Operating instructions for calling the programme Fiskalizuj.exe

(ver 18/05/2021)

Technicalities:

Installation: From fiskalizuj.online web page obtain archived Fiskalizuj.exe with necessary libraries (FiscalConnector.zip), unpack it in any directory and call the program Fiskalizuj.exe (or FiskalizujDll.dll if software is written in C#.NET) from your software, in accordance with the instructions below.

The TLS 1.0 protocol has been considered vulnerable for a long time now, so PU (Poreska Uprava = Tax Administration of Montenegro) implemented the TLS (Transport Layer Security) 1.2 cryptographic protocol, which provides a higher level of security of inter-network communication. That's why the system is built on .NET Framework ver. 4.5, which does **not support Windows XP and Server 2003.** As a result, fiscalization will not work if you are using devices with these operating systems.

For a successful operation, the following **port** must be open for outbound communication:
fiskalizuj.online:443

Proxy server:
If you need it, set the <add key = "ServerProxy" value = "VALID_IP_OR_HOST_NAME: PORT" /> in .config file. (tag in the .config file already exists, just set the appropriate blue tint value).
You can set value to "auto" for using default system proxy settings.

For possible operation **in a Linux environment or on Android**, please send an email to dusan@fiskalizuj.online in which you will describe your production environment, and we will try to ensure you access to an adequate distribution.

Program call:

FISKALIZUJ.EXE <DemandType> <Mode> F#<InvFile>|S#<InvString> [RF#<ResponseFile>] [QRS#<Size>] [QRF#<Format>][QRP#<Path>] [DT#<NumberOfSeconds>] [/debug] [/console] [/stdout [/UI]

The first three parameters are obligatory and need to be given in the exact stated order. The others are not obligatory (the reference in the square bracket illustrates their optional nature, the brackets are not a part of the parameter).
If optional parameters are not stated, default values are automatically set. Optional parameters are identified by the prefix and can therefore be stated in random order.

<Demand Type>

This parameter is obligatory, possible values being **invoice**, **invoice-s, invoice-sum, invoice-c, invoice-b.** The accurate stating of this parameter is very important, since the communication with PU (The Financial Administration of Montenegro) works in a different way in every of the stated cases.

The ***invoice*** value is the most common way since it represents the applying of a regular invoice from an electronic device.

The ***invoice-s*** (as subsequently) is used when in the time of making the connection of the invoices with PU there was no possibility to connect and the confirmation is dealt with subsequently. While connectivity was not attainable (the internet on the location of the invoicing or the PU service was down) Fiskalizuj.exe was only able to return the IKOF (the receipt issuer's commercial seal) and the QR code, but not JIKR (unique invoice identifier granted by PU). This number needs to be attained later (in two days time). In this particular case, only the JIKR is added to the response file. The IKOF and the QR code bar are not included in the response file itself as they were already stated (and printed) on the invoice.

The ***invoice-sum*** (as summary) value is used when the summary invoice is issued.

The ***invoice-c*** (as correction) value is used in accordance with the instructions when, after the invoice confirmation, there has been a change in the invoice content (a wrong sum was used, a wrong tax rate was used, etc.). In this case, the invoice is sent again with the exact same invoice number for which the new IKOF, JIKR and QR code is acquired (all of this must be stated on the page of the issuer's record of the issued invoices). The record of the invoice that is sent as correction must include the data of the first invoice issue (about the invoice which needs corrections) in the fields RefIKOF, DatVrijemIzdKorRac, VrstaKorRac. The stated text is entered only in case of subsequent alternation of invoice data (without reversing the original invoice and issuing a new one).

The ***invoice-b*** (as book) value is used when an account issued from a pre-numbered invoice book is applied. This situation can occur because the issuer does not (yet) use the electronic device that is connected with PU and rather applies invoices subsequently, as far as the law allows in the transition period. This might also occur in examples such as accidental power cuts and when the invoice was not issuable in any other way but as a pre-numbered invoice book.

There are two more values: **version** and **update**. Most users can forget about these two values since Fiskalizuj.exe has an integral mechanism that automatically updates. When using the **version** value, a recording of the currently installed version and of the actual version on our server is made in the response file (or on the screen, should you use /console as well). With the **update** command, a renewal of the programme version is set off manually (if the two versions are different on the local drive and on the server).
In case these two values are used, the only needed parameter is **/console**.
Both orders are available for control purposes only. They are also available when the autoupdate option in the .config file would want to be switched off (which is not recommended whatsoever).

<Mode>

This is an obligatory parameter, possible values being **demo, dev, test, prod.** Here the caller himself can decide when the development or test service will become a production service.

In **demo** mode program returns randomly generated IKOF and JIKR, fictitious device, fictitious operator, fictitious QR content string and picture, and the current time (that's all to be placed on the invoice). The response is returned regardless of the correctness of the input data. In about 5% of cases, the response returns an error regarding the input data, in about 5% it returns an error regarding the implementation of fiscalization. In this mode, no actual link to the Tax Administration portal is made.

In mode **dev**, the input data is already fully controlled. If they are not ok, an error is returned describing the problem. If the input data testing succeeds, the fiscalization data, prepared in the same way as when operating in DEMO mode, are returned.
In this mode also, there is no actual connection to the portal of the tax administration, otherwise, the overall behaviour is exactly the same as in the production operation so the integrators can do all the options before obtaining the licenses.

Mode **test** - an appropriate license must already be entered for this operation. You still don't need to buy a license. You just have to obtain it from the fiskalizuj.online service and connect it to your device. In this mode, the input data is controlled in exactly the same way as in dev and prod mode. If they are not ok, an error is reported describing the problem. If the input data testing succeeds, the extensions return real IKOF, QRContent, invoice number, registered device, registered operator, and the current time. The real JIKR is also returned as the test environment of the tax administration returns it for this invoice. This operation mode differs from dev basically just in the way that fiscalisation data on the invoice are publicly verifiable. For integration purposes, there are no differences at all. All input data are fully controlled and all output data are technically the same as in dev mode.
But again - **JIKR is granted by the Tax Administration of the Montenegro and the fiscalisation proceeding is publicly verifiable through the QR code on the invoice**.

Mode **prod** - all the same as TEST, accepts that the fiscalization is carried out on the production environment of the tax administration. The production communication only begins when the parameter is set (the moment of the transition is determined by the caller) and the server side has already approved that all the necessary registration numbers (company, location, device, operator) is set and the server side has access to the actual certificate. And another little important thing – the license for this device is paid.

F#<File_name> | S#<inv_string>

The parameter is obligatory. In the first form it includes the path and the name of the file in which you have prepared the invoice information. It is the safest to state the parameters in double quotation marks. More detailed instructions on the preparation of the text file including the invoice data are located in a separate file.
Instead of in the file, the invoice data could be prepared in the form of a serialized string (how described for .TXT form) and handed to fiskalizuj.exe in that form. S# prefix must be used in this case.

RF#<ResponseFile>

If the parameter is not stated, the value from the .config file is taken as default. This .config file is set by default to ***response/STResponse.txt***, which means the file with the contents needed by the caller will be written in the directory response that is located inside the directory, where Fiskalizuj.exe was installed. In a multi-user environment (if Fiskalizuj.exe is installed on the network disc and is called by multiple cash-registers or if Fiskalizuj.exe is installed on a terminal server) the parameter must be stated so that an automatic response is assured for the caller. We recommend to always state the parameter. We also propose that you set clearly an exact location where the returned data is expected. You can for example use the parameter with invoice number and you will get the separated response file for every call.
In the response file are stated: IKOF, JIKR, FiscalizationTime and (optional) the name of the file with the prepared bar code. The file containing the QR bar code is on the exact same spot as the response file. It has a generic name that is stated in the response file.
More detailed instructions on the content of the response file are located in a separate file.

QRS#<Size>

The QRcode size - the approximate height in centimetres.
The possible values are **from 2 to 5** and approximately correspond to the picture height which is returned by the programme. If the parameter is not stated, the default value is 2.

QRF#<Format>

Possible values are **PNG, JPG, IMG, GIF, WMF and TIFF** (additional desired formats may be added, let us know if you want us to do so).
If the parameter is not stated, the default value is PNG.

QRP#<Path>

Possible values are **false**, **true**, **<file_path>**
Value **false** - the image file will not be generated and written. In this case, the programmer takes the base64 coded value from the response and does everything necessary with it.
Value **true** - the image files will be written to the same location as set for the response file (by default in the response directory, otherwise where the RF# parameter says).
If you specify **a path** to an existing directory, image files will be written to that location.
If the parameter is not stated, the default value is false (which means that by default the image is not created and not written to disk).

DT#<NumberOfSeconds>

DT as in Delay Time. The parameter is not obligatory, yet it enables the time setting when the decision is made that the communication with PU will not be established and that the JIKR is not attainable. As a result, an error will be stated in the JIKR. An invoice of this type must be sent into subsequent confirmation. If the parameter value is not stated, the default value of 5 seconds will be used. If there is no response after the sending of the http request in the stated period of time, the connection is recognised as unsuccessful.

/console

This is an optional parameter used for the test run of the command from the command bar (cmd, command, powershell, etc.). If the parameter is present, the content of the response is written on the screen as well. This is not a detour because everything in the response file remains the same. It only means that you do not need to open the response file and check it at the same time since the response appears simultaneously on the screen.

/stdout

This is an optional parameter intended for the programmers who prefer to read the server's response from stdout instead of the response file. If the parameter is present, the content of the response is written on the stdout only (the response file is not made).

/debug

This is an optional parameter for the test check of the used time by steps. If this parameter is used, a log.txt file is created in the work directory. In this file, the essential steps are written in milliseconds stating the time spent doing it.

/UI

Fiskalizuj.exe is a console application, made to integrate smoothly into your application. UI is used only in the initial time for uploading the certificate and all the necessary registration numbers (obtained from PU in SEP portal) to fiskalizuj.online servers, and to obtain the license from the fiskalizuj.online servers.
However, you can integrate this UI into your application also. When /UI is used, all other parameters are ignored.

Example1 - Test call with the obligatory parameters only:

```
FISKALIZUJ.EXE invoice test F#''c:\fiskalizuj\invoice.txt''
```

Example2 - Prod call with response file and QR code as picture stated:

```
FISKALIZUJ.EXE invoice prod F#''c:\inv\inv1.txt'' RF#''c:\inv\response\resp.txt'' QRF#PNG QRS#3
QRP#true
```