

**Analysis Isabel connect6**

A logo with a black background

Description automatically generated

[info@tracked.be](mailto:info@tracked.be)  
Tel +32 456 764 87

# Contents

[Contents 1](#_Toc177551678)

[2 Intro 3](#_Toc177551679)

[3 Functional 3](#_Toc177551680)

[3.1 Payments 3](#_Toc177551681)

[3.2 Bank Statements 3](#_Toc177551682)

[4 Implementation 3](#_Toc177551683)

[4.1 Authentication 4](#_Toc177551684)

[4.2 Payment journal 4](#_Toc177551685)

[4.3 Bank Account 5](#_Toc177551686)

|  |  |  |  |
| --- | --- | --- | --- |
| **W**ho | **W**hen | Version | **Comment** |
| Bart Peirens | 18/09/2024 | 1 |  |
|  |  |  |  |
|  |  |  |  |

# Intro

The purpose of this analysis is to describe functionally and technically an application that can connect to Isabel6 connect and can be installed in Microsoft Dynamics 365 Business Central.

Isabel 6 connect is a cloud platform of the Isabel group that aims to simplify the multi-banking process of companies. This way you can have payments ready for multiple banks as well as withdraw the bank debits.

More info can be found [here](https://isabel.multibanking.eu/nl/).

This app should also be available in the Microsoft AppSource, but only with a contact button and no install button.

# Functional

## Payments

Within Isabel connect, payments can be prepared for various banks. These payments must then be checked and signed by a user with sufficient rights so that they can be processed.

The intention is that BC users will not have to download and upload SEPA payment documents for this system to work. There should be an action in which the amount to be paid can be sent directly to Isabel connect together with the bank information and structured or unstructured communication from the payment journal.

Users within BC should also have an overview where they can follow up on the status of these payments. The status must come from Isable connect and cannot be changed in BC.

## Bank Statements

The Isabel group also has a system to read bank statements via coda files. It is already possible to load files manually within BC Code. However, it is desirable to automatically retrieve these files via Isabel connect via the Bank card. We are not going to do this directly via then bank, which is also a solution, but via Isabel connect because in this application all the rights are set to read the files in the file. We prefer Isabel connect because only 1 system needs to be implemented.

We would also need a setting for each bank where we can set up an automatic poll to read and process the files on a recurring schedule.

# Implementation

It is important for the implementation that the functionality described in this document may change or additional functionality may be added in later versions. It is desirable to take this into account as much as possible when implementing so that later adjustments can be carried out smoothly.

The Isabel group has a well-documented API that you  [can find](https://documentation.ibanity.com/) here.  
It is therefore advisable to read these documents carefully before reading further.

This analysis is written in English, but all texts and labels must be translated into at least

* English
* Dutch
* French
* German
* Spanish
* Italian

It should be easy to add further translations.

## Authentication

An Isabel Connect configuration card should be created on which the user can fill in all the data needed to log in to the API. This configuration information must also be able to be created as a package to be easily transferred to another environment. Isabel provides a sandbox environment for development. Here we would prefer to use it in all sandbox environments within BC. Only if you are in a production environment within BC will you work within BC. This should be clearly visible on the configuration map.

Afbeelding met tekst, schermopname, nummer, Lettertype

Automatisch gegenereerde beschrijving

This mockup is an example of another application that also connects to Isabel6 and can be used to start but does not have to be 100% copied.

## Payment journal

Within the payment journal card, there should be a button between the "Check Payment Lines" and "Export Payment Lines" with the text "Send to Isabel".

When clicking, all payments that are visible will be forwarded. A unique key will also have to be created per line so that the status can be validated later.

Afbeelding met tekst, schermopname, lijn, nummer

Automatisch gegenereerde beschrijving

For the user, this functionality should be the same as an export of SEPA files that they import into Isabel. This export, import will be done automatically and the validation itself must be in order because they can do this in BC. What remains is to sign the SEPA files in Isabel to forward to the bank.

## Bank Account

On the bank account card, there should be extra action with the text "Import CODA Isabel" between "Code Statements" and "Import CODA File". If you click on this action, all CODA lines must be retrieved via Isabel CODA connect and imported to the selected bank. The ability to automatically reconcile should be possible when performing this action. For periodic retrieval, users must also have a configuration for this per bank.  
The booking itself will be done by a manual action after having checked it manually.

Afbeelding met tekst, schermopname, lijn

Automatisch gegenereerde beschrijving

For the automatic application of the statements, we are stuck with the limitations of BC, so we have no choice but to use the fixed rules. See [this page](https://learn.microsoft.com/en-us/dynamics365/business-central/localfunctionality/belgium/coda-bank-statements)

The current support for CODA bank statements can be used in Business Central online and on-premises. However, partners cannot extend the capabilities in an app for Business Central online. The code is not deprecated but can only be extended or customized for on-premises deployments.

Download CODA statements from Isabel

Automatically apply CODA Statements

YES

Notify users by Mail / Teams

Has all lines the status “applied”

NO

Automatically transfer and post CODA statements

Process CODA statements