Installation Guides

The steps you need to follow for implementing COBI.wms depend on the type of your SAP Business One installation. Follow the steps under the section that describes your environment.

On-premises or private cloud, with Service Layer

If you have an **on-premises or private cloud** installation of SAP Business One with **Service Layer available** (which means it is either based on SAP HANA, or the version of SAP Business One is 10.0 or newer) then follow these steps:

1. Create the **Management Database**
    1. Create the database itself by copy-pasting the correct creation script into *SQL Server Management Studio* or *HANA Studio*
    2. Add SAP Business One connections by using the appropriate `INSERT` statement from the Management Database
    3. Import licenses by using the appropriate `INSERT` statement from the Management Database documentationdocumentation
2. (For HANA: Install and start the **HANA Proxy**)
3. Make sure that the TCP ports for direct DB queries and Service Layer are not blocked
    1. MS SQL Server: 1433
    2. HANA Proxy: 30075
    3. Service Layer: 50000 (https) or 50001 (http)
4. Install COBI.wms from the Google Play Store on an Android device that can access the SAP Business One server
5. (optional) Use **cobiwms-config.html** to generate a Settings QR Code to scan with the app
6. Open COBI.wms on the Android device and scan the Settings QR Code or enter the settings manually
7. Click Connect and take note of the device ID on the right side of the screen below the Login button
8. Assign a license to that device ID (see License Management)
9. Log in to COBI.wms and make a test booking
10. (optional) Create COBI.wms users in the Management Database (see COBI.wms Users)
11. (optional) Disable all modules for all devices and enable them for users instead to force user login (see Permission Management)

(SBO 9.3 or earlier only) On-premises or private cloud, MS SQL Server

Warning: The use of Integration Framework with COBI.wms is deprecated and highly discouraged for new installations. Most new COBI.wms features are only available when using Service Layer.

If you have an **on-premises or private cloud** installation of SAP Business One **Version 9.3 or older**, based on Microsoft **SQL Server** then please refer to the following steps. Note that the

Integration Framework must be installed already.

1. Import the **Integration Framework Scenario Package**
    1. Afterwards, copy the **Trigger** URL as described in the linked page and temporarily save it somewhere
2. Follow all the steps of the previous section (on-prem with Service Layer), with the following differences:
    1. Use the Trigger URL for the APIURL field of the SAP Business One company connection added to the Management Database
    2. Make sure that TCP Port 8080 for Integration Framework is opened on the server

Public cloud

Prerequisites

If you are using SAP Business One from a public cloud (SAP-hosted or partner-hosted) then first of all, **your cloud provider must make Service Layer accessible**. Talk to your provider for details about this.

Additionally, to make all features of COBI.wms available and for better performance, the app needs to create a number of SQL Queries over Service Layer. This requires **SAP Business One Version 10 FP2011** or newer. Also, you either need a superuser to create the SQL Queries, or a user with the permission **Modify SQL Queries in Service Layer** which is only available starting with **FP2108**. Please see the section **Creating SQL Queries** below for more details.

Configuring the app

To configure the connection of the app to your SAP Business One cloud account, open **cobiwms-config.html** in a new tab of your browser, and follow these steps:

1. Choose **Companies (public cloud)** at the top of the page.
2. Click on **New** to start creating a new Company connection.
3. Enter the connection settings as instructed. (Fill at least Service Layer URL, Username, and Password.)
4. (To add additional company connections, like for a testing database, repeat steps 2 and 3.)
5. Click on **Generate QR Code** to display a QR Code containing the entered settings.
6. Start COBI.wms on your Android device and open the **Settings** screen if it doesn't open automatically.
7. Scan the QR Code on your monitor. (If your device's built-in scanner doesn't support QR Code, or you're using a regular Android device without a barcode scanner, then tap the **barcode icon** at the top right of the screen to open the device's camera for scanning. After the QR Code is recognized, tap on **Use** on the bottom right.)

Creating SQL Queries

When connecting to a cloud deployment of SAP Business One, COBI.wms cannot establish a direct database connection and instead has to request all data over Service Layer. To improve the

functionality and performance of these data requests, the app can create a number of SQL Queries while logging in to a company database.

This is done by checking **Update SQL Queries** in the login screen, below the login button, before logging in. (This will create the Queries if they were not created yet, and update them to the newest version if they were already created but not up-to-date, so it's recommended to do this once for every new version of COBI.wms.)

For this SQL Query creation/update to work correctly, the SAP user you've entered for the connection settings (see step 3 of the previous section) has to be either a superuser, or have the permission **Modify SQL Queries in Service Layer** under the “General” category. This permission is only available since **SAP Business One Version 10 FP2108**. In older versions, you must use a superuser.

All types of installation

There are some steps you might need to take which are independent of the SAP Business One installation:

- **DataWedge Settings**: If you have a Zebra scanner, see this page.
- **Barcode scanning via Bluetooth**: See here for special instructions regarding Bluetooth-based barcode scanners.
- **App Options**: COBI.wms has many options to customize the app for your needs.
- **Print Settings**: See here for connecting to ZPL-based label printers.
- **Label Templates**: See here for a guide on designing COBI.wms label templates.

FAQ

SQL Server Port Issue:

When connecting to the SQL Server, you may encounter issues that are often related to firewall restrictions or SQL Server configuration problems.

Explanation: The issue is typically due to one of the following reasons:

1. **Firewall Blockage:** The SQL port (commonly port 1433) might not be open between the COBI.wms device and the SAP Server.
2. **Server Configuration:** The SQL Server might be configured to disallow TCP/IP connections.

Steps to Diagnose and Resolve:

1. **Check Port Accessibility:**

1. From any Windows client on the same network as the COBI.wms device, open a PowerShell console.
2. Execute the following command (replace `<SAP_Server_IP>` with the actual IP address of your SAP Server)

PowerShell:

```
Test-NetConnection <SAP_Server_IP> -Port 1433
Expected Output:
The PowerShell output should show `TcpTestSucceeded : True`.
```

If it shows **False**, the port is likely blocked by the firewall.

2. **Verify SQL Server Configuration:**

1. If the port is accessible (i.e., `TcpTestSucceeded` is True) but the connection still fails, the SQL Server might be blocking TCP/IP connections.
2. On the server running the SAP MSSQL Server, open **SQL Server Configuration Manager**.
3. Navigate to **SQL Server Network Configuration → Protocols for [InstanceName]**

(Note: The instance name may correspond to your SAP system, e.g., “SAP”).

1. **Ensure that TCP/IP is enabled.**

3. **Using a Dynamic TCP/IP Port (if necessary):**

1. If problems persist, consider connecting via the dynamic TCP/IP port:
    1. In **SQL Server Configuration Manager**, double-click on **TCP/IP**.
    2. Go to the **IP Addresses** tab and locate the dynamic port setting.
    3. Enter the dynamic port number in COBI.wms under the “Optional Fields: DB-Portnummer” (DB Port Number).
    4. Make sure the dynamic port is also allowed through the firewall.

You can verify this by running the PowerShell command again, substituting port 1433 with the dynamic port number.

Customer Confirmation: One customer confirmed that switching to the dynamic port resolved their connection issues.