



EVOLIS SDK

Evolis Services provider API
How-to manage errors

Contents

Abstract	2
Prerequisite.....	2
Technical limitations	2
Testing thanks to the demo tool	2
Focus on Print.print sequence	3
Typical sequence :	3
Set of requests to print a single side monochrome job with error management	4
THREAD 1 : single side monochrome black print job	4
Request 1: Start the sequence.....	4
Request 2: Begin print session.....	5
Request 3: Set printing parameters.....	5
Request 4: Send monochrome bitmap.....	6
Request 5 : Run the print job	6
Request 6: End the print job	7
THREAD 2 : Error management	7
No error	7
Error management.....	8
Disclaimer	9

Abstract

This documentation demonstrates **how-to manage errors while printing** a monochrome black design thanks to Evolis Services Provider API. The same way to manage errors could be implemented for any usecases.

For monitoring purpose or check printer's status outside of a print job, use the CMD service and the GetStatus command, see [evolIS_sdk_binary-status.pdf](#) in the [help_and_support](#) folder.

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)).

Pay attention than the error management must be managed in another thread than the print job (ex : thread 1 / print job, thread 2 / error management).

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.

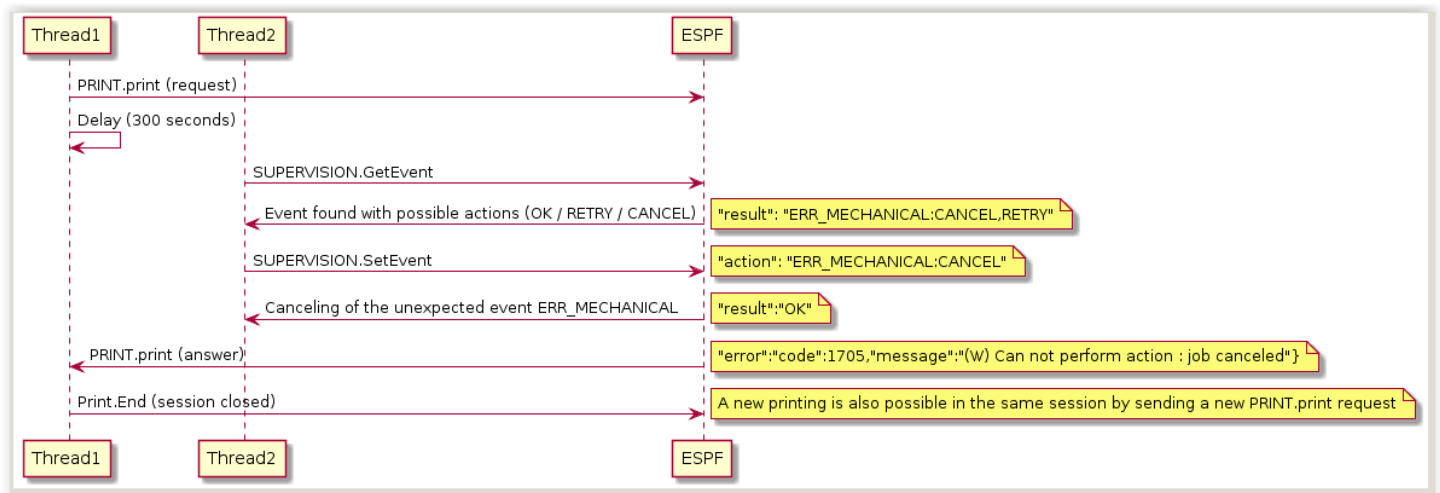
Testing thanks to the demo tool

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

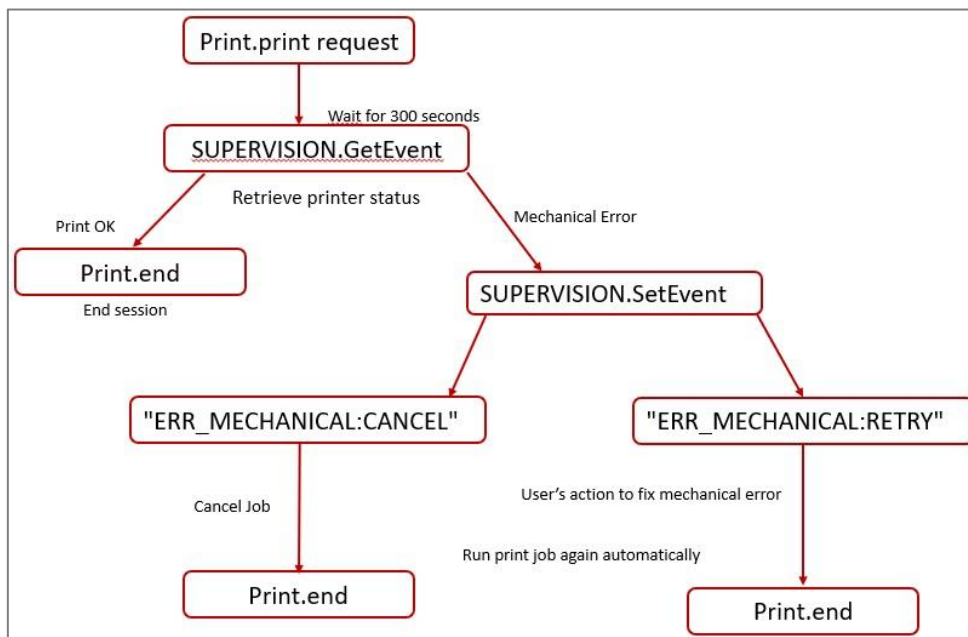
select your communication mode in the first section . Copy / paste requests one by one sequentially (the expected results/answer shows up in the lower pane of the demo application).

Focus on Print.print sequence

Typical sequence :



The error management (**SUPERVISION.GetEvent** and **SUPERVISION.SetEvent**) must be managed in **a separate thread** than the **PRINT.Print**.



Set of requests to print a single side monochrome job with error management

A print job requires one printing session. One session is used per printing job to avoid mixing jobs. A session **must** be opened then closed.

A session has an automatic unique ID (here JOB00001) provided by the Evolis Services provider or given (managed) by the client application software.

THREAD 1 : single side monochrome black print job

- A print job requires one printing session. Session system is used per printing job to avoid mixing jobs.
- A session **must** be opened and closed.
- A session has an automatic unique ID (here JOB00001) provided by the Evolis Services provider or given (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"
  }
}
```

Expected result :

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

Request 3: Set printing parameters

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.Set",
  "params": {
    "data": "GRibbonType=RM_KBLACK;Duplex=NONE;IGSendSpoolerSession=OFF",
    "session": "JOB000001"
  }
}
```

Expected result :

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

Request 4: Send monochrome bitmap

```
{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "PRINT.SetBitmap",
  "params": {
    "session": "JOB000001",
    "face": "front",
    "panel": "resin",
    "data": "base64:BlackBitmapEcodededinBase64 ..."
  }
}
```

Expected result:

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

Request 5: Run the print job

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

Expected result:

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

Request 6: End the print job

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

Expected result:

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

THREAD 2 : Error management

No error

```
{
  "id": "1",
  "jsonrpc": "2.0",
  "method": "SUPERVISION.GetEvent",
  "params": {
    "device": "Evolis Primacy"
  }
}
```

Expected result:

```
{
  "jsonrpc": "2.0",
  "result": "NONE",
  "id": "1"
}
```


Error management

```
{
  "id": "1",
  "jsonrpc": "2.0",
  "method": "SUPERVISION.GetEvent",
  "params": {
    "device": "Evolis Primacy"
  }
}
```

Expected result:

```
{
  "id": "1",
  "jsonrpc": "2.0",
  "result": "ERR_MECHANICAL:CANCEL,RETRY"
}
```

- Set an action (**CANCEL** OR **RETRY**)
 - **CANCEL** = Need to terminate current job via **SUPERVISION.Set** method and then **PRINT.End**
 - **RETRY** = Need to set the user's action to retry the job via **SUPERVISION.Set** method (will resume the current print job).

```
{
  "id": "1",
  "jsonrpc": "2.0",
  "method": "SUPERVISION.SetEvent",
  "params": {
    "action": "ERR_MECHANICAL:RETRY"
    "device": "Evolis Primacy"
  }
}
```

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.