



EVOLIS SDK

Evolis Services Provider API
Print a dual sided color design

Table of Contents

Abstract3

Prerequisite.....3

Technical limitations.....3

Testing thanks to the demo tool3

Set of requests to print a dual sided color design..... 4

Disclaimer 10

Abstract

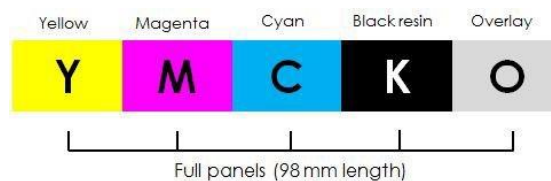
This documentation demonstrates how to print a dual sided color design thanks to Evolis Services Provider API

Prerequisite

One computer with an Evolis driver suite installed in standard or supervision mode (EvolisPremium Suite, Evolis premium suite 2, Edikio Printer Suite, Badgy Premium Suite).

Pay attention to name your printer (designed as "Device" in the request) as it is instantiated on the server (Ex : Evolis Zenius (Copy 1)).

This use case requires a genuine Evolis YMCKO ribbon and a duplex printer.



Technical limitations

Evolis Services Provider API is only available on Windows operating System for the server side.

Client must be compatible with JSON RPC V2.00 specifications.

Testing thanks to the demo tool

To test the requests set, possibly use the demo tool [here](#) (click on 1.demo.exe)

Select your communication mode in the first section. Copy / paste requests one by one sequentially (the expected results/answer shows up in the lower pane of the demo application).

Set of requests to print a dual sided color design

A print job requires a session. One session only is used per printing job, this to avoid jobs. A session must be opened then closed. A session has an automatic unique ID (here JOB000001) provided by the Evolis Services Provider or managed by the client application software.

Request 1: Start the sequence

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "CMD.SendCommand",
  "params": {
    "command": "Ss",
    "device": "Evolis Primacy",
    "timeout": "5000"
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```

Request 2: Begin print session

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.Begin",
  "params": {
    "device": "Evolis Primacy",
    "session": "JOB000001"
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "JOB000001",
  "id": "1"
}
```

Request 3: Set printer for current print session

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.Set",
  "params": {
    "data": "GRibbonType=RC_YMCKO;Duplex=HORIZONTAL;GDuplexType=DUPLEX_CC;IGSendSpoolerSession=OFF",
    "session": "JOB000001"
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```

Request 4 : Send front side color bitmap (= YMCKO), “data” is incomplete in this example.

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.SetBitmap",
  "params": {
    "session": "JOB000001",
    "face": "front",
    "panel": "color",
    "data": "base64:ColorBase64EncodedData..."
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```

Request 5: Send front monochrome black bitmap (= resin), “data” is incomplete in this example.

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.SetBitmap",
  "params": {
    "session": "JOB000001",
    "face": "front",
    "panel": "resin",
    "data": "base64:BlackBitmapBase64EncodedData..."
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```

Request 6: Send Front overlay (= varnish), “data” is incomplete in this example.

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.SetBitmap",
  "params": {
    "session": "JOB000001",
    "face": "front",
    "panel": "overlay",
    "data": "base64:VarnishBase64EncodedData..."
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```

Request 7 : Send back side color bitmap

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.SetBitmap",
  "params": {
    "session": "JOB000001",
    "face": "back",
    "panel": "color",
    "data": "base64:ColorBase64EncodedData..."
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```

Request 8: Send back monochrome black bitmap(= resin)

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.SetBitmap",
  "params": {
    "session": "JOB000001",
    "face": "back",
    "panel": "resin",
    "data": "base64:BlackBitmapBase64EncodedData..."
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```

Request 9: Send back overlay (= varnish)

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.SetBitmap",
  "params": {
    "session": "JOB000001",
    "face": "back",
    "panel": "overlay",
    "data": "base64:VarnishBase64EncodedData..."
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```

Request 10 : Run the print job (that has been buffered)

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.Print",
  "params": {
    "session": "JOB000001"
  }
}
```

Expected result :

```
{
  "jsonrpc": "2.0",
  "result": "OK",
  "id": "1"
}
```


Request 11 : End the print job

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.End",
  "params": {
    "session": "JOB000001"
  }
}
```

Expected result :

```
{
  "id": "1",
  "jsonrpc": "2.0",
  "result": "OK"
}
```

Disclaimer

While Evolis makes every effort to deliver high quality products, we do not guarantee that our products are free from defects. Our SDK, samples and demo software, any content or documentation delivered in this package (Evolis SDK) is provided "as is". The use of it is at your own risk.

Evolis makes no warranties as to performance, merchantability, fitness for a particular purpose, or any other warranties whether expressed or implied.

No oral or written communication from or information provided by Evolis shall create a warranty.

Under no circumstances shall Evolis be liable for direct, indirect, special, incidental, or consequential damages resulting from the use, misuse, or inability to use this Software Development Kit (named Evolis SDK), even if Evolis has been advised of the possibility of such damages.