

# Documentation

## Interfaces





***Software IPCS version 1.2.3***



**10/2022 EN**

iret GmbH  
Paul Engelhard Weg 50a  
D-48167 Münster

 +49.251.615138  
 +49.251.617631

E-Mail [info@iret.de](mailto:info@iret.de)  
Internet [www.iret.de](http://www.iret.de)

---

# Table of Content

<b>Revision history</b> .....	<b>5</b>
<b>1 Introduction</b> .....	<b>6</b>
1.1 General.....	6
1.2 Overview of the different import methods.....	6
1.2.1 Spreadsheet files.....	6
1.2.2 WPCS files.....	6
1.2.3 REST service with JSON.....	6
1.3 Overview of the different methods to get production feedback.....	6
1.3.1 CSV files.....	6
1.3.2 WPCS files.....	6
1.3.3 REST service.....	7
1.4 Concepts common to all data import formats.....	7
1.4.1 Importing article lists.....	7
1.4.2 Creation of jobs.....	7
1.4.3 Creation of material data during import.....	7
<b>2 Import of spreadsheet files</b> .....	<b>8</b>
2.1 Introduction.....	8
2.2 Supported file formats.....	8
2.3 Column definitions.....	8
2.3.1 Automapping.....	9
2.3.2 Print positions for inkjet marking.....	9
2.3.3 Print positions for label marking.....	10
2.3.4 Marking of Insertion depths.....	10
2.4 Import of multiple articlelists.....	10
2.5 Extended text import (optional).....	10
2.6 Troubleshooting and FAQ.....	11
2.6.1 The import fails completely, what to do?.....	11
2.6.2 Why is a print text not or incorrectly imported?.....	11
2.6.3 Why are so many lines empty and ignored?.....	11
2.6.4 Does the PrintRepeatX have to be specified for each print text?.....	11
2.6.5 Why do I get so many missing columns in the report, although I didn't specify them or don't need them at all?.....	11
2.6.6 How does the error handling work?.....	11
<b>3 WPCS-Import</b> .....	<b>12</b>
3.1 Introduction.....	12
3.2 Possible Importfiles.....	12
3.2.1 Article.dds.....	12
3.2.2 Job.dds.....	14
3.2.3 Wire.dds.....	14
3.3 Feedback files concerning errors in import files.....	14
3.4 Feedback during production.....	14
3.4.1 job.sdc.....	14
<b>4 REST-Service</b> .....	<b>16</b>
4.1 Introduction.....	16
4.2 Menu.....	16
4.3 Documentation.....	16
4.3.1 Schemes.....	17
4.3.2 Example scheme description.....	17
4.4 Production log reporting.....	17

---

4.5 Error codes and descriptions.....	18
<b>5 Settings.....</b>	<b>19</b>
5.1 Options for job creation.....	19
5.2 Report settings.....	20
<b>6 General descriptions concerning marking.....</b>	<b>21</b>
6.1 Formatting CIJ marks.....	21
6.1.1 List of dynamic placeholders.....	21
6.1.2 List of format commands.....	21

# Revision history

---

## Revision history

Revision	Date	Description of changes	Originator
initial	year 2021	initial revision	SH
0.7.0		<ul style="list-style-type: none"><li>Added REST import</li></ul>	
0.9.0	2022-02-14	<ul style="list-style-type: none"><li>Added chapter 6.1 describing formatting of CIJ marks</li></ul>	RB / SH
0.9.0	2022-02-15	<ul style="list-style-type: none"><li>Re-Added REST log image, Fix of misspelling, Updated tested format commands</li></ul>	RB
1.0.0	2022-05	<ul style="list-style-type: none"><li>Minor fixes</li></ul>	MB , SH
1.2.3	2022-11-02	<ul style="list-style-type: none"><li>Additional information for the REST and JSON-Import</li></ul>	RB
1.2.3	2022-11-04	<ul style="list-style-type: none"><li>REST menu and additional REST introduction</li></ul>	RB

# 1 Introduction

---

## 1 Introduction

### 1.1 General

This document describes the various methods to exchange data with Iret Pipe Cutting Software (IPCS). While the import of data is the most common use, this document also covers the production feedback one may get from the system.

If you are not familiar with IPCS we recommend to first read the user manual or – even better – to ask people in your production department for details of the software and the articles to be produced.

### 1.2 Overview of the different import methods

#### 1.2.1 Spreadsheet files

The interface for spreadsheet files is an easy solution to import files created either manually or by an ERP-system. While IPCS requires a fixed names for the columns of your spreadsheet it supports different fileformats:

- Excel files (“.xls” or “.xlsx” or “.xlsm”)
- OpenOffice files (“.ods” or “.odt”)
- CSV files with “;” separator
- JSON files (“.json”) (For further information on JSON-Import see the YAML documentation)

#### 1.2.2 WPCS files

These files are ASCII files commonly used in the cable processing industry and follow a standard introduced by Komax company. IPCS supports a subset of this standard.

#### 1.2.3 REST service with JSON

The REST service is a standalone software acting as a webserver capable of importing data in JSON format. This is the most flexible way to import data in IPCS but also is the most difficult one to work with. While WPCS and spreadsheet files may be generated manually, the webservice normally requires some programming on the side of the data provider.

REST service requires an additional licence while WPCS & Spreadsheet import are included in the basic licence.

### 1.3 Overview of the different methods to get production feedback

#### 1.3.1 CSV files

A CSV file is generated during production, it contains one or more lines for each article produced.

#### 1.3.2 WPCS files

IPCS will write a job file, the production file is not yet supported.

### **1.3.3 REST service**

IPCS will send a JSON document for every article produced.

## **1.4 Concepts common to all data import formats**

### **1.4.1 Importing article lists**

This is the key feature of all import methods: A list of articles will be imported and can be used afterwards to create jobs. Each article in the list has a length, an associated material (s. b.) and may have a number of different operations like inkjet markings, label marks and similar operations.

### **1.4.2 Creation of jobs**

A job is used to put an article list into production. Thus a job contains the name of an article list, a quantity and priority.

The creation of jobs can be configured inside IPCS, see chapter 5.1 for details.

### **1.4.3 Creation of material data during import**

The material of an article is used as a reference for common settings used to produce all articles with this material. Such settings may be transport speed, inkjet font etc. .

The material settings themselves can not be imported, instead it is left to the operator to prepare the settings once.

Creation of material settings may occur before or after the import. If a new material is recognized during import, a new material entry will be created in the database. All settings are default values in this case which may not lead to best production results.

# 2

## Import of spreadsheet files

---

## 2 Import of spreadsheet files

### 2.1 Introduction

IPCS supports data import from various file formats of spreadsheet software. Thereby one spreadsheet table represents an article-list and/or job with every line of the spreadsheet representing an article with print texts, possible labels and insertion depths.

### 2.2 Supported file formats

When importing data, it is possible to prepare and import data in Microsoft Office Excel formats (.xls or .xlsx), OpenOffice format (.ods, .odt), CSV formats (.csv or .txt), JSON-format (.json).

### 2.3 Column definitions

The following table shows all valid column names. One of these names should be present in every column of the first row of the import file. Some of the columns are required in every file.

Name	Type	Digits / range	Description
<b>Name</b>	Name (Required)	40	The name of the article (cable, hose, pipe, etc.)
<b>MaterialName</b>	Name (Required)	80	The name of the material that is used.
<b>LengthMM</b>	Length (Required)	0.1 - 1000000.0	The total length of this article. <b>Import filter:</b> You can use an import filter if your lengths are in meters, for example. See manual chapter 3.8.
Quantity	Integer	0..10000000	The quantity to be produced.
AlternateName	Name	40	An alternative name which is used instead of the article name in case of special cases or errors.
Depth1	Length		The insertion depth at the beginning of the cable.
Depth2	Length		The insertion depth at the end of the cable.
PrintTextX	Text		The texts that will be printed (X is a placeholder for a number here). Several texts can be transmitted. This is indicated by an appended enumeration to the column name (e.g. PrintText1, PrintText2). The same numbering indicates that they belong together to PrintPositionsX (eg. PrintPosition1, PrintPosition2 etc).
PrintPositionX	Print position	0.0 - Total length	This column contains the print positions of the texts (X is a placeholder for a number). Several print positions for different texts can be transferred. This is indicated by an appended enumeration to the column name (e.g. PrintPosition1, PrintPosition2). The same numbering indicates that they belong to PrintTextX (eg. PrintText1, PrintText2 etc).
PrintRepeatX	Integer		This value indicates how many times the text will be printed on the cable. Here, too, the appended numbering (X) refers to a text (PrintTextX). The positioning is done automatically based on the size of the text as well as the length of the article and is distributed on the cable.



Name	Type	Digits / range	Description
LabelTextX	Text		The text to be printed with the label. Again, the attached enumeration number determines which LabelText belongs to which LabelPosition and LabelTemplate.
LabelPositionX	Label position	0.0 - Label length	The position where the label is to be printed or wrapped. Again, the appended enumeration number determines which LabelText belongs to which LabelPosition and LabelTemplate.
LabelTemplateX	Template		The template into which the corresponding LabelText is inserted. Only the template name is passed here. The label must be present in the label folder.
JobName	Name	80	This column sets the name of the articlelist. One row equals one article and every article with the same JobName will be part of the same articlelist.
Priority	Integer	1-9	The priority of each article within the articlelist or in production. Default value: 5
PostProcessing	Integer	0-1	The post processing of each article. Default value: 0 → No post processing

### 2.3.1 Automapping

The data import is based on the column names, i.e. the column headings in the spreadsheet must correspond exactly to the column definitions (see 2.3).

The columns are then assigned to the internal fields they are intended for. The order of the columns in the spreadsheet is irrelevant. The column headings must be in the uppermost row of the spreadsheet so that the mapping can be done correctly and the columns can be recognized and assigned to the appropriate fields.

The columns "Article", "Material" and "LengthMM" are necessary and must be present for proper functionality. All others are either optional or can be created with default values.

### 2.3.2 Print positions for inkjet marking

The print positions indicate where the printing starts. It is assumed that the right end of the article (cable, hose, pipe, etc.) is 0, since the article usually runs through the machine from left to right.

The columns for PrintText, PrintPosition and PrintRepeat belong together and can occur several times. The PrintRepeat column is optional and can be omitted if no repetition of individual print texts is desired. The assignment of PrintText column and PrintPosition column is done by the numbering attached to the column name.

Example: "PrintText2", "PrintPosition2", "PrintRepeat2" belong together and define the necessary settings for the print text. In this example it is also not necessary to have "PrintText1" as a column. As long as the numbering at the end of the column name (here "2") matches, the columns are evaluated together.

Negative position values are possible and mark the distance from the (left) end of the article.

# 2

## Import of spreadsheet files

---

### 2.3.3 Print positions for label marking

The columns LabelText, LabelPosition and LabelTemplate can also occur several times and also here the numbering at the end of the column name determines which of the columns belong together and will be evaluated together.

### 2.3.4 Marking of Insertion depths

If insertion depths are imported (columns "DepthLeft", "DepthLRight", see 2.3), markings (" $<$ " or " $>$ ") are printed on the left or right article end. These markings cannot be changed. The value for "DepthRight" is automatically set negative, regardless of its sign.

The order of the print positions and texts in the columns as well as in the display in the article is irrelevant. Only the position specification determines the positioning on the cable.

## 2.4 Import of multiple articlelists

By default an import-file is seen as one articlelist, created with the name of the file. But you can also create multiple articlelists with multiple names out of one import-file.

In order for this to be successful your import-file needs the column "JobName". If this column is present our software recognizes this and takes the value within it as the new name for the articlelist. This means that for every different value in "JobName" an articlelist is created with this name and every row with the same "JobName" is imported into the same articlelist.

The order of the rows makes no difference so that rows with the same "JobName" do not have to be next to each other in the import-file. They just have to be in the same import-file (with maybe other articles and differing "JobName").

It is not possible to overwrite existing articlelist. If the name already exists an enumeration is added to the end of the name until the name is unique.

## 2.5 Extended text import (optional)

If IPCS has been purchased with the option "Extended print options", up to 10 single and multi-line print texts can be imported from existing Excel tables for label printing. Here, too, the "automapping" described in chapter 2.3.1 applies, i.e. only the column name of the existing Excel table is decisive for the text import, not the positions of the columns in the table

Name	Type	Digits / range	Description
„CommentShort1“ - „CommentShort5“	Text	Max. 50	Single line comment
„CommentLong1“ - „CommentLong5“	Text	max4096	Multi line comment

To use the imported print texts, one or more labels must be created on which corresponding placeholder texts exist. These are replaced by the imported texts for printing. For information on creating labels, please refer to the separate instructions "Using CAB label printers with IPCS software".

## 2.6 Troubleshooting and FAQ

### 2.6.1 The import fails completely, what to do?

Please check if the format of the import file fits and if all columns are described correctly. Does the format fit? (see point 2.2) Are all necessary columns available? (see point 2.3) Are they spelled correctly?

### 2.6.2 Why is a print text not or incorrectly imported?

Please check whether the columns "PrintTextX" and "PrintPositionX" exist and are filled correctly. In addition, the numbering of the columns must match, e.g. "PrintPosition4" and "PrintText4", as this is absolutely necessary for a correct import.

### 2.6.3 Why are so many lines empty and ignored?

This is due to the fact that you cannot reliably find out the number of valid lines in the import files. Here you get only the total number of lines. I.e. if you enter something in line 70, although there are articles only in the first three lines, we check the 70 lines, because we only have the info that a maximum of 70 lines are described and only then we check if these lines have the necessary data or are not filled.

### 2.6.4 Does the PrintRepeatX have to be specified for each print text?

If the PrintRepeatX is not specified for a print text or the PrintRepeatX column does not exist in the import file, the text is printed only once and is not repeated. If one wants however several repetitions of several print texts then one must indicate these for each print text individually.

### 2.6.5 Why do I get so many missing columns in the report, although I didn't specify them or don't need them at all?

This is due to the fact that all table columns are imported first. The columns that are not needed are then displayed in the record. This has no effect on the lines to be imported, provided that all required columns are correctly titled and filled.

### 2.6.6 How does the error handling work?

Every error and inconsistency in the import is displayed in a log window, which can also be saved separately. Finally the number of successfully imported records (1 line = 1 article in the article list) is also indicated.

The import is largely tolerant of errors, i.e. not every inconsistency leads to a termination of the import. All lines containing valid values in the "Required" columns (see 2.3) are counted as "successfully imported", regardless of data found in the "Not required" columns. This allows you to manually edit the imported data after the import.

However, all lines without valid values in one of the "Required" columns are ignored and are not imported.

# 3

## WPCS-Import

---

### 3 WPCS-Import

#### 3.1 Introduction

The WPCS import follows a standard of the company Komax for the import of cable data. The interface is based on ASCII-files with a syntax similar to Windows ini-files:

```
[DeleteArticle]
ArticleKey    = TEST-WPCS-001

[NewArticle]
ArticleKey    = TEST-WPCS-001
NumberOfLeadSets = 4
Name = "Test Article WPCS Import 001"
  [NewLeadSet1]
    Name      = " 1:"
    WireKey   = 35.00 BLACK
    WireLength = 1100
```

Complete documentation is available from Komax, in this document only the specifics of the implementation in IPCS software are described.

#### 3.2 Possible Importfiles

There are 3 different WPCS-files that can be imported:

Filename	Description
<b>Job.dds</b>	A job and productionentries will be created.
<b>Article.dds</b>	An articlelist with articles and print-, mark- and labelpositions will be created. (In case the material does not exist it will be created with default values)
<b>Wire.dds</b>	Materials with their properties.

#### Hint!

These files have to be named exactly like it is stated above or else they will not be recognized and therefore also not imported.

#### 3.2.1 Article.dds

List of supported keys:

Block	Supported key names	Comments / Limitations
DeleteArticle	ArticleKey	
NewArticle	ArticleKey	
	Name	
	Hint	
NewLeadSet#	Pieces	
	BatchSize	

---

<b>Block</b>	<b>Supported key names</b>	<b>Comments / Limitations</b>
	Name	
	Hint	
	WireKey	
	WireLength	
NewMarkingTextWire1-2	MarkingTextBegin	
	MarkingTextEndless	
	MarkingTextEnd	

# 3

## WPCS-Import

---

### 3.2.2 Job.dds

List of supported keys:

Block	Supported key names	Comments / Limitations
DeleteJob	Job	
NewJob	Job	
	ArticleKey	
	TotalPieces	
	BatchSize	
	Hint	

### 3.2.3 Wire.dds

List of supported keys:

Block	Supported key names	Comments / Limitations
NewWire	WireKey	limited to 20 characters in ipcs software
	Color	
	Diameter	

### 3.3 Feedback files concerning errors in import files

When encountering errors or not evaluating keys and sections while importing WPCS-files the following feedback files will be created:

- job.edc
- article.edc
- wire.edc

These files contain the errors or whatever went wrong when importing this file.

### 3.4 Feedback during production

If the report format of IPCS (see chapter 5.2) is set to report via WPCS it will constantly write to the file "job.sdc" as described below.

#### 3.4.1 job.sdc

List of supported keys:

Block	Supported key names	Comments / Limitations
JobStarted	DateTimeStamp	
	Job	
	ArticleKey	
	UserName	
	JobRequestedPieces	
JobRestarted	All keys from "JobStarted"(s.a.)	

---

<b>Block</b>	<b>Supported key names</b>	<b>Comments / Limitations</b>
	UserRequestedPieces	
	TotalGoodPieces	
JobInterrupted	All keys from "JobRestarted" (s.a.)	
JobTerminated	All keys from "JobRestarted" (s.a.)	
JobAborted	All keys from "JobRestarted" (s.a.)	
	Reason	

# 4

## REST-Service

---

### 4 REST-Service

#### 4.1 Introduction

Article lists can now optionally be imported through a REST API directly into IPCS. This requires both a IPCS license with REST-API option and the station running IPCS to be connected to and reachable over the network. If the application is running there will be a little tray icon.

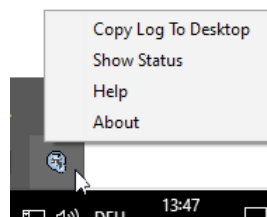
---

#### Hint!

The REST-Server has to be started separately because it is an external part of the IPCS. If it is purchased it is called "ipcs\_restserver.exe" and it should be in the program folder of IPCS. Also it can just be closed by the task manager, so that elevated privileges are required.

---

#### 4.2 Menu



Menu item	Description
Copy Log To Desktop	The log file will be copied to the desktop. There is operational and debug information about the program stored.
Show Status	Opens a dialog with two tabs: 1. The first one ("Import") is where the received JSON is shown and also the server can be stopped and started again at a custom port. 2. The second tab is to test the reporting via rest on a custom url and port. There the report should arrive (see 4.4 for further information)
Help	Opens this documentation for further information on the application and how to work with it.
About	A small "About"-dialog with copyright and company information.

#### 4.3 Documentation

The REST API documentation is provided as a separate document following the OpenAPI specification and should have been provided along with this manual (Got to the installfolder and find it under: .../doc/v1.0.yaml ).

We recommend Swagger to render the OpenAPI specification document and automatically generate client stubs based on the provided specification. The "live demo" available under <https://editor.swagger.io/> is sufficient for this.



### 4.3.1 Schemes

The schemes described in the specification document pertain to the following:

Schema	Description
error	Response sent by server when an error has occurred
articleList	Article list import format accepted by the server
article	Schema describing a single article (hose, wire, pipe, ...)
productionlog-article	Extends the <i>article</i> schema.  Only for productionlog reporting via REST (see chapter 4.4 for further information).

### 4.3.2 Example scheme description

The *articleList* schema above describes the format of a article list accepted by the server.

A matching JSON structure would be

```
{
  customer: "Customer name",
  articles: [
    { ... },
    { ... }
  ]
}
```

Attribute	Type	Description
customer	String	Value "Customer Name"
articles	Array	Array of objects pertaining to the <i>article</i> schema (here abbreviated through ...)

The \* behind an attribute name signifies this attribute as *required* by the server to process a request. Failure to provide required attributes will result in an error response from the server.

## 4.4 Production log reporting

IPCS can be configured to send production log reports to another REST server. For this, IPCS will additionally act as a REST client.

A report log entry is submitted whenever

- production is stopped
- production is finished

The submitted log entry matches the *productionlog-article* schema (see specification document). It will be sent via "POST"-Method to the URL configured in IPCS (see chapter 5.2).

# 4

## REST-Service

---

### 4.5 Error codes and descriptions

The *msg* attribute of the *error* response schema is not localized and is intended for developers only.

Further information about how the import went in detail you can find the report with more detailed information in the report folder.

The HTTP error codes are documented in the YAML structure.

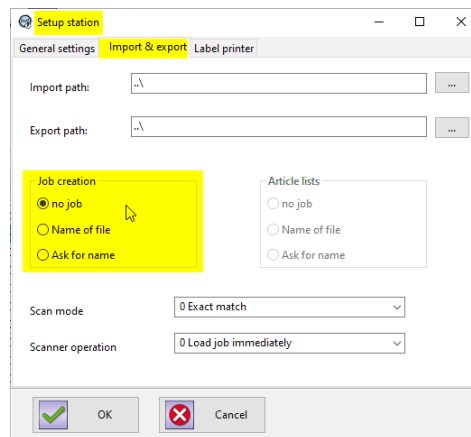
The following table contains a list of all possible return codes and their meaning.

error code	Description
38903	An error occurred reading the json file. The json could not be read and therefore no import happened.
38913	Initialization failed. No valid json, charset or other reasons that cause the initialization to fail.
39101	Payload error articlelist. If the articlelist was not found or has the wrong type.
39102	Required json key not found in articlelist.
39103	Optional json key not found in articlelist.
39201	Payload error article. If the article was not found or has the wrong type.
39202	Required json key not found in article.
39301	Payload error marking position. If the marking position was not found or has the wrong type.
39303	Optional json key not found in marking position.
39701	Method not allowed. Only "PUT" allowed everything else will be denied.
39702	Import failed. Something went wrong an no import happened.
39704	Name already exists. The name already exists and therefore couldnot be imported automatically.

## 5 Settings

### 5.1 Options for job creation

The settings for creating jobs (orders) for imported article lists can be changed in the station settings:



Function	Description
No job	No job will be created, so no entries for production are created.
Name of file	The job is created with the name of the imported file.
Ask for name	The job name is queried during import and can be entered there. (Not possible for the REST-Service import where the name of the file is automatically taken in this case)

Further details about the jobs or orders and further functionalities are described in the general manual.



#### Note!

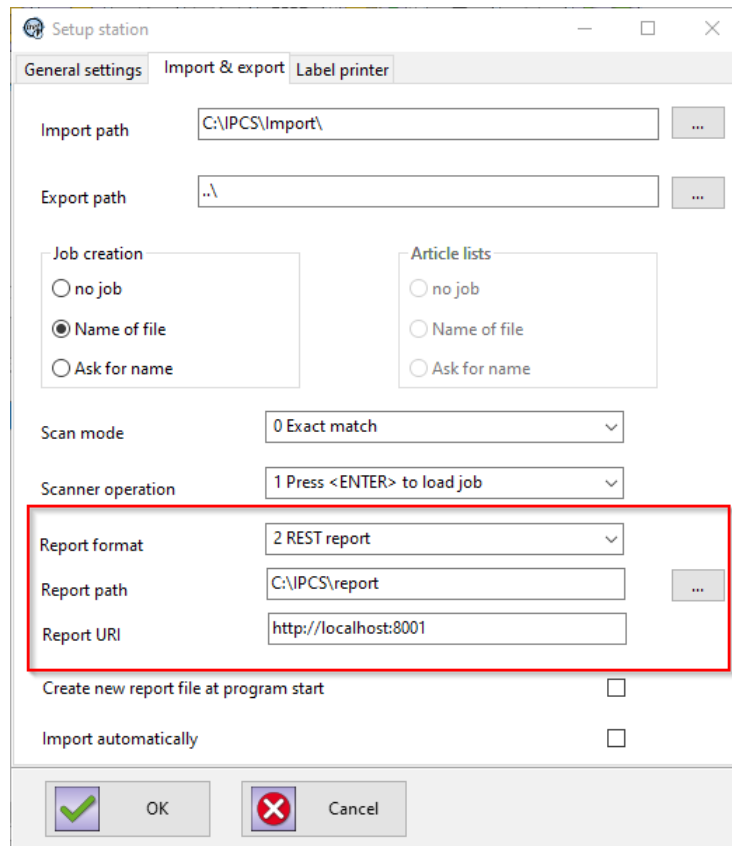
- The REST and JSON-Import do NOT use the settings above, they get the name of the articlelist or job from the JSON-file itself with the key "name" (see the YAML-documentation for further information on the syntax)
- Changing the value of the "Job creation"-property requires a restart of the REST server.

# 5

## Settings

### 5.2 Report settings

To receive the report via REST a few settings must be set:



Function	Description
<b>Report format</b>	This property must be set to “REST report” to be able to receive the report via REST.
<b>Report path</b>	This path is only important for the CSV report and a few import functionalities (For further information see general manual)
<b>Report URI</b>	This URI defines where the REST report will be sent to. i.e.: http://Servername:Port/Path Where the placeholders have to be replaced with the real values of server hostname, port and path where the report is expected.

## 6 General descriptions concerning marking

### 6.1 Formatting CIJ marks



**Note!**

The following descriptions are only valid for CIJ markings, and must not be used in clamp marks or label marks.

#### 6.1.1 List of dynamic placeholders

Tag (tbc*)	Meaning	Example
@@	<b>To print an „@“ sign an additional „@“ is needed for avoiding misinterpretation.</b>	
@1	Current time in the format hh:mm:ss	
@2	Current date in the format tt.mm.yyyy	
@3 (tbc*)	Quantity produced	
@4 (tbc*)	Batch produced	

\*tbc = to be confirmed, the function of these tags are not guaranteed to work properly and have to be checked on request first.

#### 6.1.2 List of format commands

Tag (tbc*)	Meaning	Example
\&1	Barcode Code 39	\&11234\&
\&2	Barcode 2/5 interleave	\&21234\&
\&3	Barcode EAN 13	\&31234\&
\&4	Barcode 128	\&41234\&
\&5	Barcode EAN 128	\&51234\&
\&6	Barcode UPC-A	\&61234\&
\&7	Pharmacode	\&71234\&
\&8	Barcode EAN 8	\&81234\&
\&9	Barcode UPC-E	\&91234\&
\&a	Barcode EAN 128C	\&a1234\&
\&d	Codabar	\&d1234\&
\&D (tbc*)	User defined date format	
\&F (nyi*)	Name of the Windows-Bitmap File	\&Fipcs.bmp\&
\&T (tbc*)	User defined time format	
\&	End-Tag	

\*tbc = to be confirmed, the function of these tags are not guaranteed to work properly and have to be checked on request first.

\*nyi = not yet implemented