Table of Contents:

1 INTRODUCTION

1.1 Purpose

This document describes the organization of Map files. What are the required fields and what they mean. Map files may be created for each protocol of devices (HART, Conventional, or FF).

Map files are used to download test procedures to a self documenting calibrator for a given device type, and to define a common set of parameters for Generic Exports.

There is a set of templates which can give a device developer a structure for creating the map files.

Revision History

2 MAP FILE FORMAT

The Map File translates custom DDL parameters into a common set of parameters used for Calibration Management. These map files are assumed to work for future versions of the device, regardless of Protocol revision (ie these support forward compatibility in a way similar to that DDL supports forward compatibility .

2.1 Overview

The major parts of a MAP file are listed below. Descriptions for each part are provided in the sections after this section.

- Identification
- INPUT BLOCK TYPE
- OUTPUT BLOCK TYPE
- Parameter MAP
- Parameter Special Info
- DEFAULT PARAMETER VALUES
- Formulas
- input(or output) block Units Code DD to Common Map
- Probe Type DD to Common Map
- Number of Wires DD to Common Map
- Pressure Measurement Type DD to Common Map
- Transfer Function Code DD to Common DD Map
- Wave Form Common to DD Map
- Probe Connect DD to Common Map

Definitions/Syntax

[ ]	Are used to label and identify major parts of the MAP file
;	Semi-colons are used to identify comments
<cKey>	This is a common format keyword that AMS needs to cross reference to the device type specific term. AMS searches the MAP file for the key word to determine what is the corresponding term for this device type.
<ddValue>	This is value specific for that device type.
<ddKey>	This is a device type specific keyword that AMS needs to cross reference to the common format term. AMS searches the MAP file for the keyword to determine what is the corresponding common format term.
<cValue>	This is the common format value for a specific device type's keyword.

2.2 [IDENTIFICATION]

This section of the map files contains device specific identification information. All of the values entered in this section must be of hexa-decimal type.

Format:

<cKey>=<ddValue>

2.2.1 "Map File Format Version"

This is now up to version 3. Version 1 was the initial version from 1998. Version 2 added "Use for Export". Version 3 adds "block number" for Fieldbus Support. This is an ASCII representation of a Decimal number

2.2.2 "Manufacturer"

This is a hexadecimal manufacturer ID of the device. This is an ASCII representation of a Hexadecimal number. i.e. 26 Hex = 38 Decimal

2.2.3 "Model"

This is for the model number of the device. This is an ASCII representation of a Hexadecimal number.

2.2.4 "Device Revision"

This is the device revision of the device. This is an ASCII representation of a Hexadecimal number.

2.2.5 "DD Revision"

This is the ddl revision. This is an ASCII representation of a Hexadecimal number.

2.2.6 “BlockNumber”

This identifies which block number this map file corresponds to. For Hart and Conventional devices this field will be “0” (zero). For Fieldbus devices This will be a number from 1 to the maximum value defined by the Fieldbus Foundation. It is the responsibility of devInfo to ensure that this number specifies a Transducer block. This is an ASCII representation of a Decimal number.

2.2.7 “Use for CalMgmt”

This may equal either “Yes” or “No”.

If “Yes” is specified or there is no “Use for CalMgmt” line, this map file can be used for calibration management.

If “No” is specified, then AMS Calibration Management will default to using:
Generic input block ranged 0-100 %
Generic output block ranged 0-100%
linear relationship
damping = 1.0

2.2.8 “Use for Export”

This may equal “Yes” or “No”.

If “Yes” is specified or there is no “Use for Export” line, this map file can be used for exporting data from AMS database.

If “No” is specified this device will not return values for the Common Parameters feature of Generic Exports (See SRS-356).

2.3 [INPUT (or OUTPUT) BLOCK TYPE]

The “input block type” section of the map file indicates the “input block type” of the device, and the “output block type” section of the map file indicates the “output block type” of the device. These two sections are set up in the same format. “Input block type” section must be specified in a map file, and the “output block type” section may be omitted if the “output block type” is generic. The comment lines (the line that contains a semicolon) are provided for making sense of the numeric value. If the device has more than one “input block type” then it is considered “multivariable” and a “99” should be entered for the “Input block”. If the device is multivariable then a key “Parameter for current PV” needs to be completed. “Parameter for current PV” names the parameter which must be read to determine the current input block type.

The input/output block type codes are as follows:

Generic=1	// input or output
TempRTD=2	// input
TempTC=3	// input
Frequency=4	// input or output
Pressure=5	// input or output

HART=8 // input or output

Switch=9 // output

Multivariable=99

Format:

<cKey>=<ddValue>

Examples of this section:

For a single device:

[INPUT BLOCK TYPE]

Input block=5

For a multi-variable device:

[INPUT BLOCK TYPE]

Input block=99

Parameter for current PV=sensor_type

2.4 [PARAMETER MAP]

This section provides information needed for constructing a moniker for reading and writing the values for all info types which are relevant for the device. The keys consist of all of the info types but only the relevant ones will have a value. This is fine as long as that key is not for the device that the map file is being made for. If the device supports an ItemArray mapping for an info type, the value will be of the form “ItemArray!dynamic_variables!Element!1!Member!xxx”. Otherwise, the value will be of the form “Param!xxx”. In both cases, the xxx would be replaced by the parameter name (actual or mapped, in the case of the item array). Only if the key has an Item Array or Parameter will it be listed in this section.

Format:

<cKey>=<ddValue>

2.4.1 Required Parameters

The following fields must be implemented for Calibration Management to work. Each entry should be associated with a relative moniker which directs the Device Manager from a “Block Object” to a “VariableParameter Object” (See SDD-334).

2.4.1.1 “input block Lower Range Value”

This is the lower end of the input applied to a transmitter when it is being calibrated.

2.4.1.2 “input block Upper Range Value”

This is the upper end of the input applied to a transmitter when it is being calibrated.

2.4.1.3 “input block Units Code”

This is the units of the input upper and lower range values.

2.4.1.4 “output block Lower Range Value”

This is the value at the output while the “input block Lower Range Value” is applied to a transmitter.

2.4.1.5 “output block Upper Range Value”

This is the value at the output while the “input block Upper Range Value” is applied to a transmitter.

2.4.1.6 “output block Units Code”

This is the units of the output upper and lower range values

2.4.1.7 “Transfer Function Code”

This is the relationship between input and output. Most cases this is linear. However, many HART pressure transmitters are set up with a “Square Root” relationship if the user is using “Differential Pressure” to measure Flow.

2.4.1.8 “Damping Value”

This is used by self documenting calibrators to determine how long to wait from the time the input value is changed before recording the output value.

2.4.2 Pressure Specific Parameters

2.4.2.1 “Pressure Measurement Type”

2.4.3 Temperature Specific Parameters

2.4.3.1 “Number of Wires”

2.4.3.2 “Probe Connect”

2.4.3.3 “Probe Type”

2.4.3.4 “Cold Junction Compensation”

2.4.3.5 “Manual Cold Junction Compensation”

2.4.4 Switch Specific Parameters

2.4.4.1 “Switch Number Of Set Points”

2.4.4.2 “Switch Set Point 1”

2.4.4.3 “Switch Set Point 2”

2.4.4.4 “Switch Set Point 3”

2.4.4.5 “Switch Set Point 4”

2.4.4.6 “Switch Set Point 5”

2.4.4.7 “Switch Set Point 6”

2.4.4.8 “Switch Ramp Time”

2.4.4.9 “Switch Wet Dry Contact “

2.4.4.10 “Switch Wet Voltage”

2.4.4.11 “Switch Trip Direction”

2.4.4.12 “Switch Calibration Form 1”

2.4.4.13 “Switch Calibration Form 2”

2.4.4.14 “Switch Calibration Form 3”

2.4.4.15 “Switch Calibration Form 4”

2.4.4.16 “Switch Calibration Form 5”

2.4.4.17 “Switch Calibration Form 6 “

2.4.4.18 “Switch Execute Reset Test”

2.4.4.19 “Switch Deadband”

2.4.5 Frequency Specific Parameters

2.4.5.1 “Amplitude”

2.4.5.2 “Wave Form”

2.4.6 Optional parameters

2.4.6.1 “Date”

2.4.6.2 “Descriptor

2.4.6.3 “Message”

2.4.6.4 “Tag”

2.4.6.5 “Totalizer Units”

2.4.6.6 “Totalizer Value”

2.4.6.7 “Totalizer Pulse Factor”

2.4.6.8 “Totalizer Pulse Factor Units”

2.5 [PARAMETER SPECIAL INFO]

This section indicates any special information about a parameter. The keys are info types names from the previous section. The supported values are “No Associated Parameter”, “Read only”, “Formula” and “No Prompt”. “No Associated Parameter” means that there is no parameter for the info type and there must be an entry in the **[DEFAULT PARAMETER VALUES]** section. “Read only” means that the value can only be read and the write will not be attempted. “Formula” means that a series of parameter or values must be evaluated in order to determine the current values of the info type. See the **Formulas** section for more information. “No Prompt” is used in the case where the info type “Damping Value” does not exist in that certain input block type. In this case no default parameter value is needed and within the code the default value will be set to 0.0. The only info types that use “Read only” is “output block Units Code”.

Format:

<cKey>=<ddValue>

An example of this parameter special information is as follows.

[PARAMETER SPECIAL INFO]

Cold Junction Compensation=No Associated Parameter

output block Lower Range Value=No Associated Parameter

output block Units Code=Read only

output block Upper Range Value=No Associated Parameter

Probe Connect=Formula

2.6 [DEFAULT PARAMETER VALUES]

For any info type which has **[PARAMETER SPECIAL INFO]** of “No Associated Parameter”, a default value for the info type must be provided in this section. The keys are the info types names. If the value for that info type is a float value then a float value must be entered into this section. If the info type is a word value the number entered must be the number that corresponds to the default value. The following has an example of this. The default values for info types can be found in the **Tables** section of this document.

Format:

<cKey>=<ddValue>

Example:

[DEFAULT PARAMETER VALUES]

Cold Junction Compensation=1

output block Lower Range Value=4.0

output block Upper Range Value=20.0

In the previous section it is noticed that the “Cold Junction Compensation” is set to “No Associated Parameter” therefore in this section “Cold Junction Compensation” must be set to a default value. The cold junction compensation for this device is done internally so referring to **Tables section 4.5** Cold Junction Compensation, internal is equal to the value “1”.

2.7 Formulas

If a formula must be used to determine the value of an info type then the following must be constructed. If the formula exists then in the [PARAMETER SPECIAL INFO] section must be the info type set equal to “Formula”. The section names for completing the formula section of the map file consists of the info types followed by “FORMULA PARAMETERS” OR “FORMULA”.

2.7.1 [FORMULA PARAMETERS]

This section is where the fomula’s equation variable (i.e. parameters) are defined. The key values are the actual parameter names and the keys are corresponding formula parameters invented by the mapfile maker. The formula parameter names are an integer preceded by a single ‘#’. Their types may be integer (including long, short, byte, etc.), enumeration values.

2.7.2 [FORMULA]

This section is where the formula for determining the info type is located. Formulas take the following form:

```
If (<logical expression>)
{
        <cKey> =<ddValue>
}
.
.
.
If (<logical expression>)
{
        <cKey> =<cValue>
}
else
{
        <cKey> =<ddValue>
}
```

The formulas consist of at least one “if” statement and only one “else” statement which has to be the last statement. The keywords “if”, “else” are case sensitive and they must be in lower case.

<cKey> is a common format keyword. If the formulas are for Probe Connect, <cKey> is Probe Connect.

<cValue> is a common string. If the formulas are for Probe Connect, <cValue> can be Single Probe, Differential, Not Used, None, Unknown, or Special.

<logical expression> is a C-like infix logical expression. The operands in the logical expression may be non-negative integers and may be the parameters defined in the “FORMULA PARAMETERS” section. The operators can be used in the logical expression are listed below and are all binary. The precedence for these operators is the same as that defined in C language.

- +: arithmetic addition
- -: arithmetic subtraction
- *: arithmetic multiplication
- /: arithmetic division
- !: logical NOT. This is a unary operator.
- <: relational “less than”
- >: relational “greater than”
- !=: logical “not equals”
- <=: relational ‘less than or equals”
- >=: relational “greater than or equals”
- &&: logical AND
- ||: logical OR
- ==: relational “equals”

Note: The ‘-‘ sign is only used for arithmetic subtraction can not be used to indicate a negative number in the formula.

2.7.3 Example Formula #1

The DDL file for the 3044cRev2 contains the following:

```
#define DIFF (sensor_connection == 5 || (sensor_connection == 2 && sensor_type >5 && sensor_type <12))
```

The DIFF refers to Differential. This equation may be used to determine the value of the “Probe Connect” info type since for this device there is not a corresponding parameter. The common values which probe connect can assume are “Differential” and “Single probe”. With that knowledge, we can convert to infix and construct the following MAP file entries:

```
[PARAMETER SPECIAL INFO]

Probe Connect=Formula


[Probe Connect FORMULA PARAMETERS]

#1=sensor_type

#2=sensor_connection


[Probe Connect FORMULA]

if ((#2==5) || (#2==2) && (#1>5) && (#1<12))

{

        Probe Connect =Differential

}

else
```

```
{
        Probe Connect =Single probe
}
```

2.7.4 Example Formula #2

The DDL file for the 3244MVRev1 contains the following:

```
#define DIFF ((sensor_1_type != 250 && sensor_2_type != 250) && (primary_variable_code == 0 || secondary_variable_code == 0 ||
    tertiary_variable_code == 0 || fourth_variable_code == 0))
```

The DIFF refers to Differential. This equation may be used to detemine the value of the “Probe Connect” info type since for this device there is not corresponding parameter. The common values which probe connect can assume are “Differential” and “Single probe”. With that knowledge, we can convert to infix and construct the following MAP file entries:

```
[PARAMETER SPECIAL INFO]
Probe Connect=Formula


[Probe Connect FORMULA PARAMETERS]
#1=_SNSR1_
#2=_SNSR2_
#3=primary_variable_code
#4=secondary_variable_code
#5=tertiary_variable_code
#6=fourth_variable_code


[Probe Connect FORMULA]
if ((#1 !=250 && #2 !=250) && (#3 ==0 || #4 ==0 || #5 ==0 || #6 ==0))
{
        Probe Connect = Differential
}
else
{
        Probe Connect = Single probe
}
```

2.7.5 Example Formula #3

The third example is the Brooks Tri-20. In this device there is no parameter for Number of Wires but there is a float value which determines whether the device is 3 wires or 4 wires. The name of the float value that determines this is “RTDOffset”. If “RTDOffset” is equal to 0 then the device is 4 wires otherwise the device is 3 wires. With that knowledge, we can convert to infix and construct the following MAP file entries:

```
[PARAMETER SPECIAL INFO]

Number of Wires=Formula


[Number of Wires FORMULA PARAMETERS]

#1=RTDOffset


[Number of Wires FORMULA]

if (#1 ==0)

{

        Number of Wires =4 wires

}

else

{

        Number of Wires =3 wires

}
```

2.8 [input(or output) block Units Code DD to Common Map]

The output and input units are set up in the same format. All of the information for this section can be found in the **Tables section 4.1** of this document. In addition to the unit list specifically supported by the device type, the following entries must also be included and are already listed in the template: “250=Not used”, “251=None”, “252=Unknown” and “253=Special”.

Format:

<ddKey>=<cValue>

An example of this section is listed below.

```
[input block Units Code DD to Common Map]

1=InH2O

2=InHg

3=FtH2O

4=mmH2O

5=mmHg
```

```
6=psi

7=bar

8=mbar

9=g_SqCm

10=kg_SqCm

11=PA

12=kPA

13=torr

14=ATM

250=Not used

251=None

252=Unknown

253=Special
```

2.9 [Probe Type DD to Common Map]

This section needs to exist on the same conditions as listed in the previous section. The difference is that in this section the keys and values are reversed. The keys are the common format and the values are the DD Codes. This mapping should also include the following items “Not used=250”, “None=251”, “Unknown=252” and “Special=253”.

Format:

<ddKey>=<cValue>???

The following is an example.

```
Probe Type DD to Common Map]

0=Ohms

1=RTD Pt  100    a: 0.003850

2=RTD Pt  100    a: 0.003920

3=RTD Pt  200    a: 0.003850

4=RTD Pt  500    a: 0.003850

5=RTD Ni  120    a: 0.006180

6=Ohms

7=RTD Pt  100    a: 0.003850

8=RTD Pt  100    a: 0.003920

9=RTD Pt  200    a: 0.003850

10=RTD Pt  500    a: 0.003850

11=RTD Ni  120    a: 0.006720
```

```
128=milliVolts

139=TC Type S (Pt10Rh-Pt) (IEC 584 etc.)

140=TC Type U (Cu-CuNi) (DIN 43710)

250=Not used

251=None

252=Unknown

253=Special
```

2.10 [Number of Wires DD to Common Map]

This section needs to exist on the same conditions as listed in the previous section. The keys are the common format and the values are the DD Codes. This mapping should also include the following items "Not used=250", "None=251", "Unknown=252" and "Special=253".

Format:

<ddKey>=<cValue>

The following is an example.

```
Number of Wires DD to Common Map]

2=2 wires

3=3 wires

4=4 wires

5=2 wires

250=Not used

251=None

252=Unknown

253=Special
```

2.11 [Pressure Measurement Type DD to Common Map]

This section needs to exist on the same conditions as listed in the previous section. The difference is that in this section the keys and values are reversed. The keys are the common format and the values are the DD Codes. This mapping should also include the following items "Not used=250", "None=251", "Unknown=252" and "Special=253".

Format:

<ddKey>=<cValue>???

The following is an example.

```
Pressure Measurement Type DD to Common Map]

0=Differential

1=Gage

2=Absolute

3=High line

4=Liquid level

5=Draft range

250=Not used

251=None

252=Unknown

253=Special
```

2.12 [Transfer Function Code DD to Common DD Map]

This section needs to exist on the same conditions as listed in the previous section. The keys are the common format and the values are the DD Codes. This mapping should also include the following items “Not used=250”, “None=251”, “Unknown=252” and “Special=253”.

Format:

<ddKey>=<cValue>???

The following is an example.

```
[Transfer Function Code DD to Common Map]

0=Linear

1=Square root

2=Square root third power

3=Square root fifth power

4=Table relation

5=Square

230=Switch

231=Square root special function

232=Square root third power special function

233=Square root fifth power special function

250=Not used

251=None

252=Unknown
```

```
253=Special
```

2.13 [Wave Form Common to DD Map]

This section needs to exist on the same conditions as listed in the previous section. The difference is that in this section the keys and values are reversed. The keys are the common format and the values are the DD Codes. This mapping should also include the following items “Not used=250”, “None=251”, “Unknown=252” and “Special=253”.

Format:

<cKey>=<ddValue>???

The following is an example.

```
[Wave Form DD to Common Map]

Square=0

Sine=1

Triangle=2

Sawtooth=3

Not used=250

None=251

Unknown=252

Special=253
```

2.14 [Probe Connect DD to Common Map]

This section needs to exist on the same conditions as listed in the previous section. The difference is that in this section the keys and values are reversed. The keys are the common format and the values are the DD Codes. This mapping should also include the following items “Not used=250”, “None=251”, “Unknown=252” and “Special=253”.

Format:

<ddKey>=<cValue>???

The following is an example.

```
[Probe Connect DD to Common Map]
```

3 GENERAL INFORMATION FOR MAP FILE TEMPLATE

3.1 Block Types

In the next section that deals with input block type there will be a listing of the additional parameters that must be completed for that specific input block type. To be complete, the block has to have one of the following: "No Associated Parameter" with a default value, "ItemArray" or "Parameter" in the **[PARAMETER MAP]** section with the corresponding map that is required or a "Formula" with all the required information explained in detail in the **Formulas** section 2.6. The mapping sections are explained in section 2.7-2.22. Any key in the [PARAMETER MAP] that doesn't adhere to these requirement in order to be complete will have specific information listed.

3.1.1 Pressure Input Block Type

For the pressure input block type the only additional parameter that must be completed is "Pressure Measurement Type".

3.1.2 Temperature Thermocouple Input Block Type

For the temperature thermocouple input block type the additional parameters needed are "Probe Type", "Cold Junction Compensation" and "Probe Connect. For "Cold Junction Compensation a "No Associated Parameter" will go into [PARAMETER SPECIAL INFO] with one of the following default values: 1=Internal(supplied by calibrator), 2=External(probe attached to calibrator), 3=Fixed Manual or 4=None in the [DEFAULT PARAMETER VALUES]. These default values can be referenced in **Table section 4.5**.

3.1.3 Temperature RTD Input Block Type

For the temperature rtd input block type the additional parameters needed are "Probe Type" and "Number of Wires".

3.1.4 Frequency Input Block Type

For the frequency input block type the additional parameters need are "Amplitude" and "Wave Form".

3.1.5 Device Parameters

The following keys in the **[PARAMETER MAP]** are used to identify a device over and above the AMS Tag.

- Date, Descriptor, Message, Model Name, HART Universal Revision (only for HART devices), Tag and Polling Address all already have parameters in the **[PARAMETER MAP]**. If any of these are incorrect then please put the correct corresponding parameter in the map file.
- If a parameter for Manufacturer Name doesn't exist in the device then list it as a "No Associated Parameter" in the **[PARAMETER SPECIAL INFO]** and then set the default value to the manufacturer number in a decimal value. If the device's HART Universal Revision number is less than 5 the "Unique ID" will need to be left blank and the "Serial Number" should be the "final_assembly_number". If the HART Universal Revision number is 5 or greater then in the **[PARAMETER MAP]** section of the map file the keys, "Serial Number" and "Unique ID", must have "unique_id" entered for those values.

3.1.6 General Input Block Information

The following keys are defined as float values: "Damping Value", "input block Lower Range Value", "input block Upper Range Value", "Lower Sensor Limit" and "Upper Sensor Limit". This information is needed because if a "No Associated Parameter" is used for any of these keys then a float value must be entered in the **[DEFAULT PARAMETER VALUES]** section of the map file.

3.1.7 General Output Block Information

The following keys are defined as float values: "output block Lower Range Value" and "output block Upper Range Value". This information is needed because if a "No Associated Parameter" is used for any of these keys in [**PARAMETER SPECIAL INFO]** section then a float value must be entered in the **[DEFAULT PARAMETER VALUES]** section of the map file.

3.2 Verification of Map Files

After the map files have been created they need to be verified. There is a MapFileTester application to allow testing of map files. UTP-659 and TP-659 describe the tests, and there is a MapFileTesterManual.doc which explains how to install and run the application and snap-on.

3.3 Troubleshooting Tips

It is absolutely essential that the correct parameter or item array's are used in the parameter map. In the cases where no parameter, formula or item array exists for a key and the device supports two or more possible choices for that key then pick the most commonly used option. An example of this is one of the device we currently support has a Temperature RTD probe that supports 3 and 4 wires. The code or the DD for this device doesn't know how many wires the RTD supports, the only difference is how the customer physically wires the device. In this case, the map file has "No Associated Parameter" for the Number of Wires in the **[PARAMETER SPECIAL INFO]** section and in the **[DEFAULT PARAMETER VALUES]** section is 3 wires. This is an example of how to handle non ideal cases. This is how it looks in the map file.

```
[PARAMETER SPECIAL INFO]

Number of Wires=No Associated Parameter

[DEFAULT PARAMETER VALUES]

Number of Wires=3 wires
```

Another situation that occurred was that "Transfer Function Code" was set to a dummy value. In this case, a very strange value was coming back from the AMS server. The map file had to be adjusted for this. This adjustment was made by setting Transfer Function Code to "No Associated Parameter" in the **[PARAMETER SPECIAL INFO]** section and in the **[DEFAULT PARAMETER VALUES]** section to Linear.

```
[PARAMETER SPECIAL INFO]

Transfer Function Code=No Associated Parameter

[DEFAULT PARAMETER VALUES]

Transfer Function Code=0
```

In these special cases, there is no set of directions on how to handle the situations but the map file format must be followed. AMS will not function correctly if the map files aren't correct.

When the test definition is done if a dialog box appears asking for additional information try to provide the information requested, if possible. Even if the information entered at this point is correct, investigate and determine whether anymore information is missing. If anywhere in the testing an error is encountered some piece of information is incorrect and must be investigated more to find out the reason why the error occurred. No errors can be received in order for a map file to be tested and be considered complete.

4 TABLES

The following tables list the supported enumerated values.
The tables are in the form of “DD Code=Common Format”.

4.1 Units Table

Unit String	What that string represents
"InH2O"	Pressure – inches of water (at 68 deg F)
"InHg"	Pressure – inches of mercury
"FtH2O"	Pressure – feet of water
"mmH2O"	Pressure – millimeters of water
"mmHg"	Pressure – millimeters of Mercury
"psi"	Pressure – pounds per square inch
"bar"	Pressure -bar
"mbar"	Pressure – milli-bar
"g_SqCm"	Pressure – grams per square centimeter
"kg_SqCm"	Pressure – kilograms per square centimeter
"PA"	Pressure - Pascal
"kPA"	Pressure – kilo-pascal
"torr"	Pressure - torr
"ATM"	Pressure - atmosphere
"CuFt_min"	Flow – cubic foot per minute
"gal_min"	Flow – gallon per minute
"l_min"	Flow – Liter per minute
"ImpGal_min"	Flow – Imperial gallon per minute
"CuMtr_hr"	Flow – Cubic meter per hour
"ft_s"	Generic – Feet per second

Unit String	What that string represents
"mtr_s"	Generic – Meter per second
"gal_s"	Flow – Gallon per second
"MillionGal_day"	Flow – million gallon per day
"l_s"	Flow – Liter per second
"MilL_day"	Flow – milli-liter per day
"CuFt_s"	Flow – Cubic Foot per second
"CuFt_day"	Flow – Cubic foot per day
"CuMtr_s"	Flow – Cubic meter per second
"CuMtr_day"	Flow – cubic meter per day
"ImpGal_hr"	Flow – Imperial gallon per hour
"ImpGal_day"	Flow – Imperial gallon per day
"degC"	Temperature – Degrees Centigrade
"degF"	Temperature – Degrees Fahrenheit
"degR"	Temperature – Degrees Rankin
"Kelvin"	Temperature – Degrees Kelvin
"mV"	Generic – milli-volts
"Ohm"	Generic - Ohms
"Hz"	Generic - Hertz
"mA"	Generic – milli-Amps
"gal"	Volume - gallon
"liter"	Volume - liter
"ImpGal"	Volume – Imperial gallon
"CuMtr"	Volume – cubic meter
"ft"	Generic - feet
"meter"	Generic - meter
"bbl"	Volume - Barrel
"in"	Generic – inches
"cm"	Generic – Centi-meter

Unit String	What that string represents
"mm"	Generic – milli-meter
"min"	Generic – minute
"sec"	Generic – second
"hr"	Generic – Hour
"day"	Generic – day
"centi_stokes"	
"cpoise"	
"uMho"	
"Percent"	Generic – percent
"v"	Generic – volts
"pH"	Generic – acidity
"gram"	Generic – gram
"kg"	Generic – kilo-gram
"MetTon"	Mass – Metric tonne
"lb"	Mass – pounds
"ShTon"	Mass – Short Ton
"LTon"	Mass – Long Ton
"mSiemen_cm"	
"uSiemen_cm"	
"newton"	Generic – newtons
"newton_meter"	Generic – Newton meter
"g_s"	Flow – Grams per second
"g_min"	Flow – Grams per minute
"g_hr"	Flow – Grams per hour
"kg_s"	Flow – Kilo-grams per second
"kg_min"	Flow – Kilo-grams per minute
"kg_hr"	Flow – Kilo-grams per hour
"kg_day"	Flow – Kilo-grams per day

Unit String	What that string represents
"MetTon_min"	Flow – Metric tonne per minute
"MetTon_hr"	Flow – Metric tonne per hour
"MetTon_day"	Flow – Metric tonne per day
"lb_s"	Flow – pounds per second
"lb_min"	Flow – pounts per minute
"lb_hr"	Flow – pounds per hour
"lb_day"	Flow – pounds per day
"ShTon_min"	Flow – short ton per minute
"ShTon_hr"	Flow – short ton per hour
"ShTon_day"	Flow – short ton per day
"LTon_hr"	Flow – long ton per hour
"LTon_day"	Flow – long ton per day
"deka_therm"	
"SGU"	Density – specific gravity unit
"g_CuCm"	Density – gram per cubic centimeter
"kg_CuMtr"	Density – kilo gram per cubic meter
"lb_gal"	Density – pound per gallon
"lb_CuFt"	Density – pound per cubic foot
"g_ml"	Density – grams per milli liter
"kg_l"	Density – kilo gram per liter
"g_l"	Density – gram per liter
"lb_CuIn"	Density – pounds per cubic inch
"ShTon_CuYd"	Density – short ton per cubic yard
"degTwad"	
"degBrix"	
"degBaum_hv"	
"degBaum_lt"	
"degAPI"	

Unit String	What that string represents
"Percent_sol_wt"	
"Percent_sol_vol"	
"degBall"	
"proof_vol"	
"proof_mass"	
"bush"	Volume - bushel
"CuYd"	Volume – Cubic Yard
"CuFt"	Volume – cubic foot
"CuIn"	Volume – cubic inch
"in_s"	
"in_min"	
"ft_min"	
"deg_s"	
"rev_s"	
"rpm"	
"mtr_hr"	
"normal_cubic_meter_per_hour"	Flow – cubic meter per hour
"normal_liter_per_hour"	Flow – normal
"standard_cubic_feet_per_minute"	Flow – standard usually means at 68 degree F
"bbl_liq"	Volume – barrels of liquid
"ounce"	Mass - ounce
"foot_pound_force"	
"kilo_watt"	
"kilo_watt_hour"	
"horsepower"	
"CuFt_hr"	
"CuMtr_min"	
"bbl_s"	

Unit String	What that string represents
"bbl_min"	
"bbl_hr"	
"bbl_day"	
"gal_hr"	
"ImpGal_s"	
"l_hr"	
"parts_per_million"	
"mega_calorie_per_hour"	
"mega_joule_per_hour"	
"british_thermal_unit_per_hour"	
"degrees"	
"radians"	
"percent_StmQual"	
"Ftin16"	
"CuFt_lb"	
"pico_farads"	
"Percent_plato"	
"mega_calorie"	
"kOhm"	Generic – kilo-ohms
"mega_joule"	
"british_thermal_unit"	
"normal_cubic_meter"	Volume – Normal cubic meter at 0 deg C
"normal_liter"	Volume –
"standard_cubic_feet"	Volume – cubic feet at 68 deg F
"gallons_per_day"	Flow – gallons per day
"hecto_liter"	Volume –
"mega_pascals"	
"in_H2O_4_degrees_C"	

Unit String	What that string represents
"mm_H2O_4_degrees_C"	
"Not used"	
"None"	
"Unknown"	
"Special"	
“hecto_pascals"	
"kilo_herz"	
"cpm"	Generic - Counts per minute
"cph"	Generic – Counts per hour
"mmH2O_60_degree_F"	
"kilo_ponds"	
"cm_H2O_4_degC"	
"cm_H2O_20_degC"	
"cm_H2O_60_degF"	
“in_H2O_60_degF”	
“cu_centimeter”	Volume - Cubic Centimeters
“dl_liter”	Volume - Dilliliters
“pulses”	Generic – pulses or counts
“nanometer”	Nanometer
“ml_liter”	Volume - Milli-liters
“cu_meter_per_second”	Cubic Meters per second == m^3/sec
"minutes_of_degree"	
"seconds_of_degree"	
"gon"	
"revolutions"	
"kilometer"	
"micro_meter"	
"pico_meter"	

Unit String	What that string represents
"angstrom"	
"yard"	
"mile"	
"nautical_mile"	
"square_meter"	
"square_kilometer"	
"square_centimeter"	
"square_decimeter"	
"square_millimeter"	
"acre"	
"hectare"	
"square_inch"	
"square_foot"	
"square_yard"	
"square_mile"	
"cubic_decimeter"	
"cubic_millimeter"	
"centiliter"	
"cubic_mile"	
"pint"	
"quart"	
"kilosecond"	
"millisecond"	
"microsecond"	
"millimeter_per_second"	
"kilometer_per_hour"	
"knot"	
"yard_per_second"	

Unit String	What that string represents
"yard_per_minute"	
"inch_per_hour"	
"foot_per_hour"	
"yard_per_hour"	
"mile_per_hour"	
"meter_per_second_square"	
"terraherz"	
"gigaherz"	
"megaherz"	
"per_second"	
"per_minute"	
"radians_per_second"	
"per_second_square"	
"milligram"	
"megagram"	
"megagram_per_cubic_meter"	
"kilogram_per_cubic_decimeter"	
"gram_per_cubic_meter"	
"metric_ton_per_cubic_meter"	
"kilogram_per_meter"	
"milligram_per_meter"	
"tex"	
"kilogram_square_meter"	
"kilogram_meter_per_second"	
"meganewton"	
"kilonewton"	
"millinewton"	
"micronewton"	

Unit String	What that string represents
"kilogram_square_meter_per_second"	
"newton_meter"	
"meganewton_meter"	
"kilonewton_meter"	
"millinewton_meter"	
"gigapascal"	
"millipascal"	
"micropascal"	
"feet_of_water_4degc"	
"liters_per_day"	
"ampere"	
"kiloampere"	
"microampere"	
"nanoampere"	
"picoampere"	
"megavolt"	
"kilovolt"	
"microvolt"	
"farad"	
"millifarad"	
"microfarad"	
"nanofarad"	

4.2 Transfer Function Code Table

Relationship	usage
“Linear”	Nearly all devices are of this relationship
“Square root”	Some HART/Conventional pressure transmitters are set up with this relationship to easily measure

Relationship	usage
	flow using differential pressure.
“Square root third power”	
“Square root fifth power”	
“Table relation”	
“Square”	
“Switch”	Our Conventional switches are the only ones to use this relationship
“Square root special function”	
“Square root third power special function”	
“Square root fifth power special function”	
“Not used”	
“None”	
“Unknown”	

4.3 Temperature Probe Type Table

Probe type	Description
`"Ohms"`	
`"killoOhms"`	
`"Calibrated: will supply Calendar-VanDeusen parameters"`	
`"RTD Pt   50    a: 0.003850  (IEC751)"`	
`"RTD Pt  100    a: 0.003850"`	
`"RTD Pt  200    a: 0.003850"`	
`"RTD Pt  500    a: 0.003850"`	
`"RTD Pt 1000    a: 0.003850"`	
`"RTD Pt  50     a: 0.003916  (JIS C1604-81)"`	
`"RTD Pt  100    a: 0.003916"`	
`"RTD Pt   50    a: 0.003920  (MIL-T-24388)"`	

Probe type	Description
"RTD Pt 100 a: 0.003920"	
"RTD Pt 200 a: 0.003920"	
"RTD Pt 500 a: 0.003920"	
"RTD Pt 1000 a: 0.003920"	
"RTD Pt 10 a: 0.003923 (SAMA RC21-4-1966)"	
"RTD Pt 100 a: 0.003923 "	
"RTD Pt 200 a: 0.003923"	
"RTD Pt 100 a: 0.003926 (IPTS-68)"	
"RTD Ni 50 a: 0.006720 (Edison curve #7)"	
"RTD Ni 100 a: 0.006720"	
"RTD Ni 120 a: 0.006720"	
"RTD Ni 1000 a: 0.006720\\n"	
"RTD Ni 50 a: 0.006180 (DIN 43760)"	
"RTD Ni 100 a: 0.006180"	
"RTD Ni 120 a: 0.006180"	
"RTD Ni 1000 a: 0.006180"	
"RTD Cu 10 a: 0.004270 (?\? Standard ?\?)"	
"RTD Cu 100 a: 0.004270"	
"microVolts"	
"milliVolts"	
"Volts"	
"TC Type B (Pt30Rh-Pt6Rh) (IEC 584 etc.)"	
"TC Type W5, Omega type C (W5-W26Rh)(ASTM E 988)"	
"TC Type W3, Omega type D (W3-W25Rh)(ASTM E 988)"	
"TC Type E (Ni10Cr-Cu45Ni) (IEC 584 etc.)"	
"TC Type W, Omega type G (W-W26Rh)(ASTM E 988)"	
"TC Type J (Fe-Cu45Ni) (IEC 584 etc.)"	
"TC Type K (Ni10Cr-Ni5) (IEC 584 etc.)"	

Probe type	Description
"TC Type N (Ni14CrSi-NiSi) (IEC 584 etc.)"	
"TC Type R (Pt13Rh-Pt) (IEC 584 etc.)"	
"TC Type S (Pt10Rh-Pt) (IEC 584 etc.)"	
"TC Type T (Cu-Cu45Ni) (IEC 584 etc.)"	
"TC Type L (Fe-CuNi) (DIN 43710)"	
"TC Type U (Cu-CuNi) (DIN 43710)"	
"TC Pt20Rh-Pt40Rh (ASTM E 1751)"	
"TC Ir-Ir40Rh (ASTM E 1751)"	
"TC Platinel II"	
"TC Ni-NiMo"	
"Bi-metallic strip"	
"Vapor pressure bulb"	
"Liquid expansion"	
"Mercury bulb"	
"Not used"	
"None"	
"Unknown"	
"Special"	
"TC Radiamatic"	
"RTD Pt 100 a: 0.003902"	
"RTD Ni500 a: 0.005198"	
"RTD Cu 25 a: 0.004267"	
"RTD Cu 50 a: 0.004280"	
"RTD Cu 100 a: 0.004280"	
"RTD Cu10 GE a: 0.004267"	

4.4 Probe Connect Table

This field is only used for temperature block types RTD

Probe connect type	Comments
Single probe	
Differential	
Not used	
None	
Unknown	
Special	

4.5 Cold Junction Compensation Table

Only used for thermo-couple blocks.

Cold Junction compensation	Comments
Internal	
External	
Manual	
None	

4.6 Pressure Measurement Type Table

Only used for pressure block types (either input – for transmitters, or output – for I/P and Valve devices)

Pressure measurement type	Comments
Differential	
Gage	Absolute pressure – atmospheric pressure
Absolute	
High line	High pressure >1000 psi, usually requires applying pressure with liquid not gas
Liquid level	
Draft range	
HTG	
Not used	
None	

Pressure measurement type	Comments
Unknown	
Special	

4.7 Frequency Wave Form Table

This is only used for frequency blocks. These are often used for flow devices such as the 8800 vortex.

Frequency Wave form Table	Comments
Square	
Sine	
Triangle	
Sawtooth	
Uni-Polar Square Wave	
Not used	
None	
Unknown	
Special	

4.8 Number of Wires Table

Number of wires	Comments
2 wires	
3 wires	
4 wires	
Not used	
None	
Unknown	
Special	