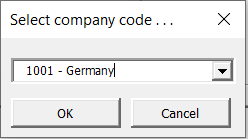
Serves as the main point for general application-runtime configuration.

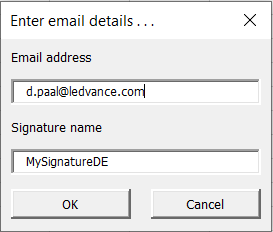
* “*SAP localization*” drop-down: contains all supported languages used in the company’s SAP systems. It is highly recommended to choose the language identical with the user’s SAP GUI localization.
  + *“Save settings”* button: Saves current user settings for later application use.
  + “*Load defaults*” button: Loads default application settings. These settings are not saved automatically. If you wish to save the default settings, you need to do so by clicking the “*Save settings*” button once the defaults are applied.

**2.2 The Files tab**

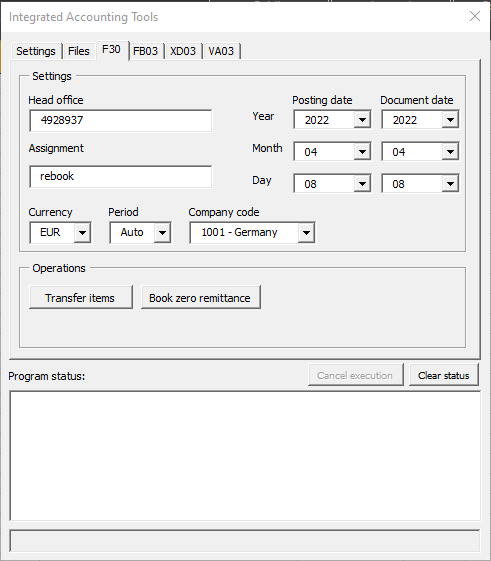
All file-related actions are available in the “Operation” side menu. First, the files to process need to be loaded and then selected from the “File list” prior to performing any action.

* “*Load files*” button: Opens file selection dialog window where user can pick files to process. A list of files is then created in the “*File list*” window located to the left of the “Operation” menu.
* “*Clear list*” button: Clears the content of the “File list” window.
* “*Upload to DMS*” button: Initiates uploading of the selected files to DMS.
* “*Upload to FD32*” button: Initiates uploading of the selected files to FD32.

***Note:*** *Both the DMS and FD32 file uploads require a valid company code. By clicking on either of the buttons, user is prompted to select the company code from a drop-down list. The last used value is then automatically saved.*

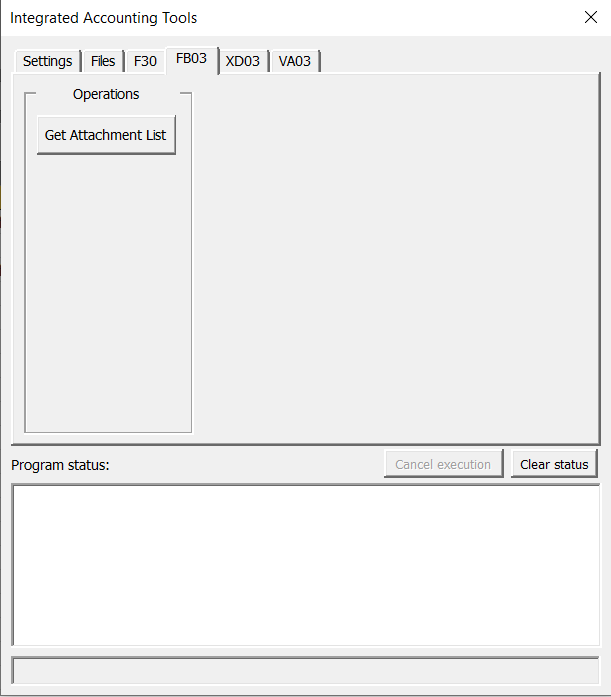
* “*Send mails*” button: Sends the selected files as attachments in separate (1 email : 1 file) outlook messages. Upon clocking the button, user is prompted to enter an email address (or confirm the existing one) to which messages with attached files will be sent and a Signature name (optional). The last used value is then automatically save for later use. ***Note***: *Omitting the signature name will result in messages being sent without any sender-identifying stamp*.

**2.3 The F30 tab**

Contains all the settings required to run a batch transfer of items according to the parameters entered in the “F30” sheet. Please, note that the final posting of transfers is not done automatically. Rather than that, it is up to the user to check the correctness of the operation output and then perform manual item posting.

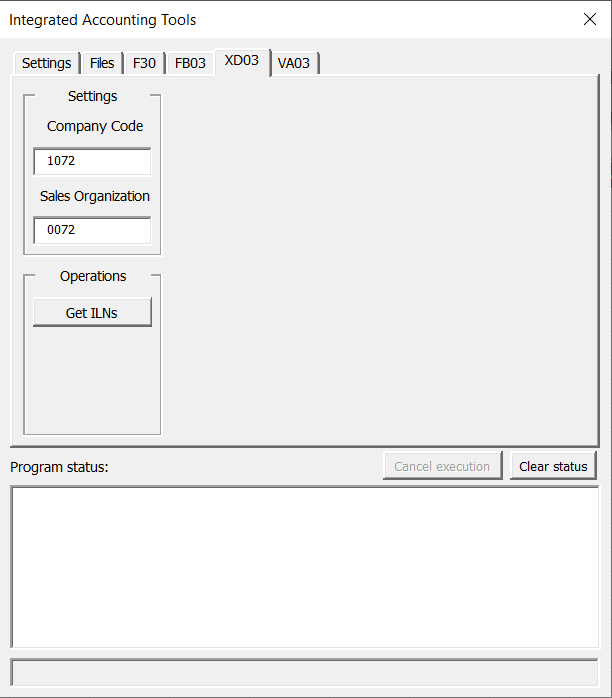
* “*Head Office*” text field: Customeraccount, where the payment bulk amount is posted.
* “*Assignment*” text field: A unique identifier of the bulk amount located on the head office account.
* “*Currency*” drop-down: Document currency used for item transfer.
* “*Period*” drop-down: Month in a fiscal year. If “Auto” option is used, the original value in F-30 transaction will be considered. Otherwise, the selected month will be applied.
* “*Posting date*” drop-down: The fiscal date on which the item transfer is performed. Note: Year, month and day must be selected with priority Year > Month > Day.
* “*Document date*” drop-down: The fiscal date on which the item document is created (may differ from the posting date). Note**:** Year, month and day must be selected with priority Year > Month > Day.
* *“Company code”* drop-down: Company code for which the transfer should be performed.
* “*Transfer items*” button: Click to transfer open items from customer accounts to destination (customer or GL) accounts according to data entered in the “F30” sheet. The user gets a notification of completion if the operation succeeds, otherwise an error message is returned.
* “*Book zero remittance*” button: Click to book a zero-amount remittance (payment) advice according to data entered in the “F30” sheet. The user gets a notification of completion if the operation succeeds, otherwise an error message is returned.

**2.4 The FB03 tab**

Automates repeated activities related to FB03 transaction.

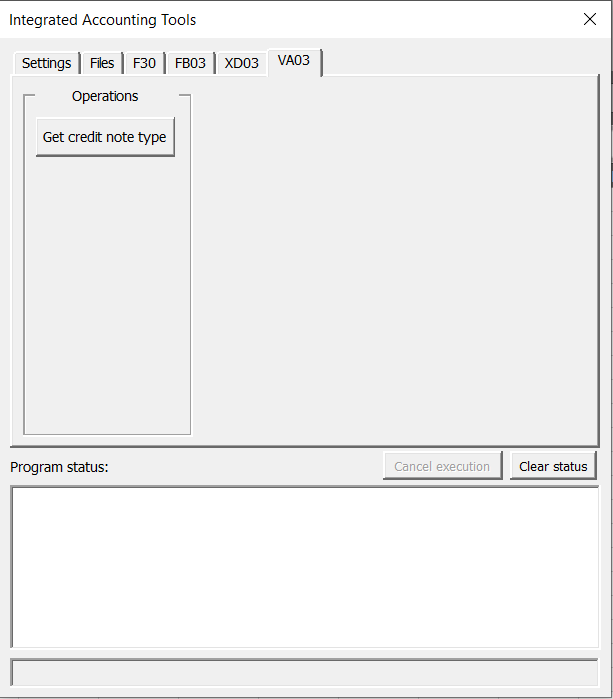
* “*Get attachment list*” button: Click to load data from the “FB03” sheet and initiate document processing in FB03 transaction. For each document, a list of attached files is created and written in the “Attachments” field of the table.

**2.5 The XD03 tab**

Automates repeated activities related to XD03 transaction.

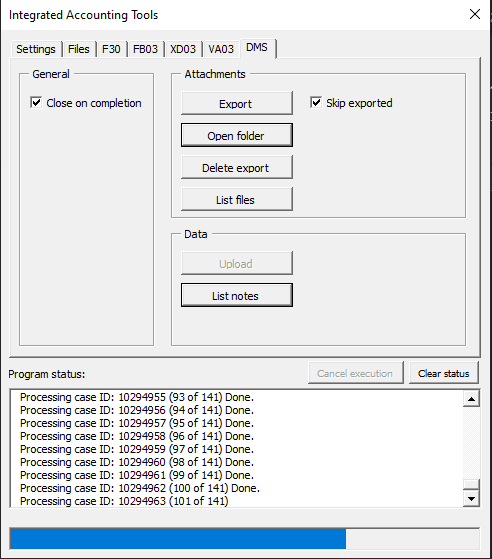
* “*Company code*” text field: Country identifier of the transferred items.
* “*Sales organization*” text field: Identifies the sales area for which the customer accounts were created.
* “*Get ILNs*” button: Click to load the data from the “XD03” sheet and initiate its processing in XD03 transaction. For each customer account, the corresponding ILN is then listed in the “ILN” field of the table.

**2.6 The VA03 tab**

Automates user activities related to VA03 transaction.

* “*Get credit note type*” button: Loads data from the “VA03” sheet and initiates document processing in VA03 transaction. The credit note type (external / internal) and fiscal year is listed for each document, which must be a valid credit note number.

**2.6 The DMS tab**

Automates user activities related to DMS transaction.

Section “General”

* “*Close in completion*” checkbox: Indicates whether the DMS should be closed upon completion of an operation.

Section “Attachments”

* “*Export*” button: Exports case attachments form DMS into a local folder. For each case, a subfolder named by the case ID is created, and all the files attached to that case are saved into that subfolder.
* “*Open folder*” button: Opens the folder with the exported attachments.
* “*Delete export*” button: Deletes the content of the parent export directory.
* “List files” button: Creates a list of attached files for the given cases.

Section “Data”

* “Upload” button: Uploads new values of case parameters to DMS.
* “Get notes” button: Reads case notes.

The MD\_DMS module

The module provides an interface to communicate with the UDM\_DISPUPTE transaction. Before using any of the processing procedures, the transaction must be started by calling the "Start" subroutine. Attempting to use any of the operation procedures before starting the transaction will cause the program to stop with an assertion. Once the desired operation is completed, the running transaction can be terminated by calling the “Quit” subroutine, which closes the transaction and clears the internal state of the module.

Sub Start(ByRef Sess As GuiSession)

Description:

Starts the UDM\_DSIPUTE transaction in a new session.

Parameters:

*Sess*: An SAP session in which the transaction will run.

Sub Quit()

Description:

Closes a running UDM\_DSIPUTE transaction.

Function ListAttachments(ByVal CaseID As Long) As Collection

Description:

Generates a list of files attached to a case.

Parameters:

*CaseID*: A DMS case ID number.

Returns:

File names stored as strings.

Function ExportAttachments(ByVal CaseID As Long, ByVal DirPath As String) As Collection

Description:

Exports files that are attached to a case.

Parameters:

*CaseID*: A DMS case ID number.

*DirPath*: Path to the directory to which the files will be exported.

Returns:

String names of the exported files.

Function GetCaseAttibutes(ByVal CaseID As Long, Names As Collection) As Object

Description:

Returns case parameters and their values.

Parameters:

*CaseID*: A DMS case ID number.

*Names*: Names of the case parameters. The following public constants of type **String** representing the case attributes may be used:

|  |  |
| --- | --- |
| **Constant literal** | **DMS case parameter** |
| ATTRIB\_CASE\_ID | Case ID |
| ATTRIB\_FAX\_NUMBER\_COUNTRY | Ctry of Fax No |
| ATTRIB\_CLOSED\_ON | Closed On |
| ATTRIB\_SOLVED\_ON | Solved On |
| ATTRIB\_TELEPHONE\_NUMBER | Telephone No. |
| ATTRIB\_CHANGED\_ON | Changed On |
| ATTRIB\_CREATED\_ON | Created On |
| ATTRIB\_ESCALATION\_DATE | Escalation date |
| ATTRIB\_ESCALATION\_LEVEL | Escalationlevel |
| ATTRIB\_PLANNED\_CLOSE\_DATE | Planned Close Date |
| ATTRIB\_PERSON\_RESPONSIBLE | Person Responsible |
| ATTRIB\_CUSTOMER\_CONTACT\_DATE | Cust.cont.dat |
| ATTRIB\_CASE\_TYPE | Case Type |
| ATTRIB\_CUSTOMER | Customer |
| ATTRIB\_EMAIL | e-mail |
| ATTRIB\_CLOSED\_BY | Closed By |
| ATTRIB\_SOLVED\_BY | Solved By |
| ATTRIB\_CHANGED\_BY | Changed By |
| ATTRIB\_CREATED\_BY | Created By |
| ATTRIB\_HEAD\_OFFICE | Head office |
| ATTRIB\_CURRENCY | Currency |
| ATTRIB\_COMPANY\_CODE | Company Code |
| ATTRIB\_ROOT\_CAUSE\_CODE | Root Cause Code |
| ATTRIB\_STATUS\_AC | Status AC |
| ATTRIB\_STATUS\_SALES | Status Sales |
| ATTRIB\_STATUS | Status |
| ATTRIB\_CATEGORY | Category |
| ATTRIB\_REASON | Reason |
| ATTRIB\_EXTERNAL\_REFERENCE | External refer. |
| ATTRIB\_TITLE | Title |
| ATTRIB\_ASSIGNMENT | assignement |
| ATTRIB\_AMOUNT | Cust.-Disputed |
| ATTRIB\_CONTACT\_PERSON | Contact Person |
| ATTRIB\_BRANCH | Branch |
| ATTRIB\_PROCESSOR | Processor |
| ATTRIB\_COORDINATOR | Coordinator |
| ATTRIB\_FAX\_NUMBER | Fax Number |
| ATTRIB\_NOTES | Text stored in the case “Notes” |
| ATTRIB\_NOTIFICATION | Notification |

Returns:

A dictionary of case parameter names and the associated values stored as strings.

Sub ModifyCaseAttributes(ByVal CaseID As Long, ByRef Params As Object)

Description:

Updates case parameter values.

Parameters:

*CaseID*: A DMS case ID number.

*Params*: Parameter names and their values.

Invalid parameter names will be ignored.

Raises:

*Error -1*: If a case cannot be open.

*Error -2*: If a case cannot be edited.

*Error -3*: If a parameter has a valid name but an invalid value.

Sub AttachFile(ByVal CaseID As Long, ByVal Path As String)

Description:

Attaches a file to a case.

Parameters:

*CaseID*: A DMS case ID number.

*Path*: Path to the file to be attached.

# Directory and file structure

**Tab. 1:** Files contained in the “app” folder:

|  |  |
| --- | --- |
| **Name** | **Description** |
| IAT.xlsb | Main application execution file |
| Icon.ico | Application icon file in the “ico” format |
| Icon.jpg | Application icon file in the “jpg” format |
| log.txt | Application events are written to this file during program execution |

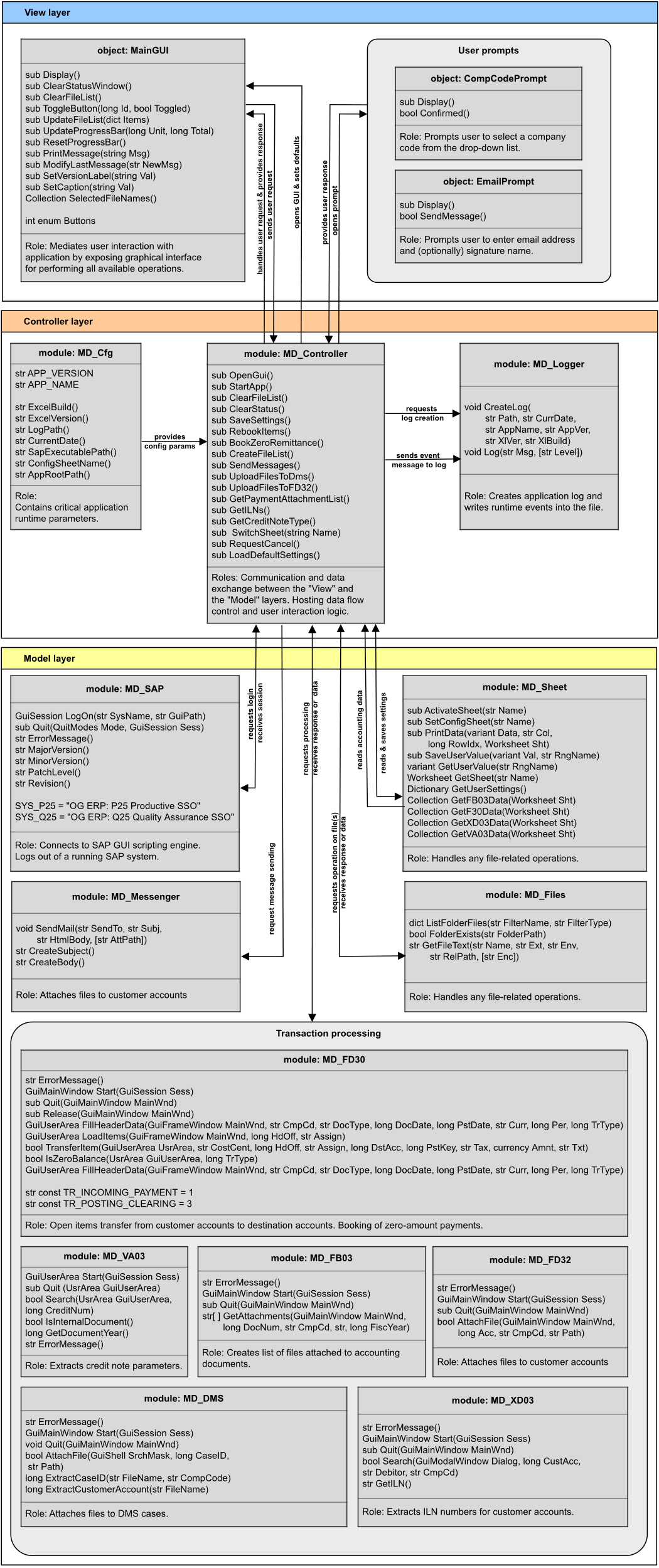
# Application design and workflow

The application was created using a customized procedural Model-View-Controller (MVC) design pattern to attain flexibility, extendibility, and easy debugging of the code (Pic. 4). The design was modelled using “UMLet 14.3.0“, and “Visual Studio Code UMLet” extension. A detailed design is available by opening [this link](file:///C:\Users\D.Paal\AppData\Roaming\Microsoft\Word\detailed_design.uxf). All modules and their roles are listed in Tab. 2.

The “View” layer consists of excel-based form objects facilitating user interaction with the application. The “Model” layer is responsible for data and file management and processing in SAP transactions or MS Outlook, respectively. Data processing and user interaction logic are contained in the “Controller” layer that bridges the “View” with the “Model” layer. This layer requires initialization by calling the “MD\_Controller.StartApp()” procedure, that formats and configures the “View” layer with user settings, and sets variables needed for its interaction with the “Model” layer. All runtime events are recorded explicitly at the “Controller” layer by passing the event description to the “MD\_Logger.Log()” procedure. The text is then written in application log file, serving as the primary reference point for error handling.

**Tab. 2:** Application modules and their roles

|  |  |
| --- | --- |
| Module/object name | Description |
| MD\_Controller | Manages data and control flow between Model and View |
| MD\_Cfg | Contains from-factory application configuration |
| MD\_Sheet | Manages data operations on application sheets |
| MD\_Files | Handles local file operations |
| MD\_SAP | Connects to SAP GUI and creates a new session |
| MD\_F30 | Automates item transfers from a bulk amount |
| MD\_VA03 | Automates document data retrieval for credit notes |
| MD\_DMS | Automates operations in the UDM\_Dispute transaction |
| MD\_FD32 | Automates file attaching to customer credit accounts |
| MD\_XD03 | Automates file information retrieval for customers |
| MD\_FB03 | Automates attachment list creation for accounting docs |
| MD\_Messeger | Automates messaging of files to a specific email address |
| MainGUI.frm | Primary window for user application interaction |
| CompCodePrompt.frm | Dialog prompting the user to enter company code |
| EmailPrompt.frm | Dialog prompting the user to enter email details |

**Pic. 4:** High-level overview of the application design with modules organized into three separate layers. The view layer (blue) mediates user interaction with the application. The model layer (yellow) maintains application data storage and processing logic. File handling, data and control flows are managed by the Controller layer (red). During its initiation, application configuration data is loaded from the “MD\_Cfg” module. All runtime events occurring at this layer are recorded using the “MD\_Logger” module.

# Testing and error handling

# Each module is extensively unit tested before it's integrated into the application. The application includes the "MD\_Tester" module for this purpose. A series of final integration tests are performed in close cooperation with end users to ensure program correctness and stability. All program runtime events are recorded in the log file. The data is kept in the log file until the next GUI launch of the application. The contents of the file are then deleted. If an error or unexpected behavior occurs, the developer handles it by reviewing the log file and/or the user error report.

# Revision

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 27.09.2021 | Dusan Paal | Initial version. |
| 1.1 | 11.04.2022 | Dusan Paal | Added booking of zero payment advices in F-30, as well as a few bug fixes. |
| 1.2.1 | 17.05.2023 | Dusan Paal | Added DMS data uploading, case attachment listing and export. |