



PANNIER UNIVERSAL DATA STREAM PROTOCOL

This document describes the structure of Pannier's Universal Data Stream protocol and each of the possible parameters. The Universal Data Stream is used by the host systems to send tag data for each print job. The Universal Data Stream provides the user with a means to select any format loaded in the printer, supply specific data for any object in that format, print a series of sequenced tags and/or print multiple copies of the same tag, have an Email sent when the last tag of a print job is complete, and have the tags presented for break-off after the last tag of a print job is complete. A sample tag and two sample data streams are provided as examples for printing tags.

Described below is the structure of the data stream protocol. Repeatable parameters are denoted by the double [] brackets and optional parameters are denoted by the single [] brackets framing them. **Bold** text indicates required text if that parameter is used. The system will maintain the last values passed for an Object until the application is shut down. Therefore, if an object is ever to be in the data stream, it should always be included along with each object parameter (**M**, **V**, and **I**) that will be used.

{*Format Name*~

[**N***value*~]

[**D***value*~]

[**E***value*~]

[**P***value*~]

[[*Object Name*~

[**M***value*~]

[**V***value*~]

[**I***value*~]

`]]

}

Symbol	Parameter	Description
{	Start of Data Stream Identifier	The left cursive bracket ({) character is used to identify the beginning of each data stream.
<i>Format Name</i>	Name of Format to Print	Identifies the name of the format that you want to print.
N <i>value</i>	Number of Sequenced Tags Parameter	The N identifies the parameter that states the number of sequenced tags that are to be produced by this data stream. The value following this identifier is the number of sequenced tags to mark of the specified tag. If this parameter is included it must contain a value from 1 to 2000. If a value greater than 1 is entered then a series of tags (quantity specified by this value) will be produced by the printer from this one data stream received from the host system. The first tag will contain the values specified in the data stream. Any object that contains a non-zero value for the Increment Parameter will be modified by that parameter value after each sequenced tag is printed. If this parameter is not in the data stream, then its value is 1 by default.

PANNIER CORPORATION

207 Sandusky Street • Pittsburgh PA 15212-5823 USA

15-Apr-08

412-323-4900 tel • 412-323-4962 fax • sales@pannier.com • www.pannier.com

Page 1 of 5

Dvalue	Number of Duplicate Tags (Copies) Parameter	The D identifies the parameter that states the number of copies to print for each tag. The value following this identifier is the number of copies of this tag to print. If this parameter is included it must contain a value from 1 to 100. If a series of sequenced tags are being printed, then this quantity of duplicate tags will be printed before the objects are incremented or decremented for the sequenced tags. If this parameter is not in the data stream, then its value is 1 by default.
Evalue	Send Email after last tag of job prints Parameter	The E identifies the parameter that is one half of the two part flag that enables the transmission of an Email message to indicate the completion of this print job. The configuration file contains the other half of this two part flag. If both halves are enabled, then an Email message will be sent to the designated Email address after the last tag of this print job is complete. A value of 0 disables this half of the flag. A value of 1 enables this half of the flag. If this parameter is not in the data stream, then its value is 0 by default.
Pvalue	Present Tags for Break-off after last tag of job prints Parameter	The P identifies the parameter that indicates whether to present the tags for break-off after the last tag of the print job is complete. If the present tag method in the configuration file is set to manual, then the data stream can determine when the tags should be presented for break-off by including P1 as a parameter. If the present tag method in the configuration file is set to automatic, then this parameter is ignored so it should not be included. The present tag configuration parameters indicating number of advances, present tag position, and recovery method will be utilized when tags are presented through the data stream. A value of 1 enables tag presenting for this job. A value of 0 defaults to the configuration parameters for presenting tags. If no present tag method is indicated in the data stream, then the present method in the configuration file will be used.
Object Name	Object Name in Format File	This identifies the object name in the format whose print value, increment value, and/or mark value that you want to modify with the following three parameters. If an object is not included in the data stream, it will use the last value supplied by a data stream or the value that is specified in the tag format (if it was never included in any data stream.)
Vvalue	Object Value Parameter	The V identifies the parameter that states the new value for the specified object. The value following this identifier is the value that will be marked on the tag for the object. This parameter is required for text, barcode, and 2D Code type field objects. It should not be supplied for any other object types since they do not have a displayed value associated with them.
Ivalue	Object Value Increment Parameter	The I identifies the parameter that states the increment/decrement value for the specified object. The value following this identifier is the value to automatically increment (positive number) or decrement (negative number) a text or barcode type object. This is the value that will be added/subtracted from this object' value after each sequenced tag is printed. If the value for this object does not change, then you should omit this parameter since it defaults to 0.
Mvalue	Object Mark Parameter	The M identifies the parameter that states the mark state of the specified object. The value following this identifier indicates whether the object should be printed on the tag. If an object is always marked then this parameter should be omitted since its initial default value is to mark.. If an object is marked only some of the time on a particular format then this parameter is required all of the time since the last value for marking is maintained from tag to tag. This parameter is commonly used for picture objects (bitmaps or vector images) that are only printed on a tag format under certain conditions. The only valid values for this parameter are: 1 – Mark and 0 - No Mark. The initial state for all objects is to mark.

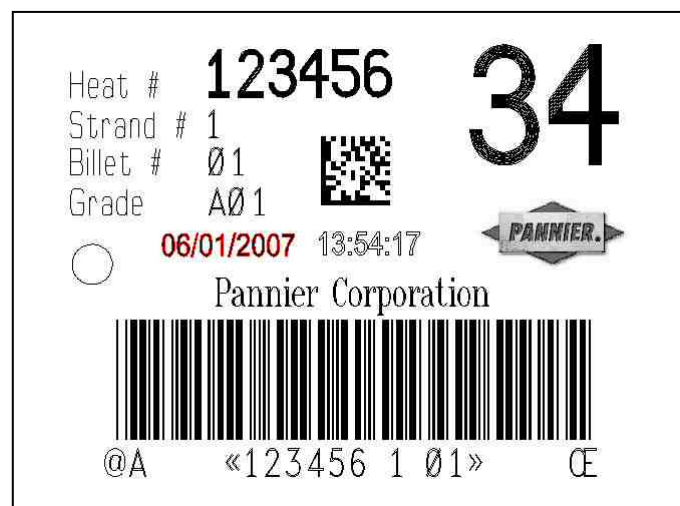
PANNIER CORPORATION

207 Sandusky Street • Pittsburgh PA 15212-5823 USA

~	End of Parameter Identifier	The tilde symbol (~) character is used to denote the end of each parameter value.
`	End of Object Identifier	The backwards quote (accent grave) (`) character is used to denote the end of the format parameters and the end of each object's parameters.
}	End of Data Stream Identifier	The right cursive bracket (}) character is used to indicate the end of each data stream.

The following data stream sample is representative of the data stream that you will need to send in order to produce the tag format(s) configured for you. Each object's parameters have been placed on a new line for readability and numbered for reference. The actual data stream sent to the printers must not contain any unwanted spaces. Spaces that are part of the variable data are allowed. Carriage returns <CR> and line feeds <LF> must be used carefully. If they are placed within the value then they will become part of that value. If tier text is required, then <CR><LF> are needed in the value. They must be placed between each character so the characters will be placed one on top or the other. If the <CR><LF> is placed outside of a value then they will be ignored. All control characters except for Group Separator (GS), Record Separator (RS), and End of Transmission (EOT) are ignored. If UID is not enabled on this system, then the GS, RS, and EOT will also be ignored. Extended ASCII characters are allowed and will be printed as long as the font for that object contains the character.

The first set of parameters designates the tag format name, the number of tags to produce, the number of copies of that tag format, one half of enabling an email after the print job is complete, and whether to present tags for break-off after the last tag of the print job is complete. If a format parameter is not shown, then the default value for that parameter will be used. The subsequent sets of parameters designate the objects that need to change, along with their respective value, increment value, and/or marking value. If an object parameter is not shown, then the last value provided for that parameter will be used. If no value was ever provided for a format parameter, then the initial default value for that format parameter will be used. The underlined text in the examples below designates the variable data for that tag. All other data in the data stream for that tag is fixed text.



PANNIER CORPORATION

207 Sandusky Street • Pittsburgh PA 15212-5823 USA

412-323-4900 tel • 412-323-4962 fax • sales@pannier.com • www.pannier.com

The following is an example of the data stream for the **Pannier1** tag format shown above. It contains values for all object fields that change. The object fields that are not addressed by the data stream are static values that never change on the tag, therefore they do not need to be supplied in the data stream. This data stream will print one tag using the values in the data stream.

```

1.    {Pannier1~`
2.    txtHeat~V123456~`
3.    txtStrand~V1~`
4.    txtBillet~V01~`
5.    txtGrade~VA01~`
6.    dmDM~V123456-1-01~`
7.    txtSort~V34~`
8.    bmpPannierLogoBW~M0~`
9.    bmpPannierLogoGS~M1~`
10.   pltPannierLogo~M0~`
11.   txtCompany~VPannier Corporation~`
12.   bc3of9~V123456 1 01~`
13.   txtSpecial1~V@A~`
14.   txtBCVal~V«123456 1 01»~`
15.   txtSpecial2~VÆ~`
16.   }
```

The following data stream example demonstrates the capability of sending down one data stream to have multiple tags printed. This data stream will produce 50 tags. The data stream will produce two copies of 25 different tags where the bc3of9, txtBarcodeValue, and txtBillet objects increment by 1 after the two copies of each tag are printed. The data stream indicates that the tags are to be presented for break-off after the last tag is printed. The Email parameter is not indicated in this data stream so no emails will be sent. The first tag will utilize the values sent in the data stream and then increment/decrement the object's values for each following tag. When incrementing, character types are maintained by column position. Therefore, a "9" will roll over to a "0", a "z" will roll over to an "a", and a "Z" will roll over to an "A". Decrementing is the exact opposite. Columns that do not contain incrementable characters will maintain that character. Any column value that rolls over will search left until it finds the first incrementable character.

```

1.    {Pannier1~N25~D2~P1~`
2.    txtHeat~V123456~`
3.    txtStrand~V1~`
4.    txtBillet~V01~I1~`
5.    txtGrade~VA01~`
6.    dmDM~V123456-1-01~I1~`
7.    txtSort~V34~`
8.    bmpPannierLogoBW~M0~`
9.    bmpPannierLogoGS~M1~`
10.   pltPannierLogo~M0~`
11.   txtCompany~VPannier Corporation~`
12.   bc3of9~V123456 1 01~I1~`
13.   txtSpecial1~V@A~`
14.   txtBCVal~V«123456 1 01»~I1~`
15.   txtSpecial2~VÆ~`
16.   }
```

PANNIER CORPORATION

207 Sandusky Street • Pittsburgh PA 15212-5823 USA

Oversized (Two-Part) Tags

If the tag size you want to use is longer than the marking window of the laser printer you are using, PLUS3 (which is 4.75") / LT-21 (which is 8.0"), then you must create two paired formats in order to mark on the whole tag. A single tag format can only mark on the first or last 4.75" / 8" of the tag. Tags can be positioned at the top of tag or bottom of tag positions. Tags that are less than 9" / 16" in length have an overlap area in the middle, which can be marked in either the top or bottom of tag position. Any single object (text, barcode, graphic, etc.) must be marked in one of those halves. No single object can be split between the two halves. A single text value that is larger than the marking window can be split up into multiple text objects if the data can be sent from the host system or entered manually as a split up value.

Formats that are used for two part tags must be named the same except for the last 4 characters of the name. The last four characters must use the reserved characters "1of2" and "2of2". When those last four characters are seen in a tag format name, this designates that these formats are part of a two-part tag. If we are developing a two-part test tag the names of the two formats could be "LargeTag1of2.mkh" and "LargeTag2of2.mkh".

The object names between the two formats must be unique. No two objects can have the same name or you will receive an error when you try to print a two-part tag manually. If multiple oversized tags, emails, or presented tags are desired, those format parameters must be specified in the first half (1of2) data stream. The format parameters are ignored in the second half (2of2) data stream.

Data streams for the two-part tags must be sent down in pairs. The data stream for the first half must be sent down first, followed by the data stream for the second half. Even though these are two different formats, the laser printer software waits for data streams for both halves to arrive before it will begin printing the tag. If the data streams are not received in the proper order or only one of them is received, the printer will generate a data stream error.

A data stream example of an oversized tag is shown below. This example will print one oversized tag.

1. {LargeTag1of2~`
2. txtLeftVal~VLARGE TAG 1 OF 2 (LEFT SIDE)~` }
3. {LargeTag2of2~`
4. txtRightVal~VLARGE TAG 2 OF 2 (RIGHT SIDE)~` }

PANNIER CORPORATION

207 Sandusky Street • Pittsburgh PA 15212-5823 USA