REA JET

INDUSTRIAL CODING AND MARKING SOLUTIONS - MADE IN GERMANY

Controlling the REA JET print and marking devices from an external processing unit using REA API

Version 3.9.0· 2023-11-15

Credits

Version:	3.9.0 English
Article no.:	---
Technical Editing:	Karlu Diehl Dr. Matthias Mann Rainer Wiefelspütz

REA Elektronik GmbH
Teichwiesenstrasse 1
D-64367 Mühltal, Germany

Tel.: +49 (0)6154-638-0
Fax: +49 (0)6154-638-195
Email: info@rea-jet.de

Version	Author	Date	Annotation
1.0	MBA	20.10.2008	document created
1.0.1	MBA	21.11.2008	reviewed and published
1.60.0	KD	13.12.2009	reviewed and published
1.70.1	MKU	20.05.2010	reviewed and published
1.81.0	KD	22.11.2010	reviewed and published
2.01	AP	18.02.2011	reviewed and published
3.02.3	KD	16.11.2011	reviewed and published
3.20	KD	2011-08-19	reviewed and published
3.20.5	KD	2012-02-15	reviewed and published
3.21.1	KD	2012-06-19	reviewed and published
3.3.0.1	KD	2013-10-16	New event MissingContent and behaviour of no printbuffer mode. Changes in event PrintRejected and behaviour of print reject mode
3.3.0.2	KD	2014-10-03	Added Configuration of NoPrintbufferMode and command PrintRejectConfirm. In addition some corrections and many changes in format…
3.3.5.1	KD	2014-11-20 2015-06-19	Labelmanipulation for ContentBuffer with multiple content values. Added new label content properties buffer size, printed and expired. New function to retrieve production data.
3.4.01	KD	2015-11-23	Command GetDeviceInfo aktualisiert: neue Eigenschaften Artikelnummer und FPGA-Version
3.5.0.0	MMa	2015-06-26	Revised content buffer.
3.5.0.1	KD	2015-07-23	Added functionality for hardware productsensors, digital input and digital output . Revised chapter device settings I/O configuration.
3.5.0.2	MMa	2015-10-28	GETLABELCONTENT and GETLABELOBJECT examples corrected
3.6.0.1	KD	2015-12-18	Funktionalität Kommando TRIGGERGENERICEVENT und Event GENERICEVENT
3.6.0.2	KD	2016-07-21	Erweiterung Print Events bei Mehrtriggerfähigkeit
3.6.0.3	KD	2016-07-21	Korrekturen bei Version, Format, …
3.6.0.4	KD	2016-09-09	Korrekturen für Version 3.6 …
3.6.0.5	LNE	2017-08-09	Neues CI
3.6.0.6	KD	2017-08-22	Add description for new device types DOD and CIJ, especially DEVICEINFO
3.6.0.7	MMa	2017-08-25	Review
3.6.0.8	KD	2017-09-04	Change image on front page
3.6.0.9	KD	2017-09-18	Change Parameter of SET- and GETIOCONFIGURATION, JobID is obsolete
3.6.0.10	KD	2017-12-06	Obsolete document references removed
3.6.0.11	KD	2018-10-12	Korr. Arguments Event Subscription, div. corrections
3.6.0.12	KD	2019-09-18	Format and correct label and Labelcontent properties
3.9.0	RW	2023-11-15	Change image on front page, minor adjustments / extensions, document numbering modified

REA JET

Table of Contents:

1 Introduction

The **R**EA **E**lektronik **A**pplication - **P**rogramming **I**nterface (REA-PI) is an XML based communication protocol for controlling coding and marking systems based on the REA JET TITAN Platform. At time of writing, this comprises the following devices:

- HR product family, e.g., HR, HR pro, HR pro OEM, HR 2.0
- Laser Systems, e.g., CL, FL
- DOD 2.0
- SC 2.0
- ST
- GK 2.0
- UP

REA-PI is implemented by exchanging XML-documents over a TCP connection. All REA JET TITAN devices listen on TCP port 22171 for incoming REA-PI connections, i.e., act as *server*. A *client* is any host connecting to and controlling a coding and marking system. Connectivity between clients and servers is *many-to-many*, i.e., a single host might control several coding and marking systems, whereas several clients can also connect to a single REA JET device the same time.

2 Data Exchange

All REA JET TITAN devices offer a network file share (CIFS/SMB) as well as FTP access for data exchange. For further technical instructions refer to the respective user manuals for the device.

3 Message Frame

Outgoing and incoming XML messages both consist of a header and user data. This applies to the XML handshake messages as well. The header includes the length of the following XML content.

The header consists of information about the following user data. Every header item is terminated by a line feed token. All header items together are finally terminated by an additional line feed. The header is not explicitly stated in the following chapters.

Currently only the header item '`Content-Length: value`' with value is supported. The Content-Length value is the number of bytes of the following xml data.

In previous versions the '`Content-Length: `' the empty space after the colon was strictly required.
Since Version FW3.37 this behaviour is changed. Now '`Content-Length:`' allows none or any number of spaces after the colon followed by the number of bytes.
The following number of digits is variable. It depends on the number of bytes.
E.g. '`Content-Length:123`', '`Content-Length: 1045`'.

In the following example the Content-Length will be the length of outgoing or incoming byte stream including the XML string from <REA-JET> up to the end </REA-JET>.

```
Content-Length: 107

   <REA-JET>
      <REA-PI version="3.9">
         <Command name="COMMAND NAME" id="identifier">
            <Data>Hello</Data>
         </Command>
      </REA-PI>
   </REA-JET>
```

This listing of the REA-PI message user data is formatted in a human readable form, where 'COMMAND_NAME' and 'identifier' are placeholders for a concrete command and identifier, which is a positive numeric value. <Data> most of the time has to be filled with command specific parameters.

The following listing describes the complete message as byte stream. The **red** marked area is the header with the content length of the following data, which is marked **blue**.

```
Content-Length: 123[x0A][x0A]<REA-JET><REA-PI version="3.9"><Command name="COMMAND NAME"
id="identifier"><Data>Hello</Data></Command></REA-PI></REA-JET>
```

The resulting byte array can be displayed as:

```
 43 6F 6E 74 65 6E 74 2D  4C 65 6E 67 74 68 3A 20
 31 32 33 0A 0A 3C 52 45  41 2D 4A 45 54 3E 3C 52
 45 41 2D 50 49 20 76 65  72 73 69 6F 6E 3D 22 33
 2E 39 22 3E 3C 43 6F 6D  6D 61 6E 64 20 6E 61 6D
 65 3D 22 43 4F 4D 4D 41  4E 44 5F 4E 41 4D 45 22
 20 69 64 3D 22 69 64 65  6E 74 69 66 69 65 72 22
 3E 3C 44 61 74 61 3E 48  65 6C 6C 6F 3C 2F 44 61
 74 61 3E 3C 2F 43 6F 6D  6D 61 6E 64 3E 3C 2F 52
 45 41 2D 50 49 3E 3C 2F  52 45 41 2D 4A 45 54 3E
```

For the character encoding of REA-PI messages always UTF-8 is required.

Remark, that UTF-8 uses one or more byte to encode a character! This is resulting in the statement:
Length of the XML-command string is not necessarily equal to length of resulting byte stream!
Examples for UTF8 byte encoding:

- german umlaut "<Char>Ü</Char>" [String length 14] as bytes "<Char>[xC3][x9C]</Char>" [Byte size 15]
- chinese character "<Char>中</Char>" [String length 14] as bytes "<Char>[xE4][xB8][xAD]</Char>" [Byte size 16]

It is important that the value of the **Content-Length is calculated on the byte stream instead of the string**, because the data is UTF-8 encoded. A content length calculation on the string will work, if only ASCII characters are used. If there are UTF-8 characters which need more than one byte for encoding, the calculated length would be wrong!

Make sure, that the content is ready for XML usage!

Pay attention:

- ***Xml node tag and attribute names*** are case sensitive.

	- For Example '<ContentValue value="102">' for ***not allowed characters in XML node values***! For example a string may contain something like '<Content>100>87</Content>'. The character '>', '<', '&' and some more characters have to be encoded as XML Entity, e.g. '<Content>100>87</Content>'.

```
<REA-JET>
   <REA-PI version="3.9>
      <Command name="COMMAND_NAME" id="identifier">
         <Data>
            <Xml>100&gt;87</Xml>
            <Ger>Ü</Ger>
            <Cn>中</Cn>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

```
Content-Length: 163[x0A][x0A]<REA-JET><REA-PI version="3.9><Command name="COMMAND_NAME"
id="identifier"><Data><Xml>100&gt;87</Xml><Ger>[xC3][x9C]</Ger><Cn>[xE4][xB8][xAD]</Cn></Data></Co
mmand></REA-PI></REA-JET>

 43 6F 6E 74 65 6E 74 2D  4C 65 6E 67 74 68 3A 20
 31 36 33 0A 0A 3C 52 45  41 2D 4A 45 54 3E 3C 52
 45 41 2D 50 49 20 76 65  72 73 69 6F 6E 3D 22 33
 2E 39 22 3E 3C 43 6F 6D  6D 61 6E 64 20 6E 61 6D
 65 3D 22 43 4F 4D 4D 41  4E 44 5F 4E 41 4D 45 22
 20 69 64 3D 22 69 64 65  6E 74 69 66 69 65 72 22
 3E 3C 44 61 74 61 3E 3C  58 6D 6C 3E 31 30 30 26
 67 74 3B 38 37 3C 2F 58  6D 6C 3E 3C 47 65 72 3E
 C3 9C 3C 2F 47 65 72 3E  3C 43 6E 3E E4 B8 AD 3C
 2F 43 6E 3E 3C 2F 44 61  74 61 3E 3C 2F 43 6F 6D
 6D 61 6E 64 3E 3C 2F 52  45 41 2D 50 49 3E 3C 2F
 52 45 41 2D 4A 45 54 3E
```

Avoid trailing characters!

Remark, in the previous example the last characters are '**</REA-JET>**', not followed by anything else.

In some cases the content is trailed by allowed characters like space, line feed or CR. In this case the characters are part of the content and do influence the content length!

4 Communication with the REA-PI

The communication with REA-PI is separated in three parts:

- Handshake, Protocol and Version selection
- Login and Logoff, user administration
- Communication with device

4.1 Handshake

After the client opens a connection to the server, the server automatically sends the supported protocols in ASCII mode. After the client has chosen to use REA-PI all following communication has to be done in XML using the message frame described in the next chapter, until the TCP connection is closed. The character encoding is always UTF-8.

The server now sends the „Welcome“-Message, which offers the supported versions of REA-PI. In response the client chooses one of these by sending the ‘VersionSelect’ message and the server confirms the clients request with sending a ‘VersionSelection’ message back or denies the connection by closing it without sending any further responses. This communication is strictly alternating. It does not work if the client sends the version selection immediately after the protocol selection. The client has to respond to a server command.

If the client does something unexpected during handshake, the server immediately closes the connection. In addition, the server may close the connection after a timeout of a few seconds.

	Server	**Client**
1.	TCP/IP Listening…	
2.		TCP/IP Connection atempt …
3.	`[STX]REA-PI;REA-PLC[ETX][LF]`	
4.		`[STX]REA-PI[ETX][LF]`

5.	`<?xml version="1.0" encoding="UTF-8"?>`
	`<REA-JET>`
	`  <REA-PI>`
	`  <Welcome>`
	`  <Version>1.8</Version>`
	`  <Version>2.0</Version>`
	`  <Version>2.1</Version>`
	`  ...`
	`  <Version>3.9</Version>`
	`  </Welcome>`
	`  </REA-PI>`
	`</REA-JET>`	

6.		`<REA-JET>`
		`  <REA-PI>`
		`  <VersionSelect>`
		`  <Version>3.9</Version>`
		`</VersionSelect>`
		`  </REA-PI>`
		`</REA-JET>`

7.	`<?xml version="1.0" encoding="UTF-8"?>`
	`<REA-JET>`
	`  <REA-PI>`
	`  <VersionSelection>`
	`  <Version>3.9</Version>`
	`  </VersionSelection>`
	`  </REA-PI>`
	`</REA-JET>`	

4.2 Communication

The device is now available for further communication.

4.2.1 Protocol commands

Clients can send commands to the server. The body of every REA-PI commands looks like this:

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="COMMAND NAME" id="identifier">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

The root node `<REA-JET>` indicates a REA XML structure (besides communication protocol also the labels and the jobs are defined in XML). The only sub node is `<REA-PI>`, which may contain multiple nodes `<Command>` (and one `<Status>` element in case of a reply, see below).

Every command has an attribute '`name`' for the name of the command, e.g. SUBSCRIBE, SETJOB or GETDATETIME. The client may pass a command identifier attribute '`id`', which can be used as a sequence number and is returned with the response to the command. In this way the client should always be able to get the context of a response.

The `<Data>` element is a generic element, which is redefined for every command (distinguished by the name attribute).

The server will reply to every command. A reply looks like a command, but it has an additional node `<Status>`, which contains the information whether a command succeeded or not:

```
<REA-JET>
   <REA-PI>
      <Command name="COMMAND NAME" id="identifier">
         <Data>
            ...
         </Data>
      </Command>
      <Status>
         <Domain>99</Domain>
         <Code>0</Code>
         <Message>... successful</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

An external client can interpret `<Code>0</Code>` as successful execution of a command independent from the domain `<Domain>nn</Domain>` and all other values as failure of execution. The specific meanings of the codes are listed in "Appendix A1: Error Codes".

4.2.2 Protocol events

REA-PI events can be seen as responses without a request.

For example the REA-PI forwards events of the print engine like ‘PRINTSTART’, ‘PRINTEND’ and the event ‘GETCARTRIDGES’ which is raised by cartridges to inform about changes in their state.
Usually REA-PI events have to be subscribed to in order to receive them or unsubscribed from when they are no longer wanted. But some events are raised automatically without subscription. An example for such an event is ‘GETCARTRIDGES’. The exact behavior depends on the event and is explained later in this document.

4.2.2.1 Publish / Subscribe events

There are several events which can be subscribed by a client. Subscriptions are ordinary commands with request and response. The following example shows the subscription of the event ‘PRINTSTART’, which is raised when an actual print of a label starts.

Request:

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="3169">
         <Data>
            <EVENT>PRINTSTART</EVENT>
            <EVENT_ARGS name="JobID">0</EVENT_ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The <Data> node of the command SUBSCRIBE contains in <EVENT> the event name, to which the client subscribes. The nodes <EVENT_ARGS> contain additional arguments with name and value. The name must be specified as attribute ‘name’, the value can be specified as node value or as xml attribute ‘value’.

```
         <EVENT_ARGS name="Identifier" value=”MyEvent" />
```

With named arguments it is possible to declare more than one event argument.

For example the command above subscribes to a PRINTSTART event with one named argument ‘JobID’ with value ‘0’.

Response:

```
<REA-JET>
   <REA-PI>
      <Command name="SUBSCRIBE" id="3169">
         <Data>
            <EVENT>PRINTSTART</EVENT>
         </Data>
      </Command>
      <Status>
         <Domain>400</Domain>
         <Code>0</Code>
         <Message>subscribtion successful done</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

4.2.2.2 Event notification

An event looks similar to a response to a command:

```
<REA-JET>
   <REA-PI>
      <Command name="PRINTSTART" id="">
         <Data>
            <Job>0</Job>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The main difference is the absence of an identifier in the <Command> node.

Events are used as a possibility for the server to push information to its clients. Therefore there is no need for the clients the poll for state changes.

```
<REA-JET>
   <REA-PI>
      <Command name="GETCARTRIDGES" id="47">
         <Data>
            <CARTRIDGE>
               <PRINTHEAD ID>0</PRINTHEAD ID>
               <NAME>REA Dye black</NAME>
               <InkID>054950100</InkID>
               …
            </CARTRIDGE>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

For more detailed descriptions on automatic events see chapters below.

4.2.2.3 Unsubscribe events

The subscription of every event will expired and unsubscribed automatically if the connection to the device will be closed, but it is recommended to unsubscribe to each event, which has been subscribed.

Request:

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="3169">
         <Data>
            <EVENT>PRINTSTART</EVENT>
            <EVENT ARGS name="0">0</EVENT ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

Response:

```
<REA-JET>
   <REA-PI>
      <Command name="UNSUBSCRIBE" id="3169">
         <Data>
            <EVENT>PRINTSTART</EVENT>
         </Data>
      </Command>
      <Status>
         <Domain>400</Domain>
         <Code>0</Code>
         <Message>subscribtion successful done</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The description of command UNSUBSCRIBE is similar to SUBSCRIBE.

Some events depend on preconditions and existing subscriptions are automatically unsubscribed. For example the event PRINTSTART depends on a loaded job and the subscription has to be renewed after a job on the device has changed.

4.2.3 Get Response / Event Error Values

Responses and notifications contain a <Status> node with information about the success on a command.

```
<REA-JET>
   <REA-PI>
      <Command name="COMMAND NAME" id="identifier">
         <Data>
            ...
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>10</Code>
         <Message>cannot replace a running job</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The Parameter <Status> contains the error code for the reasons of abortion. Do not mix the Node <Status> for the event success with the Event <Data> for reasons of abortion!

The values of domain, error codes and warning are described in chapter Appendix A1: Error Codes

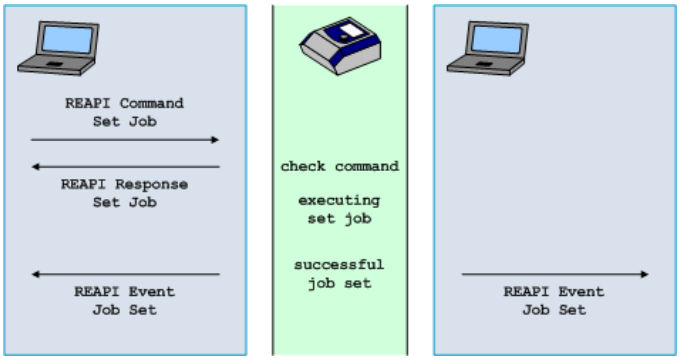
5 Job management commands

5.1 Assigning Jobs

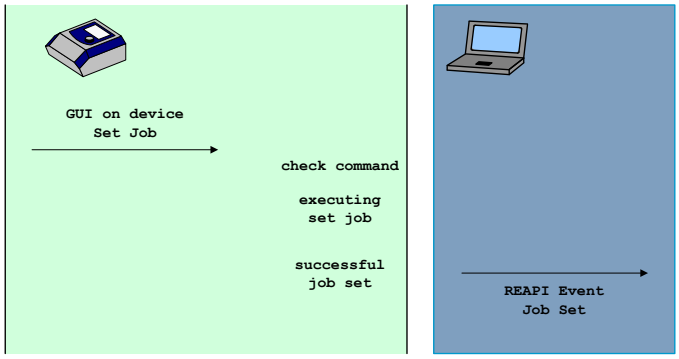
Assigning a job denotes the procedure, which is started by issuing the command for setting a job, i.e., processes the label to be printed w.r.t. the given settings. Subsequently, the print environment is configured such that printing can be activated. Any previously assigned job will automatically be removed when a new job is set. Note, that a job can be removed manually, too, by the command for clearing a job.

The event, when a (new) job is set, is called Job-Set Event and contains the name (filename) of this job as a parameter. Note, that this kind of event will also occur when a job was cleared. In that case an empty filename will be returned. No job will be assigned then.

In order to always receive the most recent data, it is strongly recommended to subscribe the Job-Set Event immediately after a connection was established. The subscription of the event can either be terminated by calling the unsubscribe function or will be terminated automatically when the network connection is closed.



When setting or clearing a job, the response from the device only indicates whether the command has been accepted. The response MUST NOT be used to determine which job is currently assigned. Since a print job could also be assigned by another application or the GUI, only the Job-Set Event gives relevant information about the correct state of the job assignment.



5.1.1 Command SETJOB

Request the device to load the specified job file and prepare settings and label(s) for printing. On success, all but the JOBSET event will have been unsubscribed and the currently assigned job will have been replaced by the new one.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SETJOB" id="208">
         <Data>
            <Job id="0"></Job>
            <Filename>demojob 2ph.job</Filename>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Job identifier of job to be set. The specification of the job id is optional; if not assigned; the first or only job is used. 0 for first job.
Note: All versions of device firmware up to current version support only one job

`Filename` Filename of job definition file.

The job file has to exist in the “rea-jet/job” folder on the device share. The filename must only consist of the underscore “_”, numeric characters and lower case letters of the “Basic Latin” Unicode block.

The response from the device indicates whether the operation was accepted, verified and started successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="SETJOB" id="208">
         <Data>
            <Job id="0"/>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>job successful activated</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Repetition of the job identifier, 0 for first job.
Note: All versions of device firmware up to current version support only one job

In the node <Status> an error code is specified whether the command is successful or not. Typical errors include a missing job, label or setting file or a mismatch of the device type the setting was created for. Furthermore, an error will be raised, when a job is tried to be assigned while the print system is started.

5.1.2 Command SETJOBANDDATA

With the command, the device is requested to load the specified job file and to prepare settings and labels for printing. In addition, initial data for variable content as well as properties of label objects and contents can be set.

This command enables the similar command SETJOB and the commands SETLABELCONTENT and SETOBJECTPROPERTIES to be combined.

If successful, the currently assigned job was replaced by the new one and all REAPI events except for the JOBSET event were unsubscribed.

This command is available with a REAPI Version ≥ 3.9.

```
<REA-JET>
    <REA-PI version="3.9">
      <Command name="SETJOBANDDATA" id="208">
        <Data>
          <Job id="0"/>
          <Filename>demojob_2ph.job</Filename>
          <Entries>
            <Entry>
              <Reference>
                <Group>Front</Group>
                <Object>article</Object>
                <Content>text 1</Content>
              </Reference>
              <Value>803.124.0038</Value>
            </Entry>
            <Entry>
              <Reference>
                <Job>0</Job>
                <Group>Front</Group>
                <Object>article</Object>
                <Property>Position/X@value</Property>
              </Reference>
              <Value>2</Value>
            </Entry>
          </Entries>
        </Data>
      </Command>
    </REA-PI>
  </REA-JET>
```

The node `<Data>` specifies the parameter:

`Job` Job identifier of job to be set. The specification of the job id is optional; if not assigned; the first or only job is used. 0 for first job.

	All versions of device firmware up to current version support only one job

`Filename` Filename of job definition file.

The job file has to exist in the “rea-jet/job” folder on the device share. The filename must only consist of the underscore “_”, numeric characters and lower case letters of the “Basic Latin” Unicode block.

`Entries` The <Data> node of an REA-PI command can contain one <Entries> with one or more <Entry> for set label contents, label properties and label content properties.

All label properties consist of a reference to that property and a value. All reference will have a reference to a job and a group in that job which represents the label.

So <Reference> defines which property or content should be updated.
<Job> indicates the job identifier of the affected job, 0 for first job. Note: All versions of device firmware up to current version support only one job.

<Group> indicates which group and implicitly which label has to be updated. Please refer to the operating manual of your device for further information on how to create and modify installation settings.

`<Object>` and `<Content>` reference the label content entity as used in command to set label contents:

```
<Entry>
    <Reference>
      <Job>0</Job>
      <Group>Front</Group>
      <Object>article</Object>
      <Content>text 1</Content>
    </Reference>
    <Value>803.123.0035</Value>
  </Entry>
```

`<Property>` is not needed and must not be declared for label contents!

`<Object>` and `<Property>` reference the label object property:

```
  <Entry>
    <Reference>
      <Job>0</Job>
      <Group>Front</Group>
      <Object>article</Object>
      <Property>Position/X@value</Property>
    </Reference>
    <Value>1.00</Value>
  </Entry>
```

`<Content>` is not needed and must not be declared for label object properties!

`<Object>`, `<Content>` and `<Property>` reference the label object content property:

```
  <Entry>
    <Reference>
      <Job>0</Job>
      <Group>Front</Group>
      <Object>serial</Object>
      <Content>counter 1</Content>
      <Property>Counter/StartValue@value</Property>
    </Reference>
    <Value>100</Value>
  </Entry>
```

The response from the device indicates whether the operation was accepted, verified and started successfully:

```
  <REA-JET>
    <REA-PI>
      <Command name="SETJOBANDDATA" id="208">
        <Data>
          <Job id="0"/>
        </Data>
      </Command>
      <Status>
        <Domain>102</Domain>
        <Code>0</Code>
        <Message>job with data successful activated</Message>
      </Status>
    </REA-PI>
  </REA-JET>
```

The node `<Data>` specifies the parameter:

`Job` Repetition of the job identifier, 0 for first job.

	All versions of device firmware up to current version support only one job.

In the node `<Status>` an error code is specified whether the command is successful or not. Typical errors include a missing job, label or setting file or a mismatch of the device type the setting was created for. Furthermore, an error will be raised, when a job is tried to be assigned while the print system is started.

5.1.3 Command CLEARJOB

The command requests the device to unload / clear the job specified by the job identifier. On success, all but the JOBSET event will have been unsubscribed and the currently assigned job will have been cleared.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="CLEARJOB" id="3233">
         <Data>
            <Job id="0"></Job>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Job identifier of job to be cleared. The specification of the job id is optional; if not assigned; the first or only job is used. 0 for first job.
Note: All versions of device firmware up to current version support only one job

	Many other REA-PI functions cannot be used when no job is assigned.

The response from the device indicates whether the operation was accepted and started successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="CLEARJOB" id="3233">
         <Data>
            <Job id="0"/>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>Job successful unassigned</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Repetition of the job identifier, 0 for first job.

	All versions of device firmware up to current version support only one job

In node <Status> an error code is specified if the command is not successful. A typical error could be that the print system has not been stopped before.

5.1.4 Event JOBSET

The device sends this event when a new job was assigned by device or any client including own call.

This event requires a subscription and has event parameters. First time the event will occur immediately after subscription to inform which set is set.

This event is available since firmware version ≥ 1.80.

5.1.4.1 Subscription

The subscription is effective while a connection. It is recommended to subscribe to this event immediately after a connection to a device is established.

The subscription of the event can either be terminated by calling the unsubscribe function or will be terminated automatically when the network connection is closed.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="164">
         <Data>
            <EVENT>JOBSET</EVENT>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>

<REA-JET>
   <REA-PI>
      <Command name="SUBSCRIBE" id="164">
         <Data>
            <EVENT>JOBSET</EVENT>
         </Data>
      </Command>
      <Status>
         <Domain>400</Domain>
         <Code>0</Code>
         <Message>subscribtion successful done</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the subscription for

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="3244">
         <Data>
            <EVENT>JOBSET</EVENT>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the unsubscription for

5.1.4.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="JOBSET" id="">
         <Data>
            <Job>0</Job>
            <Filename>demojob_1ph.job</Filename>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Get the identifier of the currently set job or the job which is now empty, 0 for the first job.
Note: All versions of device firmware up to current version support only one job.

`Filename` job which is now actually set. The parameter is empty if the previously set job is cleared.

Notice that the device will not send multiple JOBSET events as long as the job state has not been changed.

5.2 Starting and stopping jobs

The print environment is activated by issuing the command to start a job. Subsequently, the system becomes ready to print. The print heads and the laser tube, respectively, will be turned on according to the settings. Furthermore, the activities of the product sensor and the encoder can be observed. Calling the command to stop a job causes the print environment to be deactivated.

When starting or stopping a job a job, the response from the device only indicates whether the command has been accepted. The response MUST NOT be used to determine the current status of an activated job. Since a print job could also be started / stopped by another application or the GUI, only the corresponding events give relevant information about the correct job status.

5.2.1 Command STARTJOB

The command starts the currently assigned job and printing will be enabled.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="STARTJOB" id="4103">
         <Data>
            <Job id="0"></Job>
            <ReloadJob>False</ReloadJob>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Job identifier of job to be started. The specification of the job id is optional; if not assigned; the first or only job is used. 0 for first job. *Note: All versions of device firmware up to current version support only one job*
`ReloadJob`	If this optional parameter is 'True', the existing job will be reloaded from its file upon start. If not specified or 'False', no reload will take place. This parameter is available with a Firmware-Version ≥ 2.0.

Many configuration changing REA-PI functions can not be used while an assigned job is active / started.

The response from the device indicates whether the operation was accepted and started successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="STARTJOB" id="4103">
         <Data/>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>job sucessfully started</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Repetition of the job identifier, 0 for first job. *Note: All versions of device firmware up to current version support only one job*

In node <Status> an errorcode is specified if the command is not successful. A typical error could be that the print system is already started.

5.2.2 Command STOPJOB

The following command stops the currently assigned job on device and printing will be disabled.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="STOPJOB" id="4105">
         <Data>
            <Job id="0"></Job>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Job identifier of job to be stopped. The specification of the job id is optional; if not assigned; the first or only job is used. 0 for first job.
Note: All versions of device firmware up to current version support only one job

Many REA-PI functions which effect the configuration cannot be used while a job is started.

The response from the device indicates whether the operation was accepted and started successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="STOPJOB" id="4105">
         <Data>
            <Job id="0"/>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>job sucessfully stopped</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Repetition of the job identifier, 0 for first job.
Note: All versions of device firmware up to current version support only one job

In the node <Status> an error code is specified if the command is not successful. A typical error could be that the print system is already stopped.

5.2.3 Events JOBSTARTED and JOBSTOPPED

The device raises these events when a job was started and stopped, respectively, by any client. This means, an application may receive a JOBSTOPPED event when it calls STOPJOB itself, as well as when the device stopped the job in case of an error condition.

Both events work within the context of the job. Therefore, these events should be subscribed immediately after a job has been set. The recommended way is to call the subscription when the JOBSET event was received to ensure reading the most recent status. The subscription of the events can be cancelled by either calling the corresponding unsubscribe functions or is terminated automatically when the job is removed or replaced by another job.

Subscribing to the events when a job was started or stopped, respectively, allows a client to be informed about the current state of the job activation. With help of the parameter of these events you can determine which job was started or stopped.

These events are available since firmware version ≥ 1.80.

5.2.4 Subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to these events when the event of a new job was set has been received.

First time the events will occur immediately after subscription to inform the set job is started or stopped.

Unsubscribe the events if they are no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="3266">
         <Data>
            <EVENT>JOBSTARTED</EVENT>
            <EVENT_ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>

<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="3267">
         <Data>
            <EVENT>JOBSTOPPED</EVENT>
            <EVENT_ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="3266">
         <Data>
            <EVENT>JOBSTARTED</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>

<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="3267">
         <Data>
            <EVENT>JOBSTOPPED</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the unsubscription for.
`Job`	Identifier of the job which will not longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

5.2.4.1 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="JOBSTARTED" id="">
         <Data>
            <Job>0</Job>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>

<REA-JET>
   <REA-PI>
      <Command name="JOBSTOPPED" id="">
         <Data>
            <Job>0</Job>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Get the identifier of the job which was started or stopped, 0 for the first job.
Note: All versions of device firmware up to current version support only one job.

5.3 Purging a printhead

5.3.1 Command STARTPURGE

The command requests the device to purge the specified printheads with corresponding parameters.

In order to purge a job must be loaded before. Only heads which are included in this job are able to be purged. It is only available in stopped print mode and not during active printing for that job.

Since Firmware-Version ≥ 3.2 the device will send a corresponding event when the process to purge printheads is finished.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="STARTPURGE" id="3647">
         <Data>
            <Printhead id="0">
               <DropPulses>1200</DropPulses>
               <Frequency>2200</Frequency>
            </Printhead>
            <Printhead id="1">
               <DropPulses>1200</DropPulses>
               <Frequency>2200</Frequency>
            </Printhead>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="STARTPURGE" id="3647">
         <Data>
            <Printhead id="-1">
               <DropPulses>1200</DropPulses>
               <Frequency>2200</Frequency>
            </Printhead>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Printhead` The heads to be purged are specified by id, which 0 is first head, 1 second head and so on. For more comfortable usage it is possible to specify only one printhead with a special identifier -1. In this case all available printheads in job 0 are purged with this parameter.

	Pay attention: it is not possible to mix both versions with exact specified printheads and the wildcard identifier -1!

`Droppulses` Number of drops to be spitted, allowed value from 500 - 6000 drops.

`Frequency` Frequency of drops in Hz, allowed value 1000-6000Hz

The response from the device indicates whether the operation was accepted, validated and started successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="STARTPURGE" id="3647">
         <Data/>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>purge sucessfully executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter.

In node <Status> an errorcode is specified if the command is not successful.

Typical errors could be that the requested printheads are not included in the loaded job or the print system is started.

5.3.2 Event PURGECOMPLETED

The device always raises this event whenever the purge process is finished. However, this event needs no subscription.

This event is available with a Firmware-Version ≥ 3.2.

```
<REA-JET>
   <REA-PI>
      <Command name="PURGECOMPLETED" id="">
         <Data>
            <Status>completed</Status>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Status` Status of purge, always completed.

6 Print management commands

6.1 Print control events and functions

The functionality described in this section mainly concerns events which occur during the printing process, either as notifications for successful operation or as error indications and warnings respectively.

The former comprise events when a print was triggered, was actually started or completed or when certain print phases are passed through:

- PRINTTRIGGER: A print was triggered.
- PRINTSTART: A print / print sequence was started.
- PRINTEND: A print / print sequence was completed.
- IMAGESTART: An individual print within a whole sequence was started.
- IMAGEEND: An individual print within a whole sequence was completed.

The following events belong to the latter group:

- MISSINGCONTENT: Upon trigger, no valid print buffer was available.
- PRINTREJECTED: No print was performed as there was no valid print buffer available.
- PRINTSPEEDERROR: Product speed too high w.r.t. maxmimum nozzle frequency and horizontal resolution.
- PRINTABORTED: A print sequence was interrupted.
- PRINTSTATUS: Miscellaneous errors / warnings.

All these events are automatically unsubscribed when a new job is assigned or the current job is cleared.

6.1.1 Event PRINTTRIGGER

The device raises this event when the trigger for a print has occurred. Normally it is caused by product sensor or digital input or any combination of these.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 1.80.

6.1.1.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event JobSet has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTTRIGGER</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTTRIGGER</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the unsubscription for.
`Job`	Identifier of the job which will not longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

6.1.1.2 Event parameters

```
<REA-JET>
  <REA-PI>
    <Command name="PRINTTRIGGER"id="">
      <Data>
        <Job>0</Job>
        <Group>1</Group>
        <PrintTrigger>0</PrintTrigger>
        <IsTriggerGroup>true</IsTriggerGroup>
        <TS>1469089417664432458</TS>
      </Data>
    </Command>
    <Status>
      <Domain>99</Domain>
      <Code>0</Code>
      <Message></Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Get the identifier of the job for which the printtrigger was occurred, 0 for the first job. *Note: All versions of device firmware up to current version support only one job.*

6.1.2 Event PRINTSTART

The device raises this event on the beginning of a printing process. Here, the term “printing process” refers to the first print head of the first label / print head group in chronological order and denotes a sequence of one or more consecutive prints, depending on the print mode:

- Using print mode “normal”, there is only one individual print per trigger. Hence every (individual) print raises one (separate) PRINTSTART event.
- Operated in print mode “cyclic”, one PRINTSTART event is raised
  - once for the configured sequence, when “FixedMode” is configured
  - for every individual print, when “FixedMode” is not configured
- The same goes for print mode “continous”, no matter whether a limitation for the number of (individual) prints is set.

Thus, the PRINTSTART event will in general be issued when the sensor offset and label delay has elapsed after the PRINTRIGGER event. In “continous” mode this applies analogously w.r.t. the setting “limited output”.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 1.80.

6.1.2.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event JobSet has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTSTART</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTSTART</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the unsubscription for.
`Job`	Identifier of the job which will no longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

6.1.2.2 Event parameters

```
   <REA-JET>
      <REA-PI>
         <Command name="PRINTSTART" id="">
            <Data>
               <Job>0</Job>
               <Group>1</Group>
               <PrintTrigger>0</PrintTrigger>
               <IsTriggerGroup>true</IsTriggerGroup>
               <TS>14690894176644324458</TS>
            </Data>
         </Command>
         <Status>
            <Domain>102</Domain>
            <Code>0</Code>
            <Message>successful executed</Message>
         </Status>
      </REA-PI>
   </REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Get the identifier of the job for which the printstart event was occurred, 0 for the first job. *Note: All versions of device firmware up to current version support only one job.*

At that time, the bitmap for the current print of all groups on this product is determined and hence the client can send the next set of data without scrambling the current printout (see also SETLABELCONTENT).

6.1.3 Event PRINTEND

The device raises this event when the printing process for the current label is finished, i.e., the last print head of the last print head group has completed the last of a sequence of one or more consecutive prints.

See also PRINTSTART

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 1.80.

6.1.3.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event JobSet has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTEND</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTEND</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the unsubscription for.
`Job`	Identifier of the job which will no longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

6.1.3.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="PRINTEND" id="">
         <Data>
            <Job>0</Job>
            <Group>1</Group>
            <PrintTrigger>0</PrintTrigger>
            <IsTriggerGroup>true</IsTriggerGroup>
            <TS>14690894176644324458</TS>
            <Duration unit="ms">36.966</Duration>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

Job	Get the identifier of the job for which the printend event was occurred, 0 for the first job. *Note: All versions of device firmware up to current version support only one job.*
Duration	Returns the time duration from the beginning to the end of the printing process for the currently assigned label

6.1.4 Event PRINTABORTED

The device raises this event if the print of a label was aborted. Normally this event occurred in continuous printing mode sequence. It is the equivalent of the PRINTEND event for this special case.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 1.80.

6.1.4.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event JobSet has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTABORTED</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTABORTED</EVENT>
            <EVENT_ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the unsubscription for.
`Job`	Identifier of the job which will no longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

6.1.4.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="PRINTABORTED" id="">
         <Data>
            <Job>0</Job>
            <Group>1</Group>
            <PrintTrigger>0</PrintTrigger>
            <IsTriggerGroup>true</IsTriggerGroup>
            <TS>14690894176644324458</TS>
            <Duration unit="ms">12.454</Duration>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Get the identifier of the job for which the printaborted event was occurred, 0 for the first job.
Note: All versions of device firmware up to current version support only one job.

This event occurs as counterpart to PRINTEND for a not properly completed printing. However this event only can occur in continuous print mode!

6.1.5 Event IMAGESTART

The device raises an event PRINTSTART in the beginning of a printing process and IMAGESTART at every individual print. Operated in print mode “cyclic” or “limited output”, IMAGESTART events are raised for every individual print, when “FixedMode” is configured.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 3.4

6.1.5.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event JOBSET has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>IMAGESTART</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTSTART</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

Event	Name of the event to do the unsubscription for.
Job	Identifier of the job which will not longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

6.1.5.2 Event parameters

```
   <REA-JET>
      <REA-PI>
         <Command name="IMAGESTART" id="">
            <Data>
               <Job>0</Job>
               <Group>1</Group>
               <PrintTrigger>0</PrintTrigger>
               <IsTriggerGroup>true</IsTriggerGroup>
               <TS>14690894176644324458</TS>
            </Data>
         </Command>
         <Status>
            <Domain>102</Domain>
            <Code>0</Code>
            <Message>successful executed</Message>
         </Status>
      </REA-PI>
   </REA-JET>
```

The node <Data> specifies the parameter:

Job	Get the identifier of the job for which the imagestart event was occurred, 0 for the first job. *Note: All versions of device firmware up to current version support only one job.*

6.1.6 Event IMAGEEND

The device raises this event when the printing process for the current label is finished, i.e., the last print head of the last print head group has completed the last of a sequence of one or more consecutive prints.

See also IMAGESTART and PRINSTART.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 3.40

6.1.6.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event JOBSET has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>IMAGEEND</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the subscription for

`Job` Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>IMAGEEND</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the unsubscription for.

`Job` Identifier of the job which will no longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

6.1.6.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="IMAGEEND" id="">
         <Data>
            <Job>0</Job>
            <Group>1</Group>
            <PrintTrigger>0</PrintTrigger>
            <IsTriggerGroup>true</IsTriggerGroup>
            <TS>14690894176644324458</TS>
            <Duration unit="ms">36.734</Duration>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Get the identifier of the job for which the imageend event was occurred, 0 for the first job. *Note: All versions of device firmware up to current version support only one job.*
`Duration`	Returns the time duration from the beginning to the end of the printing process for the currently assigned label

6.1.7 Event PRINTSPEEDERROR

The device raises this event when the product speed is too high, e.g., w.r.t. maxmimum nozzle frequency and horizontal resolution.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 1.80.

6.1.7.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event JobSet has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTSPEEDERROR</EVENT>
            <EVENT_ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the subscription for

`Job` Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTSPEEDERROR</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the unsubscription for.

`Job` Identifier of the job which will no longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

6.1.7.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="PRINTSPEEDERROR" id="">
         <Data>
            <Job>0</Job>
            <Status>
               <Domain>102</Domain>
               <Code>70</Code>
               <Message>print speed too fast, max print pulses exceeded</Message>
            </Status>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Get the identifier of the job for which the printspeederror event was occurred, 0 for the first job.
Note: All versions of device firmware up to current version support only one job.

`Status` Get the error information why this speed error occurs. For more information about errorcodes see chapter 18 Appendix A1: Error Codes, page 128.
Possible included Errors are:
* DOMAIN_ENGINE_PRINTCONTROL, CODE_SPEEDSENSORPULSELOST
* DOMAIN_ENGINE_PRINTCONTROL, CODE_SPEEDSENSORTOOFAST,
* DOMAIN_ENGINE_PRINTCONTROL, CODE_SPEEDSENSORSLICESLOST

6.1.8 Event PRINTSTATUS

The device raises this event whenever an unexpected error or warning or information on current printing occurs. However this event needs no subscription and can occur as long as a connection to the device is established.

This event is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI>
      <Command name="GETPRINTSTATUS" id="">
         <Data>
            <Job>0</Job>
         </Data>
      </Command>
      <Status>
         <Domain>300</Domain>
         <Code>12</Code>
         <Message>bolting lock not set</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Get the identifier of the job for which the printspeederror event was occurred, 0 for the first job. *Note: All versions of device firmware up to current version support only one job*.

In node <Status> the errorcode or warning code is specified. For more information about errorcodes see chapter 18 Appendix A1: Error Codes, page 128.

6.1.9 Event MISSINGCONTENT

The device raises this event when a print is triggered but no print buffer is available. This event signals the beginning of a period, in which a new printbuffer may become available and would eventually be printed. This period would end either with the event PRINTREJECTED if no print buffer was available at all or with the event PRINTSTART if it was possible to create a print buffer created in sufficient time.

For configuration of the behavior of the events MISSINGCONTENT and PRINTREJECTED see the chapter 6.2 Print Rejection and “No Print Buffer , page 53.

This event requires a subscription and has event parameters

This command is available with a Firmware-Version ≥ 3.30.

6.1.9.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event JobSet has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>MISSINGCONTENT</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>MISSINGCONTENT</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the unsubscription for.

`Job` Identifier of the job which won't be observed any longer. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

6.1.9.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="MISSINGCONTENT" id="">
         <Data>
            <Job>0</Job>
            <Group>1</Group>
            <PrintTrigger>0</PrintTrigger>
            <IsTriggerGroup>true</IsTriggerGroup>
            <TS>14690894176644324458</TS>
            <Status>
               <Domain>102</Domain>
               <Code>32</Code>
               <Message>At least one group doesn't have a valid printbuffer image</Message>
            </Status>
            <NoBufferMode>2</NoBufferMode>
            <NoBufferOffset>50.0000</NoBufferOffset>
            <PrintRejectMode>2</PrintRejectMode>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the event parameter:

`Job` Get the identifier of the job for which the missingcontent event was occurred, 0 for the first job.
Note: All versions of device firmware up to current version support only one job.

`Status` Get the error information why this missingcontent error occurs. For more information about errorcodes see chapter 18 Appendix A1: Error Codes, page 128.

`Printtrigger` Gets the identifier of the affected print trigger. For more information about this parameter see chapter 6.2 Print Rejection and "No Print Buffer , page 53.

`NoPrintBufferMode` Gets the information about the mode if no printbuffer is available. For more information about this parameter see chapter 6.2 Print Rejection and "No Print Buffer , page 53.

`NoBufferOffset` Gets the delay in millimeter for which the printer is waiting for a valid printbuffer. For more information about this parameter see chapter 6.2 Print Rejection and "No Print Buffer , page 53.

`PrintRejectMode` Get the mode for reaction if a print is finally rejected. For more information about this parameter see chapter 6.2 Print Rejection and "No Print Buffer , page 53.

6.1.10 Event PRINTREJECTED

The device raises this event when the actual printing on a product is rejected because no print buffer was available. This event signals the end of a time span, in which a new print buffer would be accepted and printed. This time span starts with an event MISSINGCONTENT. See also event MISSINGCONTENT.

For configuration of the behavior of the events MISSINGCONTENT and PRINTREJECTED see the chapter 6.2 Print Rejection and “No Print Buffer , page 53.

Particularly, the event PRINTREJECTED would occur in single print mode if variable data for the upcoming print was not refreshed and a possible tolerance (configurable, cf. chapter 6.2 Print Rejection and “No Print Buffer Mode”) has passed.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 1.80.

The meaning and the behavior of this event has been modified significantly since version 3.3 in comparison to older versions! See also the chapter 6.2 Print Rejection and “No Print Buffer , page 53 and Event MISSINGCONTENT, page 48.

6.1.10.1 Event subscription

The subscription is only effective while a job has been loaded. It is recommended to subscribe to this event when the event for setJobSet has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTREJECTED</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>PRINTREJECTED</EVENT>
            <EVENT ARGS name="JobID" value="0" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the unsubscription for.

`Job` Identifier of the job which won't be observed any longer. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

6.1.10.2 Event parameters

```
   <REA-JET>
      <REA-PI>
         <Command name="PRINTREJECTED" id="">
            <Data>
               <Job>0</Job>
               <Group>1</Group>
               <PrintTrigger>0</PrintTrigger>
               <IsTriggerGroup>true</IsTriggerGroup>
               <TS>14690894176644324458</TS>
               <Status>
                  <Domain>102</Domain>
                  <Code>32</Code>
                  <Message>At least one group doesn't have a valid printbuffer image</Message>
               </Status>
               <NoBufferMode>2</NoBufferMode>
               <NoBufferOffset>50.0000</NoBufferOffset>
               <PrintRejectMode>2</PrintRejectMode>
            </Data>
         </Command>
         <Status>
            <Domain>102</Domain>
            <Code>0</Code>
            <Message>successful executed</Message>
         </Status>
      </REA-PI>
   </REA-JET>
```

The node <Data> specifies the event parameter:

`Job` Get the identifier of the job for which the printrejected event was occurred, 0 for the first job.
Note: All versions of device firmware up to current version support only one job.

`Status` Get the error information why this printrejected error occurs. For more information about errorcodes see chapter 18 Appendix A1: Error Codes, page 128.

`Printtrigger` Gets the identifier of the affected print trigger. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

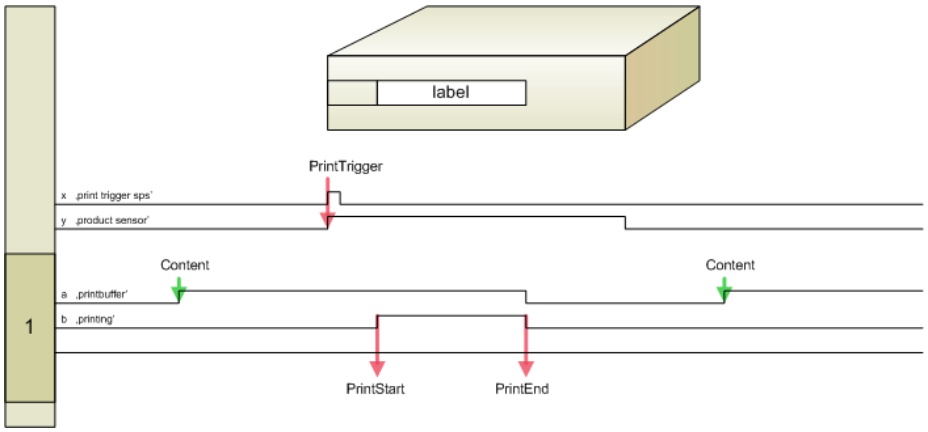
`NoPrintBufferMode` Gets the information about the mode if no printbuffer is available. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

`NoBufferOffset` Gets the delay in millimeter for which the printer is waiting for a valid printbuffer. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

`PrintRejectMode` Get the mode for reaction if a print is finally rejected. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

6.2 Print Rejection and “No Print Buffer Mode”

The following diagram depicts - in a simplified way - the sequence of some of the actions and events when executing a print. Usually, the content to be printed has been set early enough before the print is triggered. Hence the print is executed with just this content, while the events PRINTSTART and PRINTEND signal the beginning and end of the printing process respectively:



Since this represents the regular flow, the settings for the “No Print Buffer Mode” are of no relevance.

However, in case the content to be printed could not be prepared in time, the subsequent behaviour depends on the settings for NoBufferMode. In detail, there are the following possibilities:

0	Reject
1	Wait Infinite
2	Wait Delay
3	Wait Delay (incl. SO)
4	Wait Delay (incl. SO & LO)

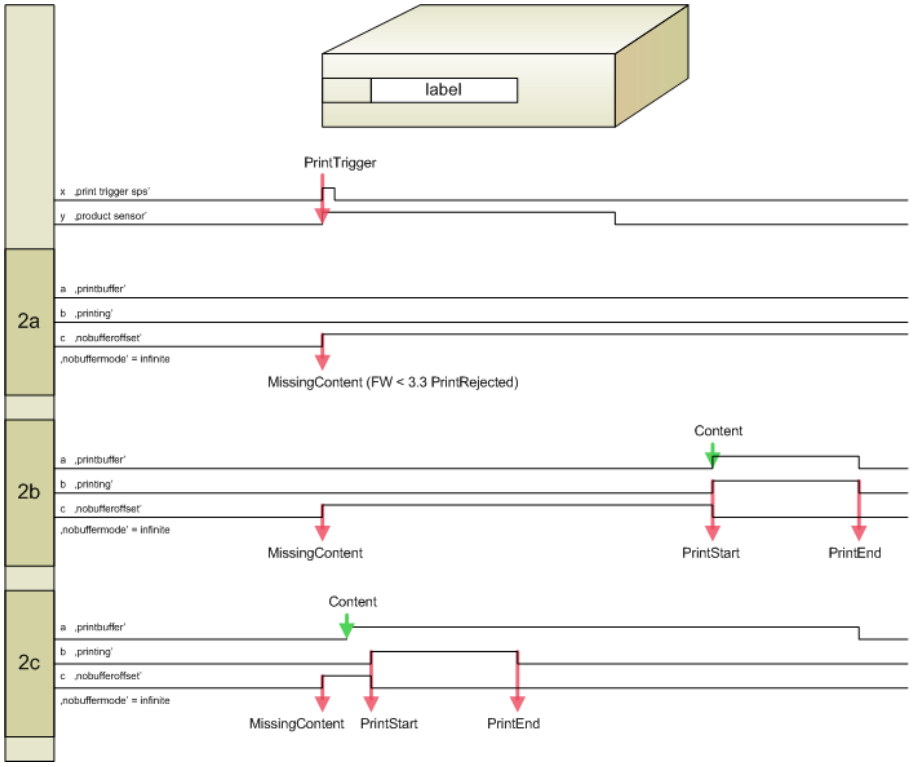
6.2.1 NoBufferMode “Wait Infinite”

If NoBufferMode is set to “Wait Infinite”, the device will directly issue an event MISSINGCONTENT upon the print trigger. Neither the parameter NoBufferOffset nor PrintRejectMode has an effect in this case.

As soon as the the content to be printed has been prepared, the print will be executed, i.e., if necessary, the device will wait indefinitely long for a content.

Therefore, a displacement may occur for the print when the content finally has arrived. The the events PRINTSTART and PRINTEND signal the beginning and end of this - possibly delayed - printing process respectively.

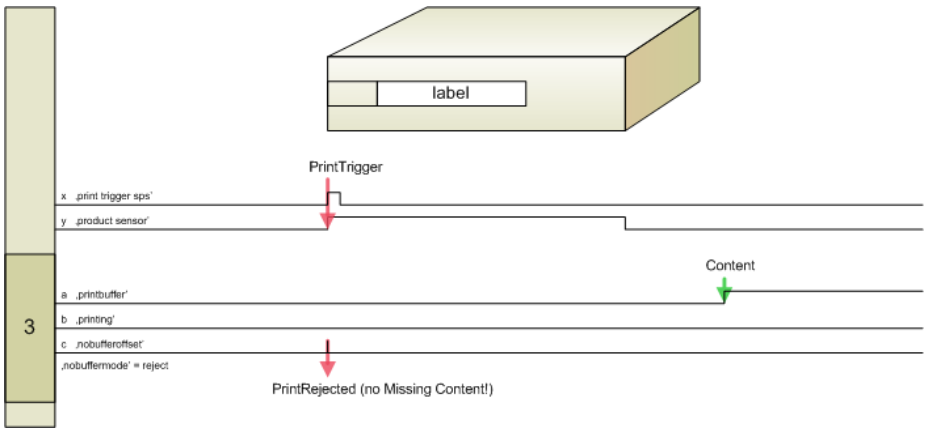
The setting described here corresponds to the behaviour of firmware version 3.25 and before with the exception that an event PRINTREJECTED will be raised instead of an event MISSINGCONTENT as described above.



6.2.2 NoBufferMode “Reject”

If NoBufferMode is set to “Reject”, the device will directly issue an event PRINTREJECTED upon the print trigger. The parameter NoBufferOffset is not applicable, but PrintRejectMode with one of the values:

0	Stop Job
1	deprecated: Wait for Confirmation
2	Ignore and Wait

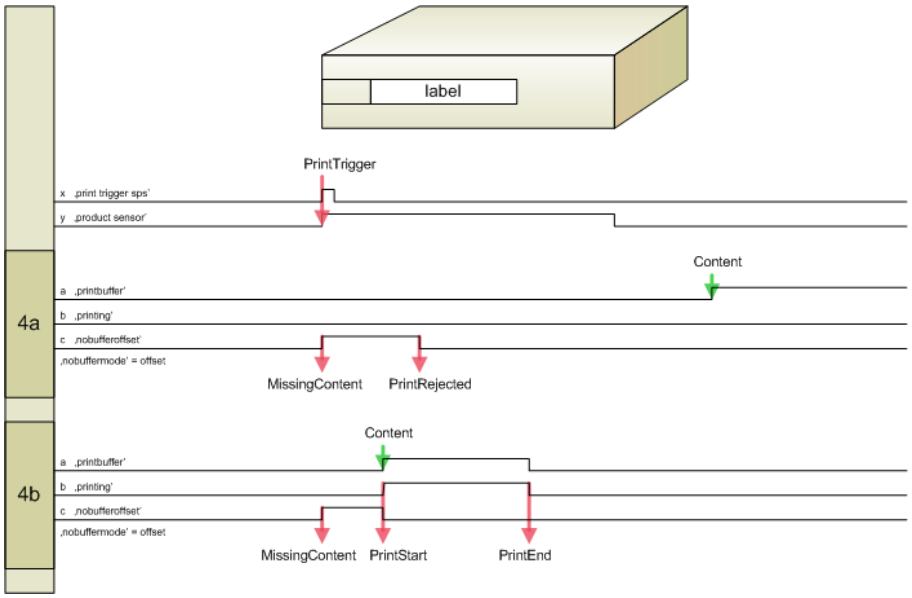


Since the content to be printed was not prepared in time, the print will not be executed. Afterwards

- the job will be stopped, when PrintRejectMode is “Stop Printing”.
- deprecated: any print trigger will be discarded until the status is explicitely confirmed (cf. Command CONFIRMPRINTREJECT), when PrintRejectMode is “Wait for Confirmation”.
- the condition is ignored and the next print trigger will result in a regular printing process, when PrintRejectMode is “Wait and Ignore”.

6.2.3 NoBufferMode „WaitDelay“

The variants “Wait Delay“,”Wait Delay (incl. SA)“ and “Wait Delay (incl. SA & LA)“, reflect some sort of combination of NoBufferMode “Wait Infinite” and NoBufferMode “Reject” respectively:



Upon print trigger an event MISSINGCONTENT will be issued immediately. The parameter NoBufferOffset specifies a tolerance (in mm) which may pass before an event PRINTREJECTED will occur.

- If the content to be printed can be prepared within the given tolerance, the print will be executed. However, a displacement will occur for the print, the events PRINTSTART and PRINTEND signal the beginning and the and of this print respectively.

If the content to be printed was not prepared within the given delay, the trigger will be discarded, i.e., no print will be executed, and an event PRINTREJECTED will be issued. The further process is identical to NoBufferMode “Reject”, i.e., depends on the setting for PrintRejectMode:

- the job will be stopped, when PrintRejectMode is “Stop Printing”.
- any print trigger will be discarded until the status is explicitely confirmed (cf. Command CONFIRMPRINTREJECT), when PrintRejectMode is “Wait for Confirmation”.
- the condition is ignored and the next print trigger will result in a regular printing process, when PrintRejectMode is “Wait and Ignore”.

6.2.4 Command GETNOPRINTBUFFERMODE

This command gets the configuration and parameters for the behavior of the printer when a print trigger instructs a print, but no print buffer is available.

This command is available with a Firmware-Version ≥ 3.30.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETNOPRINTBUFFERMODE" id="73">
         <Data>
            <PrintTrigger>0</PrintTrigger>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Printtrigger` The identifier of the affected print trigger. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

The response from the device indicates whether the operation was started successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="GETNOPRINTBUFFERMODE" id="73">
         <Data>
            <PrintTrigger>0</PrintTrigger>
            <NoBufferMode>2</NoBufferMode>
            <NoBufferOffset>50.0000</NoBufferOffset>
            <PrintRejectMode>2</PrintRejectMode>
         </Data>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message>Get parameter to handle no printbuffer event on printing successfully
retrieved!</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the returned parameter:

`Printtrigger` Repetition of the identifier of the affected print trigger. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

`NoPrintBufferMode` Gets the information about the mode if no printbuffer is available. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

`NoBufferOffset` Gets the delay in millimeter for which the printer is waiting for a valid printbuffer. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

`PrintRejectMode` Get the mode for reaction if a print is finally rejected. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

In node <Status> an errorcode is specified if the command is not successful.

6.2.5 Command SETNOPRINTBUFFERMODE

This command sets the configuration and parameters for the behavior of the printer when a print trigger instructs a print, but no print buffer is available.

This command is available with a Firmware-Version ≥ 3.30.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SETNOPRINTBUFFERMODE" id="74">
         <Data>
            <PrintTrigger>0</PrintTrigger>
            <NoBufferMode>1</NoBufferMode>
            <NoBufferOffset>50.0000</NoBufferOffset>
            <PrintRejectMode>0</PrintRejectMode>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Printtrigger` Specify the identifier of the affected print trigger. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

`NoPrintBufferMode` Set the mode when no printbuffer is available. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

`NoBufferOffset` Set the delay in millimeter for which the printer is waiting for a valid printbuffer. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

`PrintRejectMode` Set the mode for reaction if a print is finally rejected. For more information about this parameter see chapter 6.2 Print Rejection and “No Print Buffer , page 53.

The response from the device indicates whether the operation was started successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="SETNOPRINTBUFFERMODE" id="74">
         <Data/>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message>Set parameter to handle no printbuffer event on printing successfully
retrieved!</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter:

In node <Status> an errorcode is specified if the command is not successful.

6.3 Confirm PrintRejection

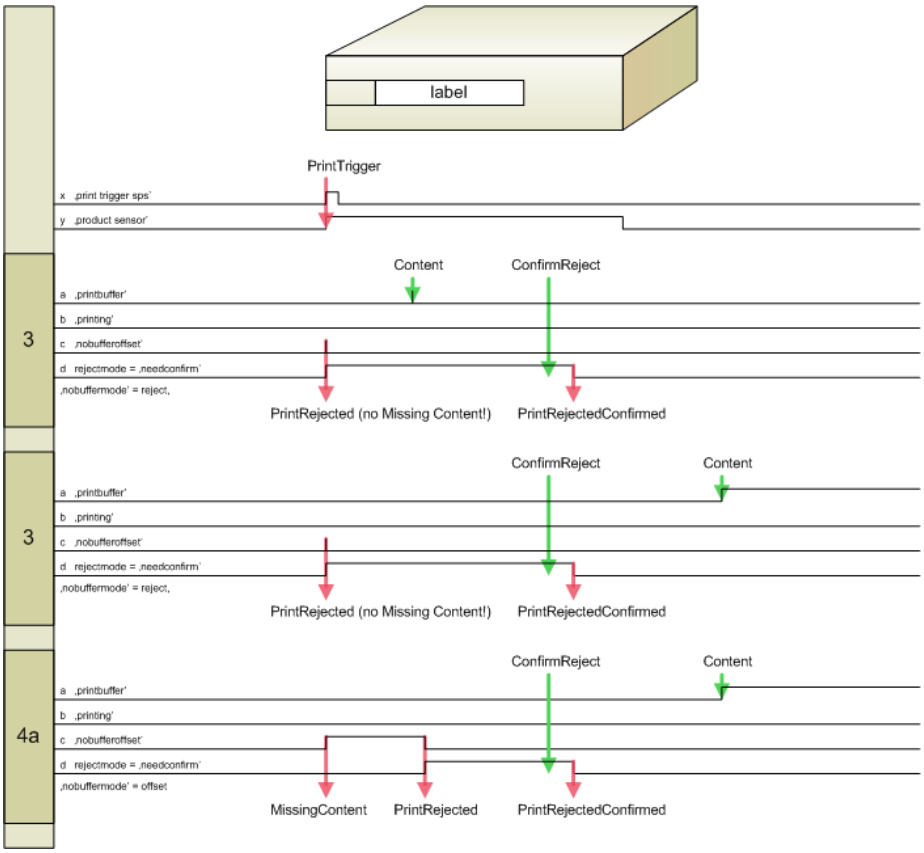
The functionality for confirmation of print rejection is deprecated since version 3.9.
This applies to the entire chapter

When PrintRejectMode is set to “Wait for Confirmation” (cf. chapter 6.2 Print Rejection and “No Print Buffer Mode”, page 53), then

- no new content will be accepted
- all of the following print triggers will be discarded

until the status is explicitely confirmed by the CONFIRMPRINTREJECT command. Any content that is currently under preparation by the print system will be discarded as well. Only after the above command a new content is allowed to be set.

Additionally, an event PRINTREJECTEDCONFIRMED may be subscribed to get informed about the new status, particularly by clients, which do not send CONFIRMPRINTREJECT themselves.



6.3.1 Command CONFIRMPRINTREJECT

The functionality for confirmation of print rejection is deprecated since version 3.9.

This command confirms the occurrence of the PRINTREJECTED event and thus terminates the dismission of further printis (see also chapter 6.2 Print Rejection and “No Print Buffer Mode”, page 53).

This command is available with a Firmware-Version ≥ 3.30.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="CONFIRMPRINTREJECT" id="242">
         <Data>
            <Job id="0"></Job>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Job identifier of the affected job. The specification of the job id is optional; if not assigned; the first or only job is used. 0 for first job.
Note: All versions of device firmware up to current version support only one job

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="CONFIRMPRINTREJECT" id="242">
         <Data>
            <Job id="0"/>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>print reject confirmed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Repetition of the identifier of the affected job, 0 for first job. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

In node <Status> an errorcode is specified if the command is not successful.

A typical error is to call this command not in scope of the event PRINTREJECTED. In this case there is nothing to confirm:

```
<REA-JET>
   <REA-PI>
      <Command name="CONFIRMPRINTREJECT" id="249">
         <Data>
            <Job id="0"/>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>39</Code>
         <Message>Device is not in confirm mode after a printrejected occured</Message>
      </Status>
```

```
      </REA-PI>
   </REA-JET>
```

6.3.2 Event PRINTREJECTEDCONFIRMED

The functionality for confirmation of print rejection is deprecated since version 3.9.

The device raises this event when a PRINTREJECTED event was confirmed.

For further information see Command CONFIRMPRINTREJECT, page 59, and chapter 6.2 Print Rejection and “No Print Buffer Mode”, page 53.

This event requires a subscription and has event parameters.

This event is available with a Firmware-Version ≥ 3.30.

6.3.2.1 Event subscription

```
   <REA-JET>
      <REA-PI version="3.9">
         <Command name="SUBSCRIBE" id="270">
            <Data>
               <EVENT>PRINTREJECTCONFIRMED</EVENT>
               <EVENT_ARGS name="0">0</EVENT_ARGS>
            </Data>
         </Command>
      </REA-PI>
   </REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the subscription for
`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

```
   <REA-JET>
      <REA-PI version="3.9">
         <Command name="UNSUBSCRIBE" id="270">
            <Data>
               <EVENT>PRINTREJECTCONFIRMED</EVENT>
               <EVENT ARGS name="0">0</EVENT ARGS>
            </Data>
         </Command>
      </REA-PI>
   </REA-JET>
```

The node <Data> specifies the parameter:

`Event`	Name of the event to do the unsubscription for.
`Job`	Identifier of the job which will not longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*

6.3.2.2 Event parameters

```
   <REA-JET>
      <REA-PI>
         <Command name="PRINTREJECTCONFIRMED" id="">
            <Data>
```

```
            <Job>0</Job>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the event parameter:

`Job` Get the identifier of the job for which the event print rejected was confirmed. 0 for the first job.
Note: All versions of device firmware up to current version support only one job.

7 Printing - Label manipulation

7.1 What are properties and variable contents

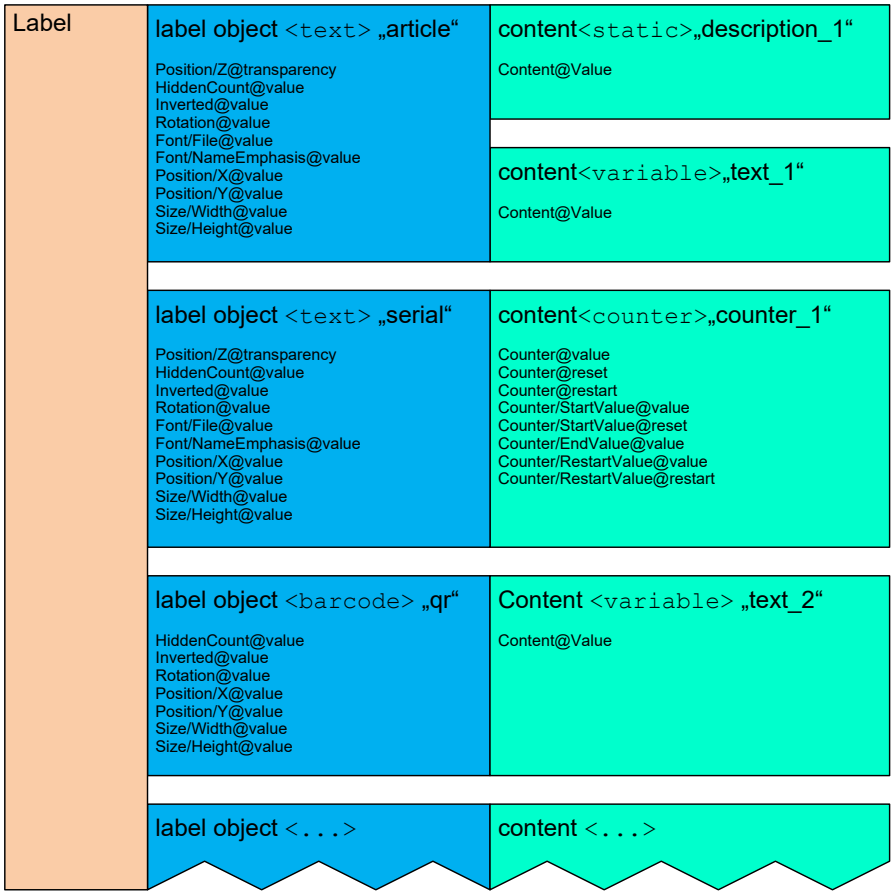
This chapter describes how to query or change label contents and label properties when a job is assigned or prints are executed.

To demonstrate label manipulating, we assume a job with one print head in a group named ‘Front’ which is printing an example label. The following example label contains three label objects ‘article’, ‘serial’ and ‘qr’, each of them with one or more contents and many label object properties, depending on the types of the label objects:

art: 803.123.0035

000001

The following diagram shows the structure of the label, label objects and label contents:



Generally there are two ways to manipulate labels during printing:

- A command to set only variable label contents of type <Variable> and <Choice> in a performant way. Since Firmware 3.50 also contents of type <ContentBuffer> are supported.
  In the example label assumed with a job in group 'Front' there are two variable contents that can be set with the command to set variable contents. This can also be done using the WebGUI:

- A command to set label object properties and content properties. This command supports also setting of variable label contents as the previous command for supporting an atomic call to set all contents and properties together.
  Setting properties and variable contents with this command is less performant than setting only variable label contents with the command mentioned above.
  In the example label assuming loaded with a job in group 'Front' there are many label object properties and variable contents that can be set with the command to set object properties and variable contents.

The current REA-PI Version supports following label object properties:

Type@value

String with literal name of the label object analog to the names used in the label, eg “Text”, “Counter”, “GS1DataMatrix”. This property is read only!

Note: Supported by devices since FW 3.40

Position/X@value

numeric float with X position in millimeter, eg “12”, “40.123”

Note: Supported by devices since FW 3.02

Position/Y@value

numeric float with Y position in millimeter, eg “12”, “40.123”

Note: Supported by devices since FW 3.02

Position/Z@transparency

boolean text, sets the transparency of an label object.

‘true’ transparent, underlying labelobjects are visible.

‘false’ opaque, not transparent, underlying labelobjects are covered.

Note: Supported by HR printing devices since FW 3.20, not supported by laser marking devices; there they are always transparent

Note: On HR printing devices supported by all types of label object, except barcodes; they are never transparent

Size/Width@value

numeric float with X dimension size in millimeter, eg “12”, “40.123”

Note: Supported by devices since FW 3.20

Note: The displayed content to be printed or marked will change its appearance. So e.g. texts could be truncated, barcodes will change its module size or graphics will be stretched!

Size/Height@value

numeric float with Y dimension size in millimeter, eg “12”, “40.123”

Note: Supported by devices since FW 3.20

Note: The displayed content to be printed or marked will change its appearance. So e.g. texts could be truncated, barcodes will change its module size or graphics will be stretched!

Rotation@value

Rotates the label object from base position in degrees.

numeric float with angle in degrees are allowed, eg “0”, “90”, “23”, “113.45”.

Note: Supported by devices since FW 1.84

Note: On HR printing devices also numeric float with angle in degrees are allowed to set, but only 90 degree can be printed, values are rounded to 0, 90, 180, 270 degrees.

Inverted@value

inverts black/white pixels in region of label object

‘true’ inverted

‘false’ not inverted

Note: Supported by HR printing devices since FW 1.84, not supported by Laser marking systems.

HiddenCount@value

instructs how often a label object should not be printed.

- 0: not hidden
- n: hide the label object for the next n prints
- -1: hide the label object infinite times

Note: Supported by devices since FW 1.72

Font/NameEmphasis@value

have to be valid strings for already installed fonts with font name and font emphasis separated by slash, e.g. “`FreeSans/Medium`”, “`FreeSerif/Bold`”

Note: Supported by devices since FW 3.20.

Note: Supported by types of label objects which displays text, as there are Text, Counter, Date or Time and barcodes which support humanreadable text.

Font/File@value

have to be valid strings for already installed fonts with font file an path, e.g. “`/usr/reajet/public/fonts/freesans.ttf`” or “`/usr/reajet/public/fonts/singleline/simhei_singleline.ttf`”

Note: Supported by devices since FW 3.20

Note: Supported by types of label objects which displays text, as there are Text, Counter, Date or Time and barcodes which support humanreadable text.

The current REA-PI Version supports following label object content properties:

Type@value

String with literal name of the label object content analog to the names used in the label, eg “Variable”, “Static”, “Counter”, “Time”, etc. This property is read only!

Note: Supported by devices since FW 3.40

Content@value

sets the text value of a content, possible for static text, variable text, variable key values of choice and static barcode identifiers.
For variable text and key values of choice this function works identical to SetLabelContent(). Static texts and barcode identifiers are still not variable, but setting is available with this function!

Note: Supported by devices since FW 3.20

Note: For variable text and key values of choice this function works identical to SetLabelContent(). Static texts and barcode identifiers are still not variable, but setting is available with this function!

Counter@value

numeric integer, sets the actual counter value, e.g “101”

Note: Supported by devices since FW 3.20 with label object of type Counter.

Counter@reset

no argument, will be ignored, resets the counter to start value in label

Note: Supported by devices since FW 3.20 with label object of type Counter.

Counter@restart

no argument, will be ignored, restarts the counter from restart value in label

Note: Supported by devices since FW 3.20 with label object of type Counter.

Counter/StartValue@value

numeric integer, sets a new start value, e.g “0”

Note: Supported by devices since FW 3.20 with label object of type Counter.

Counter/StartValue@reset

numeric integer, sets a new start value and resets the counter, e.g “0”

Note: Supported by devices since FW 3.20 with label object of type Counter.

Counter/EndValue@value

numeric integer, sets a new end value, e.g “1000”

Note: Supported by devices since FW 3.20 with label object of type Counter.

Counter/RestartValue@value

numeric integer, sets a new restart value, e.g “-15”

Note: Supported by devices since FW 3.20 with label object of type Counter.

Counter/RestartValue@restart

numeric integer, sets the actual restart value and restart the counter, e.g “-15”

Note: Supported by devices since FW 3.20 with label object of type Counter.

BufferSize@value

reserved for future use.

Note: Supported by devices since FW 3.50 with label object of type ContentBuffer.

`BufferPrinted@value`

numeric integer, gets a snapshot of number of printed entries from content buffer since last time when a job was set. This property is read only!

Note: Supported by devices since FW 3.50 with label object of type ContentBuffer.

`BufferExpired@value`

numeric integer, gets a snapshot of number of expired entries from content buffer since last time when a job was set. This property is read only!

Note: Supported by devices since FW 3.50 with label object of type ContentBuffer.

7.2 Structure for transport contents and properties

The commands for get or set label contents, label properties and label content properties are using a defined structure <Entry> for transmit data. The <Data> node of an REA-PI command can contain one or structures <Entry>.

All label properties consist of a reference to that property and a value. All reference will have a reference to a job and a group in that job which represents the label.

So <Reference> defines which property or content should be updated.

<Job> indicates the job identifier of the affected job, 0 for first job. *Note: All versions of device firmware up to current version support only one job.*

<Group> indicates which group and implicitly which label has to be updated. The name of the group is defined by the installation setting, usually “Front”, “Top” or “Back” and “1”, “2”, “3” or “4” respectively. Please refer to the operating manual of your device for further information on how to create and modify installation settings.

<Object> and <Content> reference the label content entity as used in command to set label contents:

```
<Entry>
   <Reference>
      <Job>0</Job>
      <Group>Front</Group>
      <Object>article</Object>
      <Content>text 1</Content>
   </Reference>
   <Value>803.123.0035</Value>
</Entry>
```

<Property> is not needed and must not be declared for label contents!

<Object> and <Property> reference the label object property:

```
<Entry>
   <Reference>
      <Job>0</Job>
      <Group>Front</Group>
      <Object>article</Object>
      <Property>Position/X@value</Property>
   </Reference>
   <Value>1.00</Value>
</Entry>
```

<Content> is not needed and must not be declared for label object properties!

<Object>, <Content> and <Property> reference the label object content property:

```
<Entry>
   <Reference>
      <Job>0</Job>
      <Group>Front</Group>
      <Object>serial</Object>
      <Content>counter 1</Content>
      <Property>Counter/StartValue@value</Property>
   </Reference>
   <Value>100</Value>
</Entry>
```

In all cases <Value> contains the value for the label content or object properties to be set. For label contents the type is always interpreted as a string. For object properties the type, value region and perhaps unit of the value you can see in the previous listing.

7.3 Dynamically change label contents of a loaded job

7.3.1 Command GETLABELCONTENT

This command retrieves a current snapshot of variable contents for types <Variable> and <Choice>. Since version 3.50 of the firmware variable contents of type <ContentBuffer> are supported, too.

It is possible to restrict the query to certain objects / contents by specifying job, group, object and contents respectively. Additionally, regular expressions may be given to these search parameters.

This command is available with a Firmware-Version ≥ 3.20.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETLABELCONTENT" id="318">
         <Data>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>.*</Group>
                  <Object>.*</Object>
                  <Content>.*</Content>
               </Reference>
               <Value></Value>
            </Entry>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

Entry — The defined structure <Entry> for transmitting data. For this command the structure contains parameters to define a search structure. The parameter values contain either an explicit value to be searched or a regular expression.
Use regular expression '.*' as wildcard for group, label object and content to get all variable contents at once.
For more information about the structure <Entry> see chapter What are properties and variable contents, page 62 ff.

Pay attention: in previous version 3.0 if no reference entry is given, you will get all variable contents at once, the behaviour since version 3.1 is extensive changed: no reference will get no result!

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="GETLABELCONTENT" id="318">
         <Data>
            <Entry>
               <Reference>
                  <Job>0</Job>
```

```
                  <Group>1</Group>
                  <Object>article</Object>
                  <Content>text_1</Content>
               </Reference>
               <Value>803.123.0035</Value>
            </Entry>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>qr</Object>
                  <ObjectType>Text</ObjectType>
                  <Content>Content1</Content>
                  <ContentType>Variable</ContentType>
                  <Content>text _2</Content>
               </Reference>
               <Value>http://www.rea-jet.de</Value>
            </Entry>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the returned parameter:

Entry — A list of label contents resulting on the given search. For more information about the structure <Entry> see chapter What are properties and variable contents, page 62 ff. Note that the <Entry> for a content of type <ContentBuffer> may contain several values, i.e., a <Value> list:

```
<Entry>
  <Reference>
    <Job>0</Job>
    <Group>1</Group>
    <Object>serialno</Object>
    <ObjectType>Text</ObjectType>
    <Content>queue</Content>
    <ContentType>ContenBuffer</ContentType>
  </Reference>
  <Value>124-0038</Value>
  <Value>124-0039</Value>
  <Value>124-0040</Value>
  <Value>124-0041</Value>
  <Value>124-0042</Value>
</Entry>
```

In node <Status> an errorcode is specified if the command is not successful.

Referencing undefined jobs or groups/labels., i.e., which are not contained in the current assignment, will make the whole function fail with an error. No result values are returned.

On the other hand, it is allowed to reference an object or content which might not be present. In that case this reference is merely not found and the response simply doesn't contain at least this reference/value.

	Note that the values retrieved are only a snapshot and may become obsolete at the time of receipt of the response of this command!

7.3.2 Command SETLABELCONTENT

This command sets and replaces one or more variable label contents of current loaded label(s) on device.

Apart from a content of the types <Variable> und <Choice> also a content of type <ContentBuffer> may be given, a firmware of version 3.4x and above, supporting the type <ContentBuffer>, supposed.

Fundamental to this command is, that the references provided will be treated atomically altogether, i.e., either all of the values from one SETLABELCONTENT command will be assigned or none at all.

In contrast to this, consecutive SETLABELCONTENT commands will be processed separately. Hence the modified values could be distributed over several prints in that case.

This command is available with a firmware version ≥ 1.80.

	The passed values are only used at runtime of an assigned job in device and are not saved in the label XML file. Therefore they are lost after a new job assignment.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SETLABELCONTENT" id="343">
         <Data>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>article</Object>
                  <Content>text_1</Content>
               </Reference>
               <Value>803.124.0038</Value>
            </Entry>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>qr</Object>
                  <Content>text 2</Content>
               </Reference>
               <Value>http://www.rea-jet.de</Value>
            </Entry>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

Entry	The defined structure <Entry> for transmitting data. For this command the structure contains one or more entries to be set on device. For more information about the structure <Entry> see chapter What are properties and variable contents, page 62 ff.

An <Entry> for a content of type <ContentBuffer> may contain a <Value> list:

```
<Entry>
  <Reference>
    <Job>0</Job>
    <Group>1</Group>
    <Object>serialno</Object>
    <Content>queue</Content>
  </Reference>
  <Value>124-0038</Value>
  <Value>124-0039</Value>
  <Value>124-0040</Value>
  <Value>124-0041</Value>
  <Value>124-0042</Value>
</Entry>
```

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="SETLABELCONTENT" id="343">
         <Data/>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter:

In node <Status> an errorcode is specified if the command is not successful.

Note that a content of type <ContentBuffer> is to be cleared when the corresponding value list is empty.

7.3.3 Command ADDLABELCONTENT

This command adds one or more variable label contents to a content of type <ContentBuffer>. Like the command SETLABELCONTENT, also this command treats the given references atomically, i.e., either adds all of them to the specified content buffers or none.

In contrast to this, consecutive calls of ADDLABELCONTENT result in separate assignments and thus may be spread over several prints.

This command first appeared in firmware 3.38 but is generally available with a firmware version ≥ 3.50.

The passed values are only used at runtime of an assigned job in device and are not saved in the label XML file. Therefore they are lost after a new job assignment.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="ADDLABELCONTENT" id="343">
         <Data>
            <Entry>
              <Reference>
                <Job>0</Job>
                <Group>1</Group>
                <Object>serialno</Object>
                <Content>queue</Content>
              </Reference>
              <Value>124-0038</Value>
              <Value>124-0039</Value>
              <Value>124-0040</Value>
              <Value>124-0041</Value>
              <Value>124-0042</Value>
            </Entry>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Entry` The defined structure <Entry> for transmitting data. For this command the structure contains one or more entries to be set on device.
For more information about the structure <Entry> see chapter What are properties and variable contents, page 62 ff.

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="ADDLABELCONTENT" id="343">
         <Data/>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter:

In node <Status> an errorcode is specified if the command is not successful.

7.4 Manipulate label object properties of a loaded job

7.4.1 Command GETLABELOBJECT

This command retrieves a snapshot of a specified query of variable contents of types ‘Variable’ and ‘Choice’ and the specified label object properties of all used labels in an assigned job.

This command is available with a Firmware-Version ≥ 3.20.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETLABELOBJECT" id="364">
         <Data>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>.*</Group>
                  <Object>.*</Object>
                  <Content>.*</Content>
               </Reference>
               <Value></Value>
            </Entry>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>Front</Group>
                  <Object>.*</Object>
                  <Content>.*</Content>
                  <Property>Counter@value</Property>
               </Reference>
               <Value></Value>
            </Entry>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Entry` The defined structure <Entry> for transmitting data. For this command the structure contains parameters to define a search structure. The parameter values contain either an explicit value to be searched or a regular expression.
Use regular expression ‘.*’ as wildcard for group, label object and content to get all variable contents at once.
For more information about the structure <Entry> see chapter What are properties and variable contents, page 62 ff.

Pay attention: in previous version 3.0 if no reference entry is given, you will get all variable contents at once, the behaviour since version 3.1 is extensive changed: no reference will get no result!

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="GETLABELOBJECT" id="325">
         <Data>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>article</Object>
                  <Property>Position/X@value</Property>
               </Reference>
               <Value>1.00</Value>
            </Entry>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>article</Object>
                  <Property>Position/Y@value</Property>
               </Reference>
               <Value>1.00</Value>
            </Entry>
            <Entry>
               ...
            </Entry>
            ...
          </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the returned parameter:

`Entry` A list of found label contents resulting on the previous search.
For more information about the structure <Entry> see chapter What are properties and variable contents, page 62 ff.

In node <Status> an errorcode is specified if the command is not successful.

	Referencing undefined jobs or groups/labels., i.e., which are not contained in the current assignment, will make the whole function fail with an error. No result values are returned. On the other hand, it is allowed to reference an object or content which might not be present. In that case this reference is merely not found and the response simply doesn't contain at least this reference/value.
	Remember, retrieved values are only a snapshot and can possibly be obsolete at time of receipt of the response of this command!

Additional information to label width and height.

TITAN: firmware version = / > 3.96~40587

TITAN 2.0: not yet implemented

Command:

```
<REA-JET>
    <REA-PI version="3.9">
        <Command name="GETLABELOBJECT" id="364">
            <Data>
                <Entry>
                    <Reference>
                        <Job>0</Job>
                        <Group>.*</Group>
                        <Object>none</Object>
                        <Content>none</Content>
                    </Reference>
                    <Value/>
                </Entry>
                <Entry>
                    <Reference>
                        <Job>0</Job>
                        <Group>.*</Group>
                        <Property>Size/Width@value</Property>
                    </Reference>
                    <Value/>
                </Entry>
            </Data>
        </Command>
    </REA-PI>
</REA-JET>
```

Response:

```
<REA-JET>
    <REA-PI>
        <Command name="GETLABELOBJECT" id="364">
            <Data>
                <Entry>
                    <Reference>
                        <Job>0</Job>
                        <Group>Front</Group>
                        <Property>Size/Width@value</Property>
                    </Reference>
                    <Value>75.00</Value>
                </Entry>
            </Data>
        </Command>
        <Status>
            <Domain>102</Domain>
            <Code>0</Code>
            <Message>successful executed</Message>
        </Status>
    </REA-PI>
</REA-JET>
```

7.4.2 Command SETLABELOBJECT

This command changes each one or more label object properties, content properties and variable label contents of current loaded label(s) on device.

The references of one command are treated atomically, i.e., all object properties and label contents are changed together. This means for one specific printing operation all modifications in the label are done simultaneously or no modification will be done at all.

In contrast to this consecutive SETLABELOBJECT commands are not atomic; modifications will be done successively and thus may be spread over several prints.

The passed values are only used at runtime of an assigned job in device and are not saved in the label XML file. Therefore they are lost after a new job assignment.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SETLABELOBJECT" id="369">
         <Data>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>article</Object>
                  <Content>text 1</Content>
                  <Property>Content@value</Property>
               </Reference>
               <Value>803.125.0037</Value>
            </Entry>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>article</Object>
                  <Property>Position/X@value</Property>
               </Reference>
               <Value>2</Value>
            </Entry>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Entry` The defined structure <Entry> for transmitting data. For this command the structure contains one or more entries to be set on device.
For more information about the structure <Entry> see chapter What are properties and variable contents, page 62 ff.

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="SETLABELOBJECT" id="369">
```

```
            <Data/>
         </Command>
         <Status>
            <Domain>102</Domain>
            <Code>0</Code>
            <Message>successful executed</Message>
         </Status>
      </REA-PI>
   </REA-JET>
```

The node <Data> specifies no parameter:

In node <Status> an errorcode is specified if the command is not successful.

The passed values are only used at runtime of an assigned job in device and are not saved in the label XML file. Therefore they are lost after a new job assignment.

7.5 Label manipulation control events

7.5.1 Event READYFORNEXTCONTENT

The device sends this event when it is safe to send new data for a variable, i.e., the print buffer for the current print has been locked and thus sending new data may only change what is printed subsequently. Furthermore, note that, in single print mode, this event also indicates that new data has to be sent, as otherwise MISSINGCONTENT and / or PRINTREJECT could occur upon the next trigger.

In contrast to the event READYFORNEXTCONTENT, which applies to the content types <Variable> and <Choice>, the event BUFFERUNDERRUN only occurs for contents of type <ContentBuffer> and therefore behaves differently.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 3.40.

7.5.1.1 Event subscription

The subscription is only effective while a job and a label has been loaded. It is recommended to subscribe to this event when the event for a new job was set has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="19">
         <Data>
            <EVENT>READYFORNEXTCONTENT</EVENT>
            <EVENT ARGS name="JobID">0</EVENT ARGS>
            <EVENT_ARGS name="GroupName">1</EVENT_ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Group name that will be observed

```
<REA-JET>
   <REA-PI>
      <Command name="SUBSCRIBE" id="19">
         <Data>
            <EVENT>READYFORNEXTCONTENT</EVENT>
            <EVENT ARGS name="JobID">0</EVENT ARGS>
         </Data>
      </Command>
      <Status>
         <Domain>400</Domain>
         <Code>0</Code>
         <Message>subscribtion successful done</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Repetition of the identifier of the affected job, 0 for first job. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Repetition of the group that will be observed

7.5.1.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="READYFORNEXTCONTENT" id="">
         <Data>
            <Job>0</Job>
            <Group>1</Group>
            <PrintTrigger>0</PrintTrigger>
            <IsTriggerGroup>true</IsTriggerGroup>
            <TS>14690894176644324458</TS>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the event parameter:

`Job`	Identifier of the job, where it is safe and / or necessary to send new data to. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Name of the print head group, where it is safe and / or necessary to send new data to.

7.5.2 Event INVALIDCONTENT

The device sends this event if a label contents or label properties could not be rendered by the device. In this case no print image (buffer) will be generated and subsequently printing is not possible.

This event may indicate that a content which has been set by the command SETLABELCONTENT or SETLABELOBJECT is invalid.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 1.80.

7.5.2.1 Event subscription

The subscription is only effective while a job and a label has been loaded. It is recommended to subscribe to this event when the event for a new job was set has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="443">
         <Data>
            <EVENT>INVALIDCONTENT</EVENT>
            <EVENT ARGS name="JobID">0</EVENT ARGS>
            <EVENT_ARGS name="GroupName">1</EVENT_ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Group name of the print head group that will be observed

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="443">
         <Data>
            <EVENT>INVALIDCONTENT</EVENT>
            <EVENT ARGS name="JobID">0</EVENT ARGS>
            <EVENT_ARGS name="GroupName">1</EVENT_ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Identifier of the job which will no longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Group name of the print head group that will be observed

7.5.2.2 Event parameters

```
<REA-JET>
  <REA-PI>
    <Command name="INVALIDCONTENT" id="">
      <Data>
        <Job>0</Job>
        <Group>1</Group>
        <Status>
          <Domain>domain</Domain>
          <Code>code</Code>
          <Message>message</Message>
        </Status>

      </Data>
    </Command>
    <Status>
      <Domain>domain</Domain>
      <Code>code</Code>
      <Message>message</Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the event parameter:

`Job`	Get the identifier of the job for which the missingcontent event was occurred, 0 for the first job. *Note: All versions of device firmware up to current version support only one job*.
`Group`	Get the name of the group in which the invalidcontent event was occurred.
`Status`	Get the error information why this invalidcontent error occurs. For more information about errorcodes see chapter 18 Appendix A1: Error Codes, page 128.

7.5.3 Event LABELPROPERTYCHANGED

The device sends this event if one or more label contents or label properties has been changed.

These changes are caused not only by commands to set label contents or label properties, instead they may also be caused by dynamic label objects like date/time <Time>, counter <Counter>, etc.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 2.00.

7.5.3.1 Event subscription

The subscription is only effective while a job and a label has been loaded. It is recommended to subscribe to this event when the event for a new job was set has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>LABELPROPERTYCHANGED</EVENT>
            <EVENT_ARGS name="JobID" value="0" />
            <EVENT_ARGS name="GroupName" value="Front" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Group name of the print head group that will be observed

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>LABELPROPERTYCHANGED</EVENT>
            <EVENT_ARGS name="JobID" value="0" />
            <EVENT_ARGS name="GroupName" value="Front" />
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Identifier of the job which will no longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Group name of the print head group that will be observed

7.5.3.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="LABELPROPERTYCHANGED" id="">
         <Data>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>article</Object>
                  <Content>text_1</Content>
               </Reference>
               <Value>803.124.0038</Value>
            </Entry>
            <Entry>
               <Reference>
                  <Job>0</Job>
                  <Group>1</Group>
                  <Object>article</Object>
                  <Content>text 1</Content>
                  <Property>Content@value</Property>
               </Reference>
               <Value>803.124.0038</Value>
            </Entry>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the event parameter:

`Entry` A list of changed label contents and label object properties, which are changed.
For more information about the structure <Entry> see chapter What are properties and variable contents, page 62 ff.

7.5.4 Event LABELEVENT

The device sends this event when a label event has been raised due to a change of a digital input or output and label properties are affected.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 3.20.

7.5.4.1 Event subscription

The subscription is only effective while a job and a label has been loaded. It is recommended to subscribe to this event when the event for a new job was set has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="1765">
         <Data>
            <EVENT>LABELEVENT</EVENT>
            <EVENT_ARGS name="JobID">0</EVENT_ARGS>
            <EVENT_ARGS name="LabelEvents">3500-3627</EVENT_ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

`LabelEvents` The required argument specifies the requested label events by identifier.
The format is one label event identifier or more label event identifiers separated by semicolon or a range of label event identifiers with hyphen.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="1765">
         <Data>
            <EVENT>LABELEVENT</EVENT>
            <EVENT ARGS name="JobID">0</EVENT ARGS>
            <EVENT_ARGS name="LabelEvents">3500-3627</EVENT_ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job` Identifier of the job which will no longer observed. However, the specification of the job id is optional; if not assigned; the first or only job is used.
Note: All versions of device firmware up to current version support only one job

`LabelEvents` The required argument specifies the repetition of the label events by identifier, nevertheless all subscribed label events are unsubscribed.

The format is one label event identifier or more label event identifiers separated by semicolon or a range of label event identifiers with hyphen.

The Parameter labelevents is not required, because simply all labelevents are unsubscribed.

7.5.4.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="LABELEVENT" id="">
         <Data>
            <Job>0</Job>
            <LabelEvent>3522</LabelEvent>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the event parameter:

`Job` Get the identifier of the job for which the label event event was occurred, 0 for the first job.
Note: All versions of device firmware up to current version support only one job.

`LabelEvent` Get a string with a label event identifier when a label event has been raised due to a change of a digital input or output.
Remark that each event sends only one identifier, regardless how many label events are sucscribed.

7.6 Label manipulation content buffer control

This section deals with the management of the “content buffer” functionality, i.e., a FIFO which holds values to be assigned (from print to print) to the corresponding entities in the label. There are commands for the configuration of the FIFO as well as events for the notification about the current buffer status:

- SETCONTENTBUFFERCONTROL: Set parameters for the buffer. These include
  - BufferUnderrun - threshold when event BUFFERUNDERRUN should be issued
  - BufferFull - threshold when event BUFFERFULL should be issued
  - Expiration - whether entries should expire from the buffer after given delay
  - ExpireDelay - delay for new entries to expire
- GETCONTENTBUFFERCONTROL: Get currently configured parameters for buffering
  - MaxBufferSize - currently not implemented (always 0)
  - BufferUnderrun - threshold when event BUFFERUNDERRUN should be issued
  - BufferFull - threshold when event BUFFERFULL should be issued
  - Expiration - whether new entries should expire after given delay
  - ExpireDelay - configured delay for new entries to expire
- BUFFERUNDERRUN: This event indicates that the number of entries left in the buffer is below the configured threshold (parameter “BufferUnderrun” see above).
- BUFFERFULL: This event indicates that the number of entries left in the buffer is above the configured threshold (parameter “BufferFull” see above).

7.6.1 Command GETCONTENTBUFFERCONTROL

This command requests the device to report the current buffer configuration, cf. the parameters above.

This command is available with a Firmware-Version ≥ 3.50.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETCONTENTBUFFERCONTROL" id="85">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

```
<REA-JET>
   <REA-PI>
      <Command name="GETCONTENTBUFFERCONTROL" id="85">
         <Data>
            <MaxBufferSize>0</MaxBufferSize>
            <BufferUnderrun>0</BufferUnderrun>
            <BufferFull>3812</BufferFull>
            <Expiration>false</Expiration>
            <ExpireDelay>60</ExpireDelay>
         </Data>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message>Get parameter and status informationn about content buffer control
successfully retrieved!</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

7.6.2 Command SETCONTENTBUFFERCONTROL

This command sets the buffer configuration, cf. the parameters above.

This command is available with a Firmware-Version ≥ 3.50.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SETCONTENTBUFFERCONTROL" id="86">
         <Data>
            <BufferUnderrun>0</BufferUnderrun>
            <BufferFull>3812</BufferFull>
            <Expiration>true</Expiration>
            <ExpireDelay unit="s">60</ExpireDelay>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

```
<REA-JET>
   <REA-PI>
      <Command name="SETCONTENTBUFFERCONTROL" id="86">
         <Data/>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message>Set parameter of content buffer control successfully retrieved!</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

7.6.3 Event BUFFERUNDERRUN

The device sends this event when the FIFO is going to run out of data, i.e., the number of entries in the buffer is below the threshold, which was configured by the “BufferUnderrun” parameter.

In contrast to the event READYFORNEXTCONTENT, which applies to the content types <Variable> and <Choice>, the event BUFFERUNDERRUN only occurs for contents of type <ContentBuffer> and therefore behaves differently.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 3.50.

7.6.3.1 Event subscription

The subscription is only effective while a job and a label has been loaded. It is recommended to subscribe to this event when the event for a new job was set has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="19">
         <Data>
            <EVENT>BUFFERUNDERRUN</EVENT>
            <EVENT_ARGS name="0">0</EVENT_ARGS>
            <EVENT_ARGS name="1"></EVENT_ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Group name that will be observed

```
<REA-JET>
   <REA-PI>
      <Command name="SUBSCRIBE" id="19">
         <Data>
            <EVENT>BUFFERUNDERRUN</EVENT>
            <EVENT_ARGS name="0">0</EVENT_ARGS>
            <EVENT_ARGS name="1"></EVENT_ARGS>
         </Data>
      </Command>
      <Status>
         <Domain>400</Domain>
         <Code>0</Code>
         <Message>subscribtion successful done</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Repetition of the identifier of the affected job, 0 for first job. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Repetition of the group that will be observed

7.6.3.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="BUFFERUNDERRUN" id="">
         <Data>
            <Job>0</Job>
            <Group>1</Group>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the event parameter:

`Job`	Identifier of the job, where a buffer needs to receive data. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Group name of the print head group, where a buffer needs to receive data.

7.6.4 Event BUFFERFULL

The device sends this event when the number of entries in the buffer is above the threshold, which was configured by the “BufferFull” parameter.

The event only relates to contents of type <ContentBuffer> when the fill level of the corresponding queue allows sending new data.

This event requires a subscription and has event parameters.

This command is available with a Firmware-Version ≥ 3.50.

7.6.4.1 Event subscription

The subscription is only effective while a job and a label has been loaded. It is recommended to subscribe to this event when the event for a new job was set has been received.

Unsubscribe the event if it is no longer needed. The event is automatically unsubscribed, when the job is replaced by another job or is simply unloaded.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="20">
         <Data>
            <EVENT>BUFFERFULL</EVENT>
            <EVENT_ARGS name="0">0</EVENT_ARGS>
            <EVENT_ARGS name="1"></EVENT_ARGS>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	Identifier of the job which will be observed. However, the specification of the job id is optional; if not assigned; the first or only job is used. *Note: All versions of device firmware up to current version support only one job*
`GroupName`	Group name that will be observed

```
<REA-JET>
   <REA-PI>
      <Command name="SUBSCRIBE" id="20">
         <Data>
            <EVENT>BUFFERFULL</EVENT>
            <EVENT_ARGS name="0">0</EVENT_ARGS>
            <EVENT_ARGS name="1"></EVENT_ARGS>
         </Data>
      </Command>
      <Status>
         <Domain>400</Domain>
         <Code>0</Code>
         <Message>subscribtion successful done</Message>
      </Status>
```

```
      </REA-PI>
   </REA-JET>
```

The node <Data> specifies the parameter:

Job	Repetition of the identifier of the affected job, 0 for first job. *Note: All versions of device firmware up to current version support only one job*
GroupName	Repetition of the group that will be observed

7.6.4.2 Event parameters

```
<REA-JET>
   <REA-PI>
      <Command name="BUFFERFULL" id="">
         <Data>
            <Job>0</Job>
            <Group>1</Group>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the event parameter:

Job	Identifier of the job, where a buffer allows to receive data. *Note: All versions of device firmware up to current version support only one job*
GroupName	Group name of the print head group, where a buffer allows to receive data

8 Device settings HR Family (HR + HR 2.0)

8.1 What happened with print heads and cartridges

8.1.1 Command / Event GETCARTRIDGES

This command retrieves a snapshot containing information and states of all cartridges. Some information is available only if the cartridge is involved by the loaded job.

For historical reasons, the response will also be sent automatically - as an event – by the device, whenever the state of a cartridge changes. For example this could be due to the insertion or removal of a cartridge or when the ink content falls below the error or warning level.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETCARTRIDGES" id="17521">
         <Data></Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter:

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="GETCARTRIDGES" id="17521">
         <Data>
            <CARTRIDGE>
               <PRINTHEAD ID>0</PRINTHEAD ID>
               <NAME>REA Dye black</NAME>
               <InkID>054950100</InkID>
               <MaxLevel unit="ml">42.000</MaxLevel>
               <GAUGE ML unit="ml">5.300</GAUGE ML>
               <GAUGE PERCENT>12.619</GAUGE PERCENT>
               <LABELSPRINTABLE>9093</LABELSPRINTABLE>
               <InkUsagePerPrint unit="nl">582.840</InkUsagePerPrint>
               <BULK>False</BULK>
               <TEMP>-1</TEMP>
               <DEFECTNOZZELS>-1</DEFECTNOZZELS>
               <State>
                  <Spitting/>
               </State>
               <Warning>
                  <Ink/>
               </Warning>
               <Error/>
            </CARTRIDGE>
            <CARTRIDGE>
               <PRINTHEAD ID>1</PRINTHEAD ID>
               <NAME>REA JET HR Black Eagle, Chip</NAME>
               <InkID>054950101</InkID>
               <MaxLevel unit="ml">42.000</MaxLevel>
               <GAUGE_ML unit="ml">4.400</GAUGE_ML>
               <GAUGE_PERCENT>10.476</GAUGE_PERCENT>
               <LABELSPRINTABLE>-1</LABELSPRINTABLE>
               <InkUsagePerPrint>-1</InkUsagePerPrint>
               <BULK>False</BULK>
               <TEMP>-1</TEMP>
               <DEFECTNOZZELS>-1</DEFECTNOZZELS>
```

```
                <State/>
                <Warning>
                   <Ink/>
                </Warning>
                <Error/>
             </CARTRIDGE>
          </Data>
       </Command>
       <Status>
          <Domain>102</Domain>
          <Code>0</Code>
          <Message>successful executed</Message>
       </Status>
    </REA-PI>
 </REA-JET>
```

The node <Data> specifies the returned parameter:

`Cartridge` Structure containing cartridge information and states

`HeadID` Identifier of print head which charged the cartridge: This parameter could also be called cartridge identifier as these identifiers are identical.

The first print head /cartridge is 0, second is 1 and so on.

`Name` Name of the cartridge, more precisely the name of the ink in this cartridge

`InkID` Identifier of the ink: This is derived from article number and can be interpreted as unique type identifier.

`MaxLevel` Maximum fill level / size of the cartridge

`Gauge_ML` Current fill level of selected cartridge in milliliter

`Gauge_Percent`

Current fill level / gauge of selected cartridge in percent

`LabelsPrintable`

Number of labels which can be printed with current label configuration and ink level of cartridge

-1 if the selected cartridge is not involved in any job.

`InkUsagePerPrint`

Returns ink usage per print for specified cartridge in pico liter as floating point number

-1 if the selected cartridge is not involved in any job.

Note: For compatibility reasons this node is duplicated in uppercase returning an integer.

This old uppercase variant is deprecated! Since firmware 3.20, the more precise form should be used.

`Bulk` Indicates whether a bulk cartridge is used. Returns true for Bulk or false otherwise.

`Temperature`

Returns current temperature of specified cartridge in degree centigrade or -1, if the temperature is not available.

`DefectNozzles`

Returns the number of defect nozzles or -1, if no measurement is available.

`State` Returns current states in subnodes, e.g., ‘Spitting’ or ‘PulseWarming’.

`Warning` Returns the current warning state in subnodes.

`Error` Returns the current error state in subnodes.

In node <Status> an error code will be given, if the command is not successful.

8.1.2 Command GETCARTRIDGEPARAMETER

This command is used to retrieve the cartridge parameters for all heads, which can have cartridges.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETCARTRIDGEPARAMETER" id="145">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="GETCARTRIDGEPARAMETER" id="145">
         <Data>
            <Cartridge id="0">
               <InkWarningLevel unit="ml">10</InkWarningLevel>
               <InkErrorLevel unit="ml">1</InkErrorLevel>
               <InkErrorEvent>StopJob</InkErrorEvent >
               <VoltageMode>ChipVoltage</VoltageMode>
            </Cartridge>
            <Cartridge id="1">
               <InkWarningLevel unit="ml">10</InkWarningLevel>
               <InkErrorLevel unit="ml">1</InkErrorLevel>
               <InkErrorEvent>StopJob</InkErrorEvent >
               <VoltageMode>ChipVoltage</VoltageMode>
            </Cartridge>
         </Data>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Cartridge` Structure containing cartridge parameters

`InkWarningLevel` Returns ink warning level in milliliter. The device will report an ink warning if the fill level of a cartridge falls below this level. The warning (event) itself is independent of the ink warning *behavior* described below.

`InkErrorLevel` Returns ink error level in milliliter. The device will report an ink error if the fill level of a cartridge falls below this level. This error (event) is independent of the ink error *behavior* described below.

`InkErrorEvent` Returns ink error behavior. If empty, no special behavior has been defined (apart from sending an event) for the cartridge fill level falling below the ink error level. In case of ‘StopJob’ the current job (if any) will be stopped.

`VoltageMode` DEPRECATED: Returns true if fixed voltage is used.

In node <Status> an error code will be given, if the command is not successful.

8.1.3 Command SETCARTRIDGEPARAMETER

This command is used to change one or more specified cartridge parameters. The function will apply to all cartridges the same, i.e., it is not possible to set different parameters for several cartridges.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SETCARTRIDGEPARAMETER" id="330">
         <Data>
            <VoltageMode>ChipVoltage</VoltageMode>
            <InkWarningLevel unit="ml">5</InkWarningLevel>
            <InkWarningEvent>None</InkWarningEvent>
            <InkErrorLevel unit="ml">2</InkErrorLevel>
            <InkErrorEvent>StopJob</InkErrorEvent>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`InkWarningLevel` Sets ink warning level in in full milliliter as integer. The device will report an ink warning if the fill level of a cartridge falls below this level. The value must be larger than the ink error level (see below)!

`InkErrorLevel` Sets ink error level in in full milliliter as integer. The device will report an ink error if the fill level of a cartridge falls below this level. The value must be smaller than the ink warning level (see above)!

`InkErrorEvent` Sets ink error behavior. If ‘None’ or empty, no special behavior will be defined (apart from sending an event).
If set to ‘StopJob’, the current job (if any) will be stopped immediately as soon as the fill level of any cartridge falls below the ink error level.

`VoltageMode` DEPRECATED: For the parameter VoltageMode two parameters are possible: Use ‘ChipVoltage’ to load the optimal parameters from the cartridge’s chip.
When set to ‘FixedVoltage’ a print voltage of 10.6 V will be used (DEPRECATED).

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="SETCARTRIDGEPARAMETER" id="330">
         <Data/>
      </Command>
      <Status>
         <Domain>102</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter.

In node <Status> an errorcode is specified if the command is not successful.

9 Device settings Laser Family

9.1.1 Event LASERUNITINFO

The device sends this event when the operation state of a laser unit has changed.

```
<REA-JET>
  <REA-PI>
    <Command name="LASERUNITINFO" id="">
      <Data>
        <Laserunit id="0">
          <Type>YLP-V2-1-100-50-50</Type>
          <Firmware>31.1.75.1.1;1.45;2.28;</Firmware>
          <SerialNumber>11040530</SerialNumber>
          <Power>30</Power>
          <ExtInterlock status="closed"/>
          <Shutter status="open"/>
          <PilotLaser status="inactive" io="False"/>
          <Warning/>
          <Error/>
  <Error>
    <Overtemp/>
    <ShutterSwitches/>
    <SafetyRelay/>
  </Error>
        </Laserunit>
      </Data>
    </Command>
    <Status>
      <Domain>domain</Domain>
      <Code>code</Code>
      <Message>message</Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The node <Error>can contain one or more of the following:

```
  <Error>
    <Overtemp/>
    <ShutterSwitches/>
    <SafetyRelay/>
  </Error>
```

9.1.1.1 Event subscription

This event can be subscribed after a connection to the device has been established and will occur immediately after subscription to inform about actual laser unit info state.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SUBSCRIBE" id="0">
         <Data>
            <EVENT>LASERUNITINFO</EVENT>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The command above subscribes an event LASERUNITINFO event.
No arguments are needed.

The subscription of event LASERUNITINFO will not expired or unsubscribed automatically, except if the connection to the device will be closed.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="UNSUBSCRIBE" id="0">
         <Data>
            <EVENT>LASERUNITINFO</EVENT>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The command above unsubscribes an event LASERUNITINFO. No arguments are needed.

10 Device settings DOD 2.0 / ST 2.0 Family

At time of writing, there are no REA-PI functions specific to the DOD 2.0 / ST product family.

11 Device settings SC 2.0

Note: A delay greater than 2 seconds is required between the stop job and the start job

Jet Start

```
<REA-JET>
 <REA-PI version="3.9">
 <Command name="STARTJET" id="4103">
 <Data>
 <Job id="0"/>
 <ReloadJob>False</ReloadJob>
 </Data>
 </Command>
 </REA-PI>
</REA-JET>
```

Response

```
<REA-JET>
 <REA-PI>
 <Command name="STARTJET" id="4103">
 <Data/>
 </Command>
 <Status>
 <Domain>102</Domain>
 <Code>0</Code>
 <Message>jet sucessfully started</Message>
 </Status>
 </REA-PI>
</REA-JET>
```

Jet Stop

```
<REA-JET>
 <REA-PI version="3.9">
 <Command name="STOPJET" id="4103">
 <Data>
 <Job id="0"/>
 <ReloadJob>False</ReloadJob>
 </Data>
 </Command>
 </REA-PI>
</REA-JET>
```

Response

```
<REA-JET>
 <REA-PI>
 <Command name="STOPJET" id="4103">
 <Data/>
 </Command>
 <Status>
 <Domain>102</Domain>
 <Code>0</Code>
 <Message>jet sucessfully stopped</Message>
 </Status>
 </REA-PI>
</REA-JET>
```

See also chapter 5 in the manual "REA JET SC 2.0 Manual for daily maintenance".

12 Device settings GK 2.0 Family

At time of writing, there are no REA-PI functions specific to the GK 2.0 product family.

13 Device settings UP

At time of writing, there are no REA-PI functions specific to the UP product family.

14 Device settings IO configuration

14.1.1 Command GETIOCONFIGURATION

This command is used to retrieve the name of the I/O configuration which is currently set.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="GETIOCONFIGURATION" id="37">
      <Data/>
    </Command>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter:

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
  <REA-PI>
    <Command name="GETIOCONFIGURATION" id="37">
      <Data>
        <Filename>default.dio</Filename>
      </Data>
    </Command>
    <Status>
      <Domain>500</Domain>
      <Code>0</Code>
      <Message>IO-Configuration successfully retrieved!</Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Filename` The name of the currently set I/O configuration.

In node <Status> an errorcode is specified if the command is not successful.

14.1.2 Command SETIOCONFIGURATION

Request the device to load the specified I/O configuration on device.

This command is available with a Firmware-Version ≥ 1.80.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="SETIOCONFIGURATION" id="38">
      <Data>
        <Filename>default.dio</Filename>
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Filename` Filename of I/O configuration definition file.

The io configuration file has to exist in the “rea-jet/device” folder on the device share. The filename must only consist of the underscore “_”, numeric characters and lower case letters of the “Basic Latin” Unicode block.

The response from the device indicates whether the operation was accepted, verified and started successfully:

```
<REA-JET>
  <REA-PI>
    <Command name="SETIOCONFIGURATION" id="38">
      <Data/>
    </Command>
    <Status>
      <Domain>500</Domain>
      <Code>0</Code>
      <Message>IO-Configuration successfully set!</Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter.

In the node <Status> an error code is specified whether the command is successful or not. Typical errors include a missing I/O configuration or a mismatch of the device type the configuration was created for. Furthermore, an error will be raised, when a I/O configuration is tried to be assigned while the print system is started.

14.1.3 Event IOCONFIGURATIONSET

The device raises this event when a new I/O configuration has been assigned. The event will also be raised when the I/O configuration is reloaded because a job has be (re-) assigned.

This event requires a subscription and has event parameters. After subscription the event will occur once to inform which I/O configuration is assigned.

This event is available since firmware version ≥ 1.80.

14.1.3.1 Subscription

The subscription is effective during a connection. It is recommended to subscribe this event immediately after a connection to a device is established.

The subscription of the event can either be terminated by calling the unsubscribe function or will be terminated automatically when the network connection is closed.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="SUBSCRIBE" id="3244">
      <Data>
        <EVENT>IOCONFIGURATIONSET</EVENT>
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the subscription for

The response from the device indicates whether the subscription was successfully.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="UNSUBSCRIBE" id="3244">
      <Data>
        <EVENT>IOCONFIGURATIONSET</EVENT>
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the unsubscription for

The response from the device indicates whether the subscription was successfully.

14.1.3.2 Event parameters

```
<REA-JET>
  <REA-PI>
    <Command name="IOCONFIGURATIONSET" id="">
      <Data>
         <Job>0</Job>
        <Filename>finest 2.dio</Filename>
      </Data>
    </Command>
    <Status>
      <Domain>102</Domain>
      <Code>0</Code>
      <Message>successful executed</Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Job`	This Parameter deprecated. Do not use for any reason.
`Filename`	Get the filename of the I/O configuration set.

14.1.4 Command GETPRODUCTSENSORLEVEL

This command is used to retrieve the levels for all available hardware product sensors.

It is not possible to change the state of hardware product sensors via network!

This command is available with a Firmware-Version ≥ 3.30

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETPRODUCTSENSORLEVEL" id="61">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="GETPRODUCTSENSORLEVEL" id="61">
         <Data>
            <ProductSensor index="0">
               <Level>Low</Level>
            </ProductSensor>
            <ProductSensor index="1">
               <Level>Low</Level>
            </ProductSensor>
            <ProductSensor index="2">
               <Level>Low</Level>
            </ProductSensor>
            <ProductSensor index="3">
               <Level>Low</Level>
            </ProductSensor>
         </Data>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message>PS level info successfully retrieved!</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

On success, the node <Data> contains the following information:

`ProductSensor`	Structure of parameters for product sensor. Depending of the type of device, the number of product sensors can vary.
`@index`	Identifier of the product sensor.
`Level`	Current state of hardware product sensor. If the sensor is triggered, the state is “High”, else “Low”

In node <Status> an error code is specified if the command is not successful.

14.1.5 Command GETIOINPUTLEVEL

This command is used to retrieve the levels for all available digital inputs.

It is not possible to change the state of digital inputs via network!

This command is available with a Firmware-Version ≥ 3.30

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETIOINPUTLEVEL" id="62">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="GETIOINPUTLEVEL" id="62">
         <Data>
            <DigitalInput index="0">
               <Level>Low</Level>
            </DigitalInput>
            <DigitalInput index="1">
               <Level>High</Level>
            </DigitalInput>
            <DigitalInput index="2">
               <Level>Low</Level>
            </DigitalInput>
            <DigitalInput index="3">
               <Level>Low</Level>
            </DigitalInput>
            <DigitalInput index="4">
               <Level>Low</Level>
            </DigitalInput>
            <DigitalInput index="5">
               <Level>Low</Level>
            </DigitalInput>
         </Data>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message>IO-Input level info successfully retrieved!</Message>
      </Status>
   </REA-PI>
</REA-JET>
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETIOINPUTLEVEL" id="145">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

On success, the node <Data> contains the following information:

`DigitalInput`	Structure of parameters for a digital input. Depending of the type of device, the number of digital inputs can vary.
`@index`	Identifier of the digital input.

`Level`	Current state of digital input. If the digital input is set, the state is “High”, else “Low”.

In node <Status> an errorcode is specified if the command is not successful.

14.1.6 Command GETIOOUTPUTLEVEL

This command is used to retrieve the levels for all available digital outputs.

For changing the state of digital outputs via network see SETIOOUTPUTLEVEL.

This command is available with a Firmware-Version ≥ 3.30

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETIOOUTPUTLEVEL" id="63">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="GETIOOUTPUTLEVEL" id="63">
         <Data>
            <DigitalOutput index="0">
               <Level>Low</Level>
            </DigitalOutput>
            <DigitalOutput index="1">
               <Level>Low</Level>
            </DigitalOutput>
            <DigitalOutput index="2">
               <Level>Low</Level>
            </DigitalOutput>
            <DigitalOutput index="3">
               <Level>High</Level>
            </DigitalOutput>
         </Data>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message>IO-Output level info successfully retrieved!</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

On success, the node <Data> contains the following information:

`DigitalOutput`	Structure of parameters for a digital output. Depending of the type of device , the number of digital outputs can vary.
`@index`	Identifier of the digital output.
`Level`	Current state of digital output. If the digital output is set, the state is “High”, else “Low”.

In node <Status> an errorcode is specified if the command is not successful.

14.1.7 Command SETIOOUTPUTLEVEL

This command is used to change one or more states for specified digital outputs.

It is strongly discouraged to use SETIOOUTPUTLEVEL for an output wich is controlled by the device.

	**It is strongly discouraged to use SETIOOUTPUTLEVEL for an output which is controlled by the device!**

For retrieving the states of digital outputs via network see GETIOOUTPUTLEVEL.

This command is available with a Firmware-Version Version ≥ 3.30

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SETIOOUTPUTLEVEL" id="55">
         <Data>
            <DigitalOutput index="1">
               <Level>High</Level>
            </DigitalOutput>
            <DigitalOutput index="3">
               <Level>Low</Level>
            </DigitalOutput>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`DigitalOutput`	Structure of parameters for a digital output. . Depending of the type of device , the number of digital outputs can vary.
`@index`	Identifier of the digital output. This Index is “0” for the first digital output!
`Level`	State of digital output to be set. If the digital output is to be set, the state has to be “High”, otherwise “Low”.

The response from the device indicates whether the operation was executed successfully:

```
<REA-JET>
   <REA-PI>
      <Command name="SETIOOUTPUTLEVEL" id="55">
         <Data/>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message>IO-Output levels successfully set!</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter.

In node <Status> an errorcode is specified if the command is not successful.

15 Common device settings

15.1.1 Command GETDEVICEINFO

The following command can be used to query the device information.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETDEVICEINFO" id="myid">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

<Data> is empty in the request, note the use of the shorthand <Data/> for an empty XML element.

Example for a command response:

```
<REA-JET>
  <REA-PI>
    <Command name="GETDEVICEINFO" id="1">
      <Data>
        <Firmware>3.98~38751</Firmware>
        <SerialNumber>20340000035-6</SerialNumber>
        <ArticleNumber>900.544.30</ArticleNumber>
        <FPGAVersion>FPGA V3.61, MemLay V0.1 (2023-04-06 13:03:58)</FPGAVersion>
        <DeviceType>20070004</DeviceType>
        <Capability>
          ...
        </Capability>
        <AddOn>
          ...
        </AddOn>
      </Data>
    </Command>
    <Status>
      <Domain>99</Domain>
      <Code>0</Code>
      <Message>Query device info successfull</Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The response carries version of firmware, serial number and type of device.

The information about article number is available since version 3.4.

The information about FPGA Version is available since version 3.5.

The device capabilities are available since version 3.4 for device type HR and Laser, since version 3.8 for device types DOD and CIJ and since version 3.9 for device types GK and UP.

Since version 3.4 additional Add-On Modules can be sending with device info. These items are for optional information purposes only.

15.1.1.1 Device Capabilities

The first node in device capabilities is different for different device types.

For HR Family:

```
<Capability>
   <PrintHead>
      <MaxNum>2</MaxNum>
      <Kind>HP-TIJ</Kind>
   </PrintHead>
   …
</Capability>
```

For Laser Family:

```
<Capability>
   <LaserUnit>
      <MaxNum>1</MaxNum>
      <Kind>CO2</Kind>
   </LaserUnit>
   …
</Capability>
```

For DOD Family:

```
<Capability>
   <Jet>
      <MaxNum>512</MaxNum>
      <Kind>DOD</Kind>
      <Interfaces>
         <Interface id="0">
            <MaxNum>256</MaxNum>
         </Interface>
         <Interface id="1">
            <MaxNum>256</MaxNum>
         </Interface>
      </Interfaces>
   </Jet>
   …
</Capability>
```

For SC Family:

```
<Capability>
   <Jet>
      <MaxNum>48</MaxNum>
      <Kind>CIJ</Kind>
      <Interfaces>
         <Interface id="0">
            <MaxNum>48</MaxNum>
         </Interface>
      </Interfaces>
   </Jet>
   …
</Capability>
```

For GK Family:

```
<Capability>
   <PrintHead>
     <MaxNum>4</MaxNum>
     <Kind>GK</Kind>
   </PrintHead>
       ...
 </Capability>
```

For UP Family:

```
<Capability>
 <PrintHead>
  <MaxNum>4</MaxNum>
  <Kind>UP</Kind>
 </PrintHead>
      ...
</Capability>
```

The nodes describe the cababilities of the connected REA JET device:

```
<Capability>
   ...
   <Job>
      <MaxNum>1</MaxNum>
   </Job>
   <Group>
      <MaxNum>1</MaxNum>
   </Group>
   <ProductSensor>
      <MaxNum>1</MaxNum>
   </ProductSensor>
   <DigitalInput>
      <MaxNum>12</MaxNum>
      <EventDetector>
         <MaxNum>8</MaxNum>
      </EventDetector>
      <ProductSensorGenerator>
         <MaxNum>1</MaxNum>
      </ProductSensorGenerator>
   </DigitalInput>
   <DigitalOutput>
      <MaxNum>8</MaxNum>
      <ProgrammablePulseGenerator>
         <MaxNum>4</MaxNum>
      </ProgrammablePulseGenerator>
   </DigitalOutput>
   <GUIType>
      <Type>Hyperion</Type>
      <Shape>Hyperion</Shape>
   </GUIType>
</Capability>
```

15.1.1.2 Add-On

The node <AddOn> is optional in DEVICEINFO and is only used, when any add-on software is installed.

The part for a software add-on has the following form:

```
<AddOn>
   <Name>titan-add-on-xyz</Name>
   <Version>3.90-66895</Version>
   <Description>
      Some additional software functionality
   </Description>
</AddOn>
```

15.1.2 Command GETPRODUCTIONDATA

The following command can be used to query the production data.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETPRODUCTIONDATA" id="mynewid">
         <Data/>
      </Command>
   </REA-PI>
</REA-JET>
```

<Data> is empty in the request, note the use of the shorthand <Data/> for an empty XML element.

This command is available with a Firmware-Version ≥ 3.50.

```
<REA-JET>
   <REA-PI>
      <Command name="GETPRODUCTIONDATA" id="mynewid">
         <Data>
            <SerialNumber>yymmnnn-6</SerialNumber>
            <ArticleNumber>054.120.024</ArticleNumber>
            <Printhead id="0">
               <SystemPrintCounter>76047</SystemPrintCounter>
            </Printhead>
            <Printhead id="1">
               <SystemPrintCounter>45400</SystemPrintCounter>
            </Printhead>
         </Data>
      </Command>
      <Status>
         <Domain>99</Domain>
         <Code>0</Code>
         <Message>Query production data successfull</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The response carries information about serial number, articlenumber and number of prints for each printhead since initialization of device.

15.1.3 Command SETDEVICEPROPERTY

This command is used to set the specific control functions.
All properties can be changed separately by excluding unwanted parameters from the command. Only included parameter nodes will be changed on the device.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="SETDEVICEPROPERTY" id="mynewid">
      <Data>
          <Laserunit LaserUnitId="laserid">
         <Shutter status="open"/>
         <Pilotlaser status="active"/>
        </Laserunit>
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

Valid values are:

Laserunit LaserUnitId	<0..n> (countig starts with 0)
Shutter status	“open”, “closed”
Pilotlaser status	“active”,”inactive”

The corresponding response will inform about success or failure.

```
<REA-JET>
  <REA-PI>
    <Command name="SETDEVICEPROPERTY" id="mynewid">
      <Data/>
    </Command>
    <Status>
      <Domain>99</Domain>
      <Code>0</Code>
      <Message></Message>
    </Status>
  </REA-PI>
</REA-JET>
```

15.1.4 Command GETTIMEZONE

This command can be used to query the actual set time zone info.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="GETTIMEZONE" id="mynewid">
      <Data />
    </Command>
  </REA-PI>
</REA-JET>
```

The corresponding response will inform about the actual set time zone info.

```
<REA-JET>
  <REA-PI>
    <Command name="GETTIMEZONE" id="mynewid">
      <Data>
        <Region>Europe</Region>
        <Zoneinfo>Bratislava</Zoneinfo>
      </Data>
    </Command>
    <Status>
      <Domain>99</Domain>
      <Code>0</Code>
      <Message></Message>
    </Status>
  </REA-PI>
</REA-JET>
```

15.1.5 Command SETTIMEZONE

This command is used to set a new time zone with region and the ability to automatically change daylight saving time.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="SETTIMEZONE" id="mynewid">
      <Data>
        <Region>Europe</Region>
        <Zoneinfo>Berlin</Zoneinfo>
        <SummerWinterTimeChangeover>True</SummerWinterTimeChangeover>
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

The corresponding command will inform about success or failure.

```
<REA-JET>
  <REA-PI>
    <Command name="GETZONEINFO" id="mynewid">
      <Data/>
    </Command>
    <Status>
      <Domain>99</Domain>
      <Code>0</Code>
      <Message></Message>
    </Status>
  </REA-PI>
</REA-JET>
```

15.1.6 Command GETDATETIME

This command can be used to query the actual set date and time of the onboard clock of the device.

```
<REA-JET>
   <REA-PI>
      <Command name="GETDATETIME" id="mynewid">
         <Data/>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message></Message>
      </Status>
   </REA-PI>
</REA-JET>
```

The corresponding command will inform about the actual set date and time.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETDATETIME" id="mynewid">
         <Data>
            <Year>2010</Year>
            <Month>2</Month>
            <Day>24</Day>
            <Hour>10</Hour>
            <Minute>36</Minute>
            <Second>37</Second>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

15.1.7 Command SETDATETIME

Use the following command to set the internal clock on the device. The <Data> node of this command contains several sub nodes to specify year, month, day, hour, minute and second, respectively, of the time to set.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="SETDATETIME" id="mynewid">
      <Data>
        <Year>2010</Year>
        <Month>2</Month>
        <Day>20</Day>
        <Hour>11</Hour>
        <Minute>55</Minute>
        <Second>55</Second>
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

The response from the server indicates whether the operation was successful:

```
<REA-JET>
  <REA-PI>
    <Command name="SETDATETIME" id="mynewid">
      <Data />
    </Command>
    <Status>
      <Domain>500</Domain>
      <Code>0</Code>
      <Message></Message>
    </Status>
  </REA-PI>
</REA-JET>
```

15.1.8 Command GETNETWORKCONFIG

This command can be used to query the actual network configuration of the device.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="GETNETWORKCONFIG" id="mynewid">
         <Data />
      </Command>
   </REA-PI>
</REA-JET>
```

The corresponding command will inform about the actual network configuration.

```
<REA-JET>
   <REA-PI>
      <Command name="GETNETWORKCONFIG" id="mynewid">
         <Data>
           <Dhcp>true</Dhcp>
           <IpAddress>192.168.0.2</IpAddress>
           <Subnetmask>255.255.255.0</Subnetmask>
           <Gateway>192.168.0.1</Gateway>
         </Data>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message></Message>
      </Status>
   </REA-PI>
</REA-JET>
```

15.1.9 Command SETNETWORKCONFIG

Use the following command to set the network configuration on the device.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="SETNETWORKCONFIG" id="mynewid">
         <Data>
           <Dhcp>False</Dhcp>
           <IpAddress>192.168.1.10</IpAddress>
           <Subnetmask>255.255.255.0</Subnetmask>
           <Gateway>192.168.1.1</Gateway>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The response from the server indicates whether the operation was successful:

```
<REA-JET>
   <REA-PI>
      <Command name="SETNETWORKCONFIG" id="mynewid">
         <Data/>
      </Command>
      <Status>
         <Domain>500</Domain>
         <Code>0</Code>
         <Message></Message>
      </Status>
   </REA-PI>
</REA-JET>
```

16 Generic event protocol

16.1.1 Command TRIGGERGENERICEVENT

Request the device to trigger a generic event with the specified content.

This command is available with a Firmware-Version ≥ 3.90.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="TRIGGERGENERICEVENT" id="myid">
      <Data>
        <Identifier>MyEvent</Identifier>
        <Raw> … </Raw>
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Identifier`	Text specifying generic identifier for which an event will be raised
`Raw`	Raw content of the event to be send

The response from the device indicates whether the operation was accepted, verified and triggered successfully:

```
<REA-JET>
  <REA-PI>
    <Command name="TRIGGERGENERICEVENT" id="myid">
      <Data/>
    </Command>
    <Status>
      <Domain>500</Domain>
      <Code>0</Code>
      <Message>Generic event was successfully triggered</Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies no parameter.

In the node <Status> an error code is specified whether the command is successful or not.

16.1.2 Event GENERICEVENT

The device raises this event when a generic event was triggered.

This event requires a subscription and has event parameters.

This event is available since firmware version ≥ 3.90.

16.1.2.1 Subscription

The subscription is effective during a connection.

The subscription of the event can either be terminated by calling the unsubscribe function or will be terminated automatically when the network connection is closed.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="SUBSCRIBE" id="myid">
      <Data>
        <EVENT>GENERICEVENT</EVENT>
        <EVENT ARGS name="Identifier" value=”MyEvent" />
        <Identifier>MyEvent</Identifier>
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Event` Name of the event to do the subscription for

`Identifier` Text specifying generic identifier to do the subscription for

The response from the device indicates whether the subscription was successful.

```
<REA-JET>
  <REA-PI version="3.9">
    <Command name="UNSUBSCRIBE" id="myid">
      <Data>
        <EVENT>GENERICEVENT</EVENT>
        <EVENT ARGS name="Identifier" value=”MyEvent" />
      </Data>
    </Command>
  </REA-PI>
</REA-JET>
```

`Event` Name of the event to do the subscription for

`Identifier` Text specifying generic identifier to do the un-subscription for

The response from the device indicates whether the un-subscription was successful.

16.1.2.2 Event parameters

```
<REA-JET>
  <REA-PI>
    <Command name="GENERICEVENT" id="">
      <Data>
        <Identifier>MyEvent</Identifier>
         <Raw>…</Raw>
      </Data>
    </Command>
    <Status>
      <Domain>domain</Domain>
      <Code>code</Code>
      <Message>message</Message>
    </Status>
  </REA-PI>
</REA-JET>
```

The node <Data> specifies the parameter:

`Identifier`	Generic identifier for which the event has been raised
`Raw`	Application specific payload for the namely event as raw content

17 Device File Transfer Functions

17.1.1 Command LISTDIRECTORY

This command is available since FW Version 3.2.

Use the following command to get a list of files or directories from device. This function is not depending on the existence of the device share ‘rea-jet’.

```
<REA-JET>
   <REA-PI version="3.9">
      <Command name="LISTDIRECTORY" id="35">
         <Data>
            <Directory value="jobs"></Directory>
         </Data>
      </Command>
   </REA-PI>
</REA-JET>
```

The response from the server indicates whether the operation was successful and contains the resulting file- and directory list:

```
<REA-JET>
   <REA-PI>
      <Command name="LISTDIRECTORY" id="35">
         <Data>
            <Elements>
               <Element type="file" value="demojob 1ph.job"/>
               <Element type="file" value="demojob 2ph.job"/>
               <Element type="file" value="demojob 3ph.job"/>
               <Element type="file" value="demojob_4ph.job"/>
            </Elements>
         </Data>
      </Command>
      <Status>
         <Domain>600</Domain>
         <Code>0</Code>
         <Message>successful executed</Message>
      </Status>
   </REA-PI>
</REA-JET>
```

18 Appendix A1: Error Codes

Error code (Domain/Code)	Domain / Code	Description
CODE_OK	ANY/0	no error
CODE_FAILED	ANY/2	failed for unknown reason
CODE_UNDEFINED	ANY/3	unknown error
CODE_INVALID_PARAMS	ANY/4	invalid parameters
CODE_FATAL	ANY/5	fatal error, device will restart
CODE_MEMORY	ANY/6	a memory exception occurred
DOMAIN_GENERIC		
CODE_GENERIC	99/99	Generic error occurred
DOMAIN_SYSTEM_MISC		
CODE_FILE_NOT_FOUND	500/10	file not found
CODE_FILE_OPEN_FAILED	500/11	cannot open file
CODE_SCRIPT_EXECUTION_FAILED	500/20	execution of script or sub-process failed
CODE_SCRIPT_EXECUTION_DENIED	500/21	execution of script or sub-process denied
CODE_SCRIPT_FIRMWAREUPDATE_FAILED	500/30	execute script for firmware update failed
CODE_SCRIPT_FIRMWAREUPDATE_DENIED	500/31	execute script for firmware update denied
CODE_SCRIPT_FIRMWAREUPDATE_WHILEPRINTING	500/32	execute script for firmware update while printing
CODE_SCRIPT_FIRMWAREUPDATE_IN_PROGRESS	500/33	process of firmware update is in progress
CODE_SCRIPT_FACTORYDEFAULT_FAILED	500/35	execute script for factory default failed
CODE_SCRIPT_FACTORYDEFAULT_DENIED	500/36	execute script for factory default denied
CODE_SCRIPT_FACTORYDEFAULT_WHILEPRINTING	500/37	execute script for factory default while printing
CODE_INVALID_NETWORKCONFIG	500/40	invalid network configuration to set
CODE_FONTUPDATE_FAILURE	500/80	Update fonts failed
CODE_FREETYPE_FAILED	500/81	Resource freetype not available or failed
CODE_INVALID_FONT	500/82	Font file is not a valid freetype usable font
CODE_XML_INVALID_DIOVERSION	500/90	Xml-IOConfiguration version not supported
DOMAIN_ENGINE_LABEL		
CODE_INVALID_TAG	101/10	Invalid label tag

CODE_MISSING_RESOURCE	101/20	Label used resource missing
CODE_MISSING_FONT	101/30	Label used font missing
CODE_INVALID_LABEL_OBJECT_WIDTH	101/31	Invalid width of label object
CODE_INVALID_LABEL_OBJECT_HEIGHT	101/32	Invalid height of label object
CODE_LABELOBJECT_NOTFOUND	101/40	Label object not found / available in label
CODE_OBJECTCONTENT_NOTFOUND	101/41	Object content not found / available in label object
CODE_PERMISSION_DENIED	101/90	access to label or label tag not granted
CODE_CONTENT_UNCHANGED	101/96	object content set without any changes
CODE_REQUEST_STOPPRINT	101/99	object content set produces a stop condition/request for printing
DOMAIN_ENGINE_PRINTCONTROL		
CODE_JOB_STILL_RUNNING	102/10	Job is still running
CODE_JOB_NOT_RUNNING	102/11	Job is not running
CODE_NO_JOB_ASSIGNED	102/12	There is no active assigned job
CODE_NO_GROUP_ASSIGNED	102/13	Job contains no group
CODE_JOB_NOT_EXISTS	102/14	Job does not exist
CODE_NO_LABEL_ASSIGNED	102/15	Group contains no label
CODE_GROUP_NOT_ASSIGNED	102/16	Tried action on not assigned group
CODE_INVALID_INKINFO	102/17	Invalid ink info
CODE_HEAD_NOT_READY	102/18	Printhead not ready
CODE_SHAFTENCODER_NOT_SET	102/19	Shaft encoder not set
CODE_ENTITY_ACTIVE	102/21	A entity of the printing system is active and cannot be (re-)parameterized
CODE_RENDER_FAILED	102/22	Rendering failed, no label-image from renderer available
CODE_JOB_PURGING	102/23	Purging already running
CODE_PURGING_HEAD_NOT_FOUND	102/24	Please assign a job which includes the printhead to be purged
CODE_PURGING_DUPLICATE_HEAD_FOUND	102/25	Can not purge with double existing printheads/cartridges
CODE_PURGING_PURGE_WITHOUT_HEADS	102/26	Can not purge without any printhead / cartridge
CODE_PURGING_JOB_IS_RUNNING	102/27	Can not purge with a printhead included in a running job
CODE_PURGING_CARTRIDGE_NOT_CONNECTED	102/28	Please insert a cartridge or lock the inserted cartridge
CODE_GROUP_NOT_EXISTS	102/29	Group does not exist
CODE_GROUP_NOT_ACTIVE	102/30	Group must not be active to perform

CODE_GROUP_ACTIVE	102/31	Group have to be active to perform
CODE_GROUP_NO_IMAGEBUFFER	102/32	No buffer to store the image data for group
CODE_GROUP_NO_PRINTHEAD	102/33	Group have to contain at least one printhead
CODE_GROUP_HARDWARE_NOT_SUPPORTED	102/34	Desired hardware not supported
CODE_GROUP_DUPLICATE_PRINTHEAD	102/35	Group must not contain printheads with identical IDs
CODE_GROUP_PURGING	102/36	Purging in progress. Job can not be started!
CODE_EXTERNAL_STOP_CONDITION	102/37	Group/job can not be started within external stopped condtion
CODE_PRINTIMAGE_PRINTCOUNT_EXCEEDED	102/38	No buffer data to print reasoned by printcount exceeded
CODE_JOB_DUPLICATE_GROUP	102/40	Job must not contain groups with not unique name
CODE_SPITTING_INK_TYPE	102/50	Type of ink for spitting doesn't match with settings
CODE_INK_INVALID_ERROR_WARNING_LEVEL	102/51	levels for ink warnings and ink error doesn't match together
CODE_SPEEDSENSORPULSELOST	102/70	Print speed too fast, max print pulses exceeded
CODE_SPEEDSENSORTOOFAST	102/71	Print speed too fast, speed sensor pulses exceeded for resolution
CODE_SPEEDSENSORSLICESLOST	102/72	Print speed too fast, speed sensor pulses for generating slices exceeded
CODE_GROUP_LASERPEN_NOT_FOUND	102/80	Needed set of laser pen parameter in group not found
CODE_GROUP_PARAMETER_ERROR	102/81	Erroneous parameters
CODE_CHIPREQUIRED	102/90	Required cartridge chip not present
DOMAIN_ENGINE_PARSER		
CODE_PARSE_EXCEPTION	103/10	Common error in parsing xml
CODE_XML_SYNTAX	103/11	Error in xml syntax
CODE_XML_INVALID_ROOT	103/12	xml-document doesn't contain valid root node
CODE_XML_INVALID_JOBFILE	103/14	file of xml-document for job or installation is invalid (or not found)
CODE_XML_INVALID_JOB	103/15	xml-document invalid or missing job node
CODE_XML_INVALID_JOBVERSION	103/16	Job version is not supported
CODE_XML_INVALID_INSTALLATIONVERSION	103/17	Installation version is not supported
CODE_XML_LABELFILE_NOT_FOUND	103/19	Label could not be found
CODE_XML_INVALID_LABEL	103/20	xml-document invalid or missing label node
CODE_XML_INVALID_LABELVERSION	103/21	xml-document Label version not supported
CODE_LOGIC_EXCEPTION	103/40	Logical error in xml-tag
CODE_TAG_LABEL	103/41	Error in xml-tag label

CODE_TAG_LAYOUT	103/42	Error in xml-tag layout
CODE_TAG_OBJECT	103/44	Error in xml-tag object
CODE_TAG_RENDERER	103/45	Error in xml-tag renderer
CODE_RENDERER_NOTSUPPORTED	103/46	Xml-tag renderer type (of label) not supported
CODE_TAG_CONTENT	103/47	Error in xml-tag content (of label object)
CODE_CONTENT_NOTSUPPORTED	103/48	Xml-tag content type (of label object) not supported
CODE_RENDERER_NEEDSCONTENT	103/49	Xml-tag renderer type needs content
CODE_TAG_FORMAT	103/51	Xml-tag format doesn't match specification
CODE_TAG_UNRESOLVED_REFERENCE	103/54	Referenced content can not be found
CODE_TAG_INSTALLATION	103/60	Error in xml-tag installation
CODE_FEATURE_EXCEPTION	103/70	Unknown feature
DOMAIN_ENGINE_RENDERER		
CODE_CANNOT_RENDER	104/10	Rendering print image failed
CODE_RENDERTHREAD_NOT_STARTED	104/11	Cannot create or start render thread
CODE_PURGETHREAD_NOT_STARTED	104/12	Cannot create or start purge thread
CODE_INVALID_IMAGEFACTORY	104/29	Not valid renderer reasoned by invalid image factory
CODE_NO_RENDERER_CREATE	104/30	Cannot create renderer
CODE_NO_TEXTRENDERER_CREATE	104/31	Cannot create renderer for text
CODE_NO_IMAGERENDERER_CREATE	104/35	Cannot create renderer for images
CODE_NO_BARCODERENDERER_CREATE	104/36	Cannot create renderer for barcodes
CODE_NO_GRAPHICRENDERER_CREATE	104/40	Cannot create renderer for graphical objects
CODE_INITIALIZE_FREETYPE	104/50	Cannot initialize freetype library
CODE_INITIALIZE_TBARCODE	104/60	Cannot initialize tbarcode library
CODE_CREATE_BARCODE_FAILED	104/62	Cannot render barcode object
CODE_MISSING_BITMAP_FILE	104/79	Bitmap file could not be found
CODE_INVALID_BITMAP_FORMAT	104/80	Rendering failed due to invalid format of source image
CODE_INVALID_BITMAP_SIZE	104/81	Rendering failed due to invalid size of source image
CODE_PRINTCOUNT_EXPIRED	104/83	Cannot render image because of expired print count
CODE_INVALID_RENDEREVENT	104/84	Invalid count event trigger
CODE_INVALID_SHIFTCODE	104/90	Render failed on invalid shiftcode
CODE_INSUFFICIENT_CODELISTENTRIES	104/110	Code list contains insufficient entries
CODE_INVALID_INCREMENT_ZERO	104/120	Invalid increment, must not be 0, zero
CODE_INVALID_INCREMENT_DIRECTION	104/121	Invalid increment, violate start/end direction

CODE_INVALID_STARTEND_EXCEEDED	104/122	Invalid counter have to be in range of start/end condition
CODE_INVALID_STARTEND_CONDITION	104/123	Invalid start/end condition, must not be equal
DOMAIN_FPGA_MISC		
CODE_LABEL_TOO_LATE	200/10	Label data too late for FPGA
CODE_SHAFTENCODER_TOO_FAST	200/11	Pulses from shaft encoder too fast
CODE_SHAFTENCODER_TOO_UNSTEADY	200/12	Unsteady pulses from shaft encoder
CODE_INVALID_SHAFTENCODER_PARAM	200/20	Invalid shaft encoder parameter
CODE_NO_SUCH_HARDWARE	200/30	Hardware subsystem missing
CODE_NUMBER_LAYERS_EXCEEDED	200/40	Number of layers exceeded / not supported
CODE_INVALID_CONFIGURATION	200/50	Invalid FPGA configuration
CODE_COPY_FPGA_IMAGE	200/60	Failed to copy image to FPGA
CODE_OPERATION_AND_2EDGE	200/70	Logical AND-operation with edges not allowed
CODE_OPERATION_OR_2EDGE	200/71	Logical AND-operation with falling edges not allowed
DOMAIN_HEAD		
CODE_NOTCONNECTED	300/11	Not all printheads, which are referenced from the job, are connected
CODE_VERRIEGELUNG_OPEN	300/12	Cartridge is not locked
CODE_PENDRIVER_INIT_FAILED	300/13	Pendriver init failed
CODE_NO_CARTRIDGE	300/14	Printhead has no cartridge
CODE_INK_EMPTY	300/15	Ink empty
CODE_TEMPERATURE_HIGH	300/16	Printhead temperature is too high
DOMAIN_CHIPCARD		
CODE_NO_CHIP	310/10	Cartridge with no chip
CODE_READWRITE_FAILED	310/11	Read write failed
DOMAIN_COMMUNICATION		
CODE_SUBSCRIPTION_FAILED	400/10	Event subscription failed
CODE_XML_INVALID_REAPIVERSION	400/20	Xml-document REA-PI version not supported
CODE_VERSION_ALREADY_SELECTED	400/21	Version already selected
CODE_REAPICOMMAND_INVALID	400/30	REA-PI xml-command is invalid
CODE_REAPICOMMAND_INCOMPLETE	400/31	REA-PI xml-command is missing parameters

DOMAIN_FILESYSTEM		
CODE_ACCESS_DENIED	600/10	Access to file or directory denied
CODE_FILE_LOCKED	600/11	File is locked
CODE_INVALID_PATH	600/12	Invalid path
CODE_INVALID_FILENAME	600/13	Invalid filename
CODE_FILE_ALREADY_EXISTS	600/14	File already exists
CODE_COULDNOTGETFILE	600/15	Could not get file
CODE_COULDNOTWRITEFILE	600/16	Could not create or write file
CODE_TRANSMISSION_FAILED	600/20	Transmission failed
CODE_MD5SUM_ERROR	600/21	Error in MD5 check sum
DOMAIN_LASER_MISC		
CODE_INVALID_CORRECTION_FILE	700/10	Invalid correction file
CODE_NUMBER_LABELOBJECTS_EXCEEDED	700/20	Number of LabelObjects exceeded / not supported
CODE_LABELOBJECT_DOES_NOT_EXIST	700/21	Label object does not exist
CODE_INVALID_LABELOBJECT_LIST	700/22	Invalid list of label ojects
CODE_OVERTEMPERATURE_FAULT	700/30	Overtemperature fault
CODE_SECURITY_LOCK_OPEN	700/31	Security lock open
CODE_LASER_TUBE_NOT_READY	700/32	Laser tube not ready
CODE_SAFETY_CIRCUIT_FAILURE	700/33	Safety circuit failure

Additional to error codes

In some cases one or more warnings can occure.

DOMAIN_WARNINGS		
CODE_LABEL_EXCEEDS_PRINT	800/201	Label exceeds the printable height