

# Produmex WMS Tools

When installing the Produmex Tools component with the Produmex Suite Installer, the system installs the following tools in C:\Program Files\Produmex\Produmex Tools.

## 1. Generate Documentation Tool

With the Generate Documentation Tool you can export the structure of Produmex related database tables, indexes and user defined fields into an html file. The structure of the current product version is available on the [main page](#).

## 2. HANA Migration Tool

With the HANA Migration Tool you can migrate a company database from MS SQL to HANA. For more information click [here](#).

## 3. HANA Report Setting Tool

On MSSQL, the connection data is replaced dynamically when printing and there is no need to save the report with the actual connection data.

On SAP HANA, the connection data is not replaced dynamically and it is necessary to save the selected reports with the actual connection to the right database and schema. With the HANA Report Setting Tool it is possible to modify the connection data for multiple reports. For more information click [here](#).

## 4. Import Initial SAP Stock Tool

On a database where Produmex WMS is installed for the first time, the Import Initial Stock Tool can be used to create Produmex stocks for all SAP stocks in selected warehouses. For more information click [here](#).

## 5. Import Tool

The Import Tool can be used to import elements through files to the Produmex system or to copy them directly from one database to another. For more information click [here](#) and [here](#).

## 6. Interfacing Tool

The Interfacing Tool is a part of the EDI module and with the tool certain documents can be imported to and exported from the system. For more information click [here](#).

## 7. Replenish Tool

Replenishment can be used within the same SAP Business One / PMX WMS warehouse and the aim of the function is to make sure that enough stock is available on pick locations to perform the picking process for a given period. When replenishment configurations are set for a location or an item, the Replenish Tool automatically creates replenish orders. For more information click [here](#).

## 8. Sales Delivery Tool

The Sales Delivery Tool creates SBO Sales Deliveries from PMX Sales Shipping documents. For more information click [here](#).

## 9. Robot Tool

**From 2024.10 version of Prodimex WMS the Robot Tool is capable to run custom developed C# Scripts that have been written by the users.**

**This function is opening up the possibility to create new actions for the [Notification Listener](#) where we can trigger a C# Script as an argument of the action, we can include a specific DocEntry so the action is triggered not for all the documents but only for that one identified by the DocEntry number.**

The **Robot Tool** is an executable tool that is started with command line parameters.

The command line parameters indicate the specific functionality that needs to be executed. If the `/RobotType` or `/t` parameter is not passed, the system defaults to `PickListRobot`.

### **Example 1**

If you pass command line `/?` all possible command line parameters will be displayed together with a description.

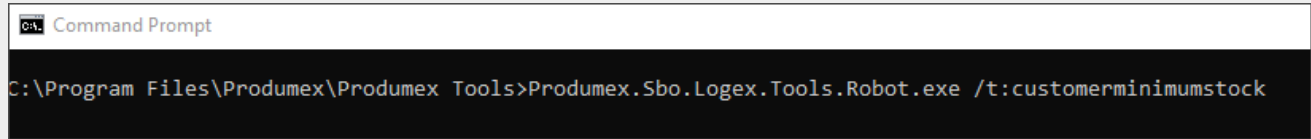
```
Prodimex.Sbo.Logex.Tools.Robot.exe /?
```



**Example 2**

To run the [Minimim Stock Level tool](#) you need to execute:

```
Produmex.Sbo.Logex.Tools.Robot.exe /t:customerminimumstock
```



```
Command Prompt
C:\Program Files\Produmex\Produmex Tools>Produmex.Sbo.Logex.Tools.Robot.exe /t:customerminimumstock
```

To run the [Minimim Stock Level tool](#) for a specific database you need to execute:

```
Produmex.Sbo.Logex.Tools.Robot.exe /t:customerminimumstock /cs:SboConnectionStringTest
```

**Example 3****Execute a C# script with the Robot Tool**

A sample C# script code (TestScript.cs):

```
using System;
using System.IO;
using System.Text;

namespace TestScript
{
    public class Program
    {
        static string CONNECTION_STRING = "it will be replaced by
robot";

        static void Main(string[] args)
        {
            StringBuilder sb = new StringBuilder();

            sb.Append("Start" + Environment.NewLine);
            sb.Append("CONNECTION_STRING: " + CONNECTION_STRING +
Environment.NewLine);

            if (args.Length > 0)
            {
                foreach (string arg in args)
                {
                    sb.Append("Argument: " + arg + Environment.NewLine);
                }
            }
        }
    }
}
```

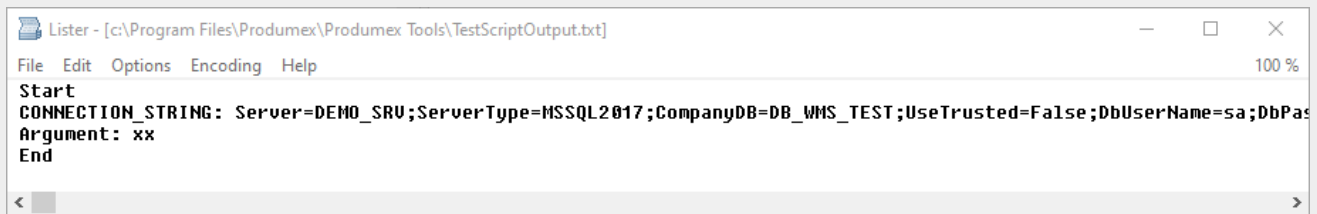
```
sb.Append("End");  
  
File.WriteAllText("TestScriptOutput.txt", sb.ToString());  
}  
}  
}
```

**The command to run the Robot Tool and execute the given script:**

```
c:\Program Files\Produmex\Produmex  
Tools>Produmex.Sbo.Logex.Tools.Robot.exe /t:csscript /a1:TestScript.cs  
/a2:xx /cs:WMSTEST
```

**The a1 argument** is the pathname of the cs-script file (mandatory).  
**The a2 argument** is passed to the script (optional).

**The result of the command:**



From:  
<https://wiki.produmex.name/> - **Produmex**

Permanent link:  
<https://wiki.produmex.name/doku.php?id=implementation:wms:tools>

Last update: **2024/10/07 08:04**

