

# NaviPac

## A1. User defined Outputs



---

## Version History

<b>Version</b>	<b>Who</b>	<b>Additions</b>
1.0	kup:	23. nov. 1998 Created
1.01	kup:	26. nov. 1998 Updated
1.1	kup:	07. Jan. 1999 Updated due to implementation phase
1.2	kup:	29. Jan. 1999 Updated from design to user guide
1.2a	OKR:	01. Feb 1999 Reviewed
1.3	OKR	13. Apr 2000 Updated to NaviPac 3.2
1.3a	OKR	25. Apr 2000 Allow output of KP in meter
1.3b	OKR	25. May 2000 Expanded event output to include event texts
1.4	OKR	25.Oct. 2002 Added new time past midnight time format
3.4	OKR	22. Jan. 2004 Added combined data acq. Item
3.4Cp7	OKR	25. May 2004 Added Runline heading
3.4Cp9	OKR	01. Sep. 2004 Expanded Lat/Long output
3.5p9	OKR	12.11.2007 Upgraded to current version
3.5p12	OKR	31.3.2008 Add possible output of DCC
3.5p13	OKR	27.5.2008 Added range/bearing to Waypoint
3.5p16	OKR	17.11.2008 Allow output of DOL in cm and feet
3.8	OKR	18.04.2012 Version 3.8

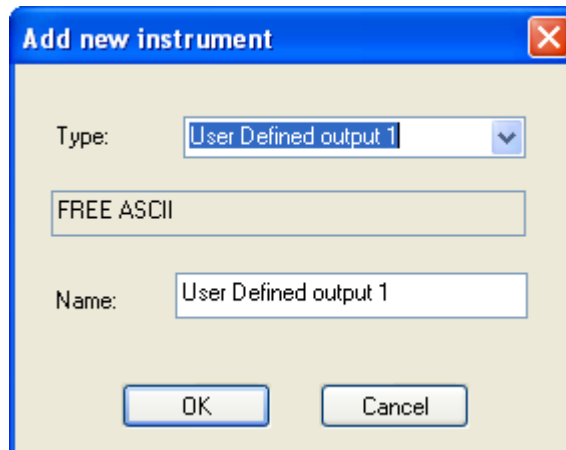
## Table of contents

<b>1.</b>	<b>DESCRIPTION OF GUI.....</b>	<b>4</b>
1.1	MAIN DIALOGUE.....	5
1.2	ITEM FORMAT (MORE BUTTON).....	7
1.2.1	Free text item.....	8
1.2.2	Position item.....	8
1.2.2.1	Special Lat/Long output.....	9
1.2.3	KP Item.....	10
1.2.3.1	Forced to meter.....	10
1.2.3.2	US Stations.....	11
1.2.4	DOL or DCC.....	11
1.2.4.1	Forced to CM.....	11
1.2.4.2	Forced to Survey Feet.....	11
1.2.5	Speed.....	12
1.2.6	Range/Bearing to Waypoint.....	12
1.2.7	Gyro item.....	13
1.2.8	Data acquisition data item.....	13
1.2.9	Motion item.....	13
1.2.10	Event item.....	14
1.2.11	Run line direction.....	14
1.2.12	Date & Time item.....	14
1.2.13	Raw instrument output.....	15
1.2.14	Geodesy information.....	15
1.2.15	Speed log data.....	16
1.2.16	Combined data acquisition.....	17
1.3	DEFINITION FILE.....	18
<b>2.</b>	<b>CUSTOM OUTPUT FORMAT STRINGS.....</b>	<b>18</b>
	FORMAT SPECIFICATION FIELDS: PRINTF AND WPRINTF FUNCTIONS.....	18
2.1	PRINTF TYPE FIELD CHARACTERS.....	19
2.2	FLAG DIRECTIVES.....	20
2.3	PRINTF WIDTH SPECIFICATION.....	20
2.4	HOW PRECISION VALUES AFFECT TYPE.....	21
2.5	SIZE PREFIXES FOR PRINTF AND WPRINTF FORMAT-TYPE SPECIFIERS.....	22
2.6	DATE & TIME FORMAT.....	22

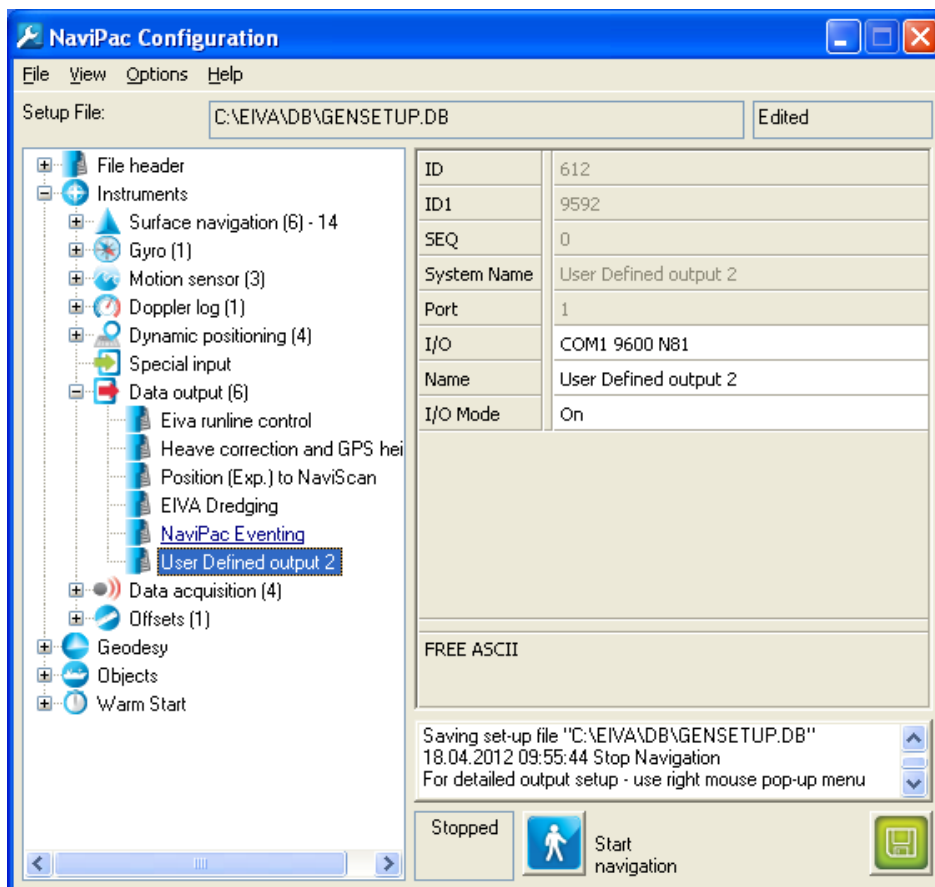
## 1. Description of GUI

User defined outputs is a utility to specify up to 10 custom defined outputs in NaviPac. It can be done from the NaviPac Configuration module NPConfig

To define a new User defined output select the Data Output tree node and activate right mouse: *Add New Item*



Select one of the 10 predefined outputs. These will be named *User\_defined\_output\_1* to *User\_defined\_output\_10*.



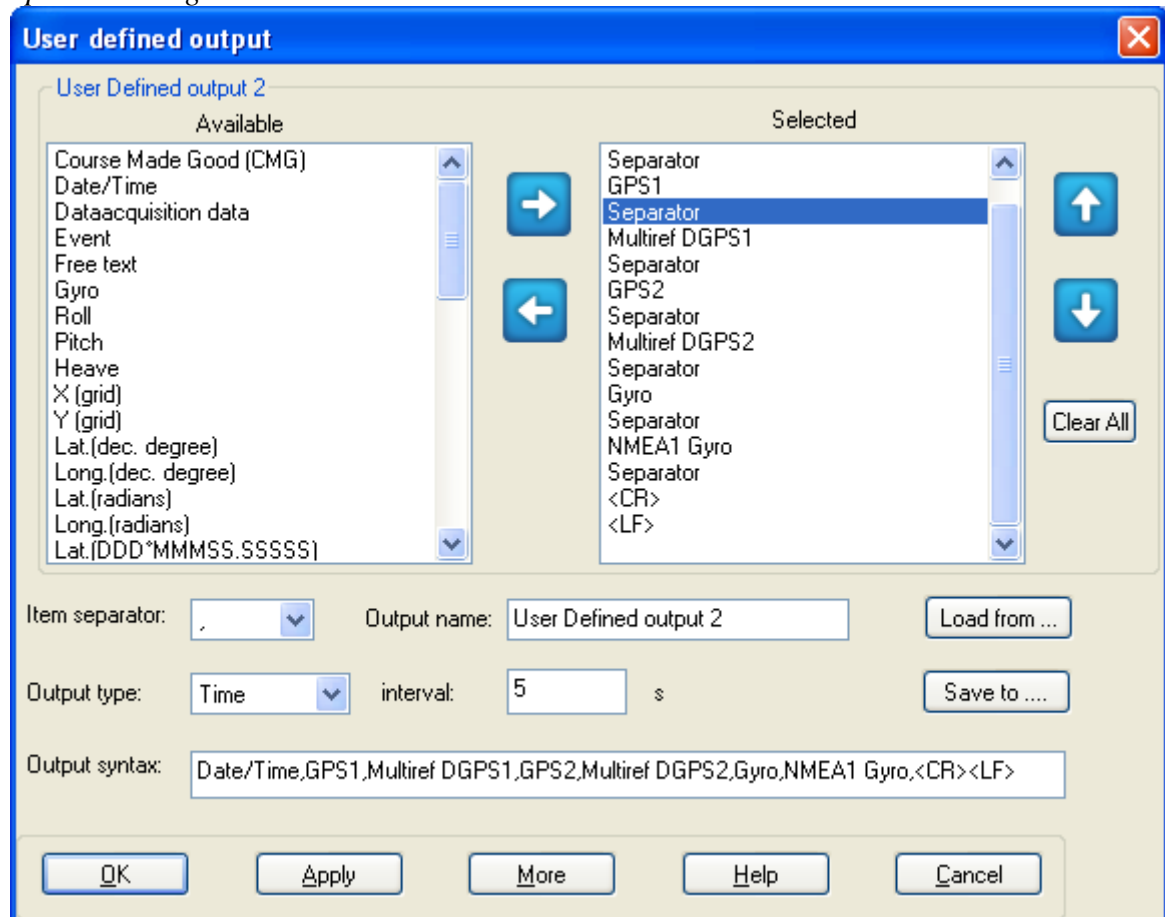
You can specify for each output specify:

- Mode: On, Off

- Interfacing details
- Naming

### 1.1 Main dialogue

To edit the user defined output select the instrument, activate right mouse and select *Edit Special Settings*:



The dialogue contains 2 lists with items that can be output on the selected port. The following items can be selected:

- Free text items – ASCII or binary
- Item separators: **comma, space, colon, and semicolon**
- Terminators: **CR, LF, STX, and ETX**
- Date/Time
- Position items:  
X, Y, KP, DAL, DOL, LAT, LONG, AGE, HEIGHT, Standard deviation
- Gyro
- Course Made Good (CMG)
- Speed Made Good (SMG)
- Motion sensor data (Roll, Pitch, Heave)
- Speed log data <esp. RDI PD0 information>
- Data-acquisition data (e.g. depths from echo sounders)
- Event.
- Raw data – copy of incoming data string
- Geodesy information

- 
- Run line control information (name, logging on/off) and direction
  - RTK based heave bias
  - Check-sum (2 character exclusive or)
  - Combined data acquisition – scaling and arithmetic
  - Run line direction (true or grid)
  - Range/bearing to waypoint

Note: There is no limit of the number of times an item is included.

**Available List:**

This list contains all output items available. To add an item to the *Selected* list, it must be selected in this list. The *Include* Button (Right arrow) is used to place an item in the *Selected* list. To move an item from the *Selected* list to the *available* list, select an item by clicking in the *Selected* list and press the *Exclude* Button (Left arrow).

**Selected List:**

This list contains all output items selected to be output to a port. To remove an item from the output string, it must be selected in this list and then exclude button must be pressed.

**Include Button:**

A selected output item (in *available* list) is moved to *Selected* list, which indicates that the output item is to be a part of the output.

**Exclude Button:**

Remove a selected output item from the *Selected* list to the *Available* List – do not output it.

**Clear All:**

Erase the entire included list.

Below the 2 lists the following can be set-up:

**Item Separator:**

How to separate field (output items in selected list): **comma, space, colon, and semicolon**. These will be “global” for all items.

**Output type:**

Specify how often the output is generated: **All updates, Time, Event**

**Interval:**

If Time is selected in ‘Output type’ the frequency to output the output string can be specified.

**Output syntax:**

This field will show how the resulting string to output will look like. This can also be displayed from main menu function: *View, Display NaviPac setup*

***Example:***

If the user selected and defined the following:

1. Free text item (time)
2. Separator (,)
3. Date/time item
4. Separator (,)
5. Free text item (position(X,Y) )
6. Separator (,)
7. Position (x)
8. Separator (,)
9. Position (Y)
10. Separator (,)
11. text item (depth)
12. Separator (,)
13. Data acquisition data (echo sounder channel)
14. CR
15. LF

The resulting output string could be:

Time, Date/Time, Position (X, Y), X (grid), Y (grid), NaviSound 2000:1<CR><LF>

OK button:

Accept last changes in Selected list and save changes.

More button:

If an item is selected in 'Selected list' this button is available. The dialogue that pop up depends on the item selected – see below.

Cancel button:

Close the dialogue.

Save to

Save a copy of the current defined output in external file. This file can then be used at later stage.

Load From:

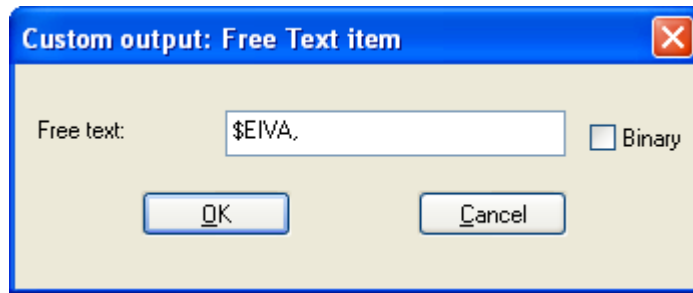
Load setting from external file – current selection will be erased.

### **1.2 Item format (More button)**

If wanted, it is possible to specify the format that a position, a gyro a motion value (RPH) or depth should have when written to the output string (and port).

The fields presented will depend on the item type selected. Below some of these will be described:

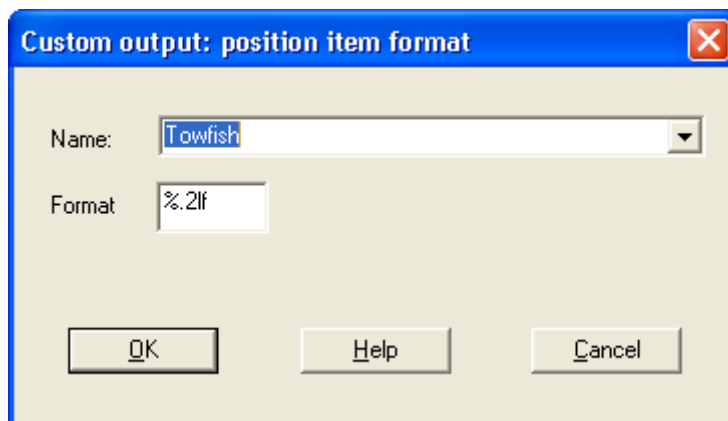
### 1.2.1 Free text item



A dialogue with an ASCII text field is shown. To output binary data, select Binary and enter the ASCII codes separated by comma.

### 1.2.2 Position item

Positions can have a fixed or a user defined output format. If fixed is selected a default format is used. If not, the user is allowed to enter a format string. This format The format field for positions, depths etc follow the ANSI C syntax – see 2.

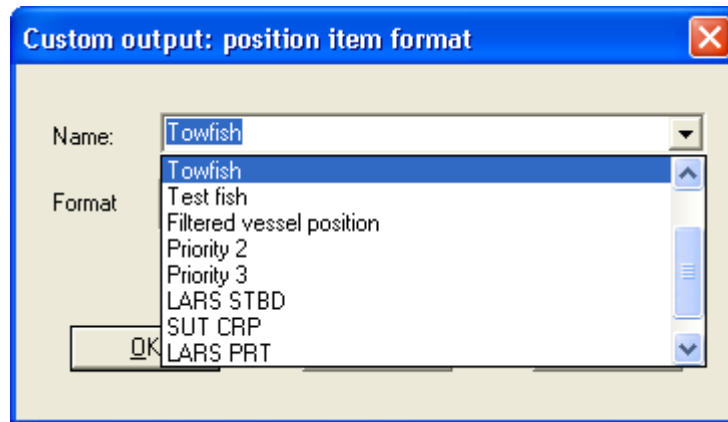


The dialogue contains a list with position items that can be output on the selected port.

The dialog will appear for the following items:

- HEIGHT,
- X,
- Y,
- KP,
- DAL,
- DOL,
- LAT,
- LONG,
- CMG
- SMG
- Standard deviation
- Range to waypoint
- Bearing to waypoint

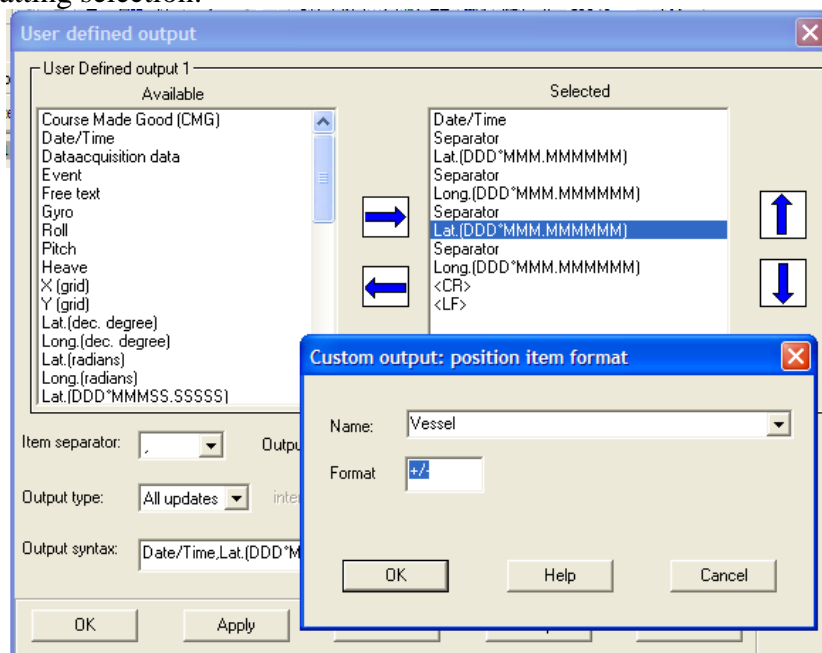
For positions the position “source” has to be selected. This is done pressing the ‘Name’ button. This will show a list that look like:



Here a position ‘source’ for the output can be specified. The dialogue is re-used from other functions ‘objects selection’.

### 1.2.2.1 Special Lat/Long output

The Lat/Long output as degree and decimal minutes (DDD°MMM.MMMMMM’) offers a special formatting selection:



The output can be in three different formats

- **Signed (Default):**  
The output will be prefix with a minus if it’s west or south.  
**Sample:** -000°01.146058'  
**Definition:** Enter +/- in the format field
- **Append indicator:**  
An E/N/S/W will be appended to the string.

**Sample:** 000°01.146058'W

**Definition:** Enter E or N or S or W in the format field

- Separator appended indicator:

A separator plus E/N/S/W will be appended to the string.

**Sample:** 000°01.146058',W

**Definition:** Enter the separator (NOT E or N or S or W) in the format field – eg. ‘,’ in the above sample.

- Special case

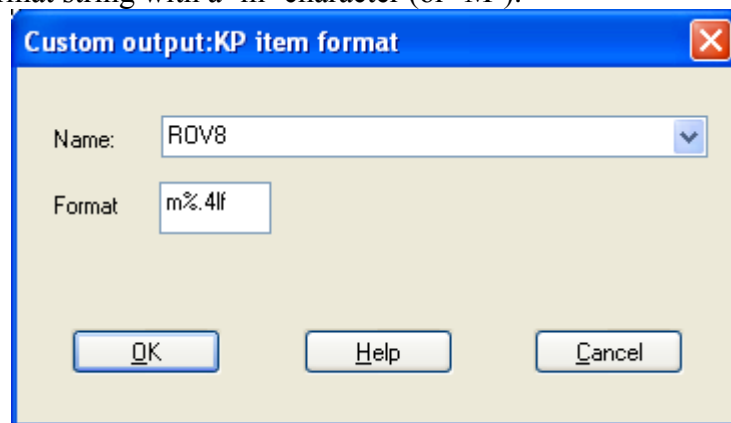
Data without degree signs with 4 decimals.

**Sample:** Lat <DD MMM.MMMM> N Long <DDD MMM.MMMM> E

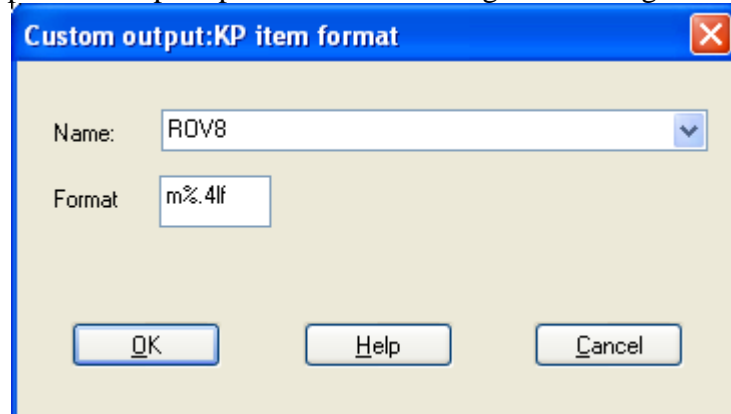
**Definition:** Enter X

### 1.2.3 KP Item

The output of KP (runline information) is similar to position output. But very often people would like to change the format unit (default is kilometre) to meter. This can be done by prefixing the format string with a ‘m’ character (or ‘M’):



The above example will output kp in meters with 5 digits including leading zero.



The ordinary kp settings outputs in km with 4 or 5 ciphers.

#### 1.2.3.1 Forced to meter

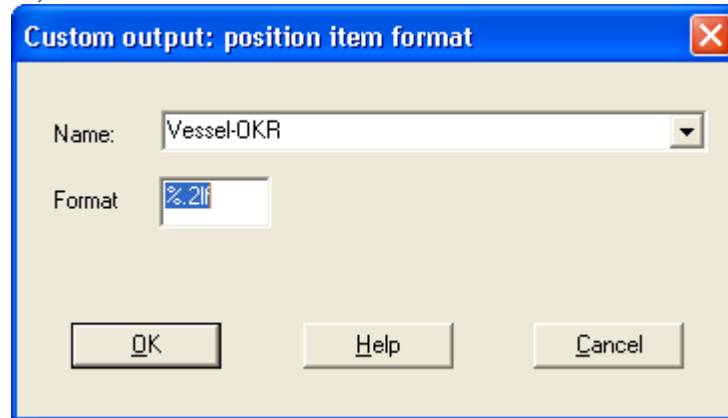
You may force the output to meter by an ‘m’ in front of the format string

### 1.2.3.2 US Stations

If used in the US it's often requested to get the value as Station instead. This can be achieved by entering 'S' or 's' character

### 1.2.4 DOL or DCC

The User defined output offers an item called DOL (Distance Off Line) measured in meter (or actual data unit)

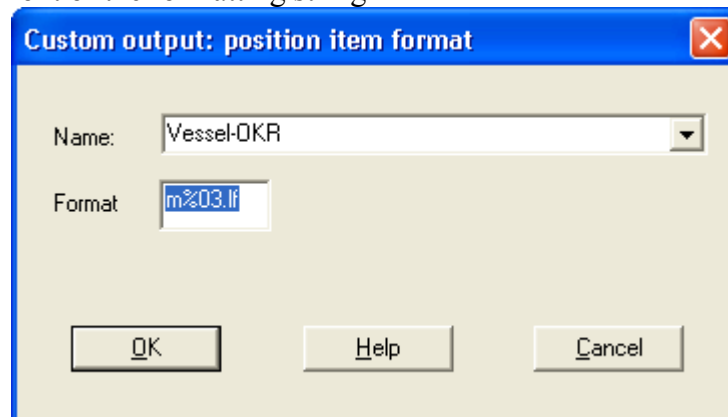


It may also output the value called DCC (Distance Cross Course) instead. To do so just put the character c (small caps) in front of the data formatting.

The output is default given in selected data unit (eg. Metric) – but you may select other formats if needed.

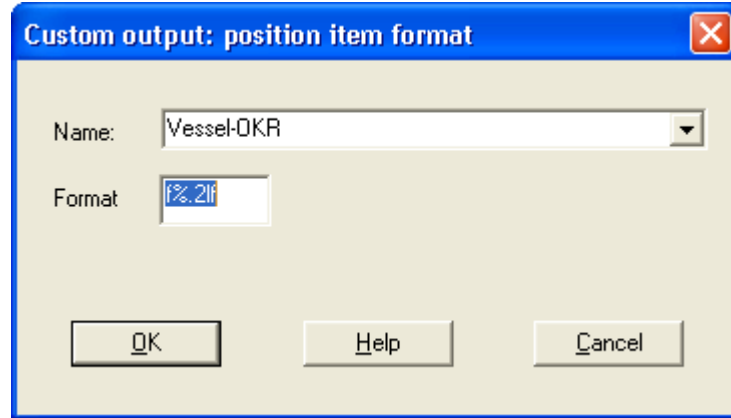
#### 1.2.4.1 Forced to CM

Enter an 'm' in front of the formatting string



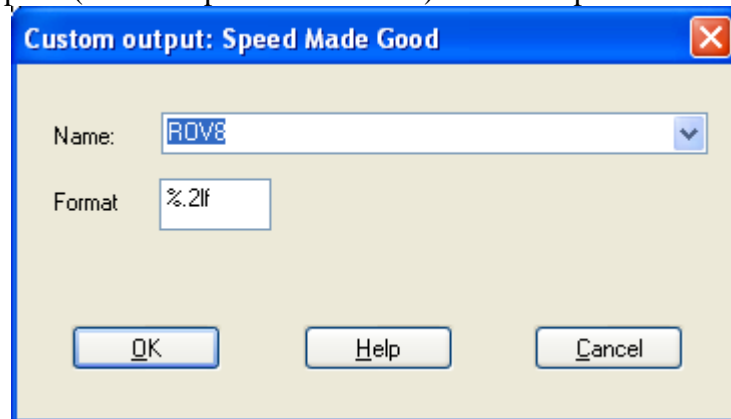
#### 1.2.4.2 Forced to Survey Feet

Enter an 'f' in front of the formatting string.



### 1.2.5 Speed

The calculated speed (SMG = Speed Made Good) can be output for a selected object.



The output will as default be given in knots, but you may choose other units by typing a special character in front of the format

- Km/h      Type k in front – eg k%.2lf
- m/s        Type s in front – eg s%.5lf
- m/min      Type m in front – eg m%.5lf

### 1.2.6 Range/Bearing to Waypoint

The output includes two fields

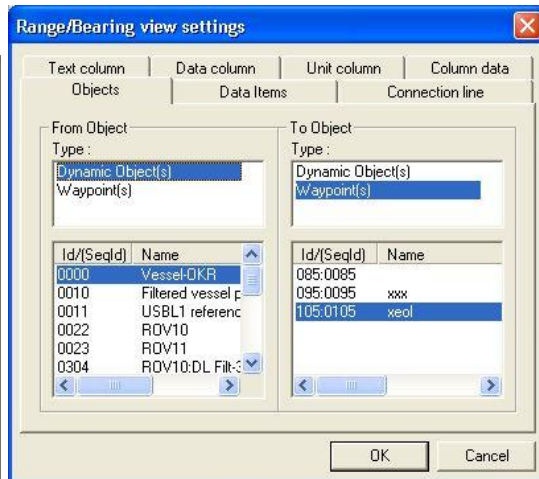
- Range to Waypoint
- Bearing to Waypoint (grid)

They are both position based – meaning that the operator selects an object to control the output.

The object must hereafter be assigned to a waypoint. This is done by opening a range/bearing view on the master Helmsman's Display:

```

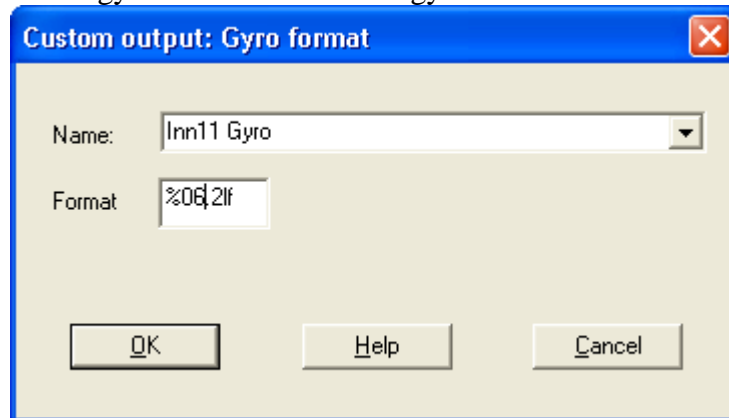
Rng      3705.17 m
TBrg     358.85 °
GBrg     357.79 °
From Vessel-OKR
To       xeol
    
```



The range bearing must be defined from an active object (the object selected in the item setup) to a waypoint.

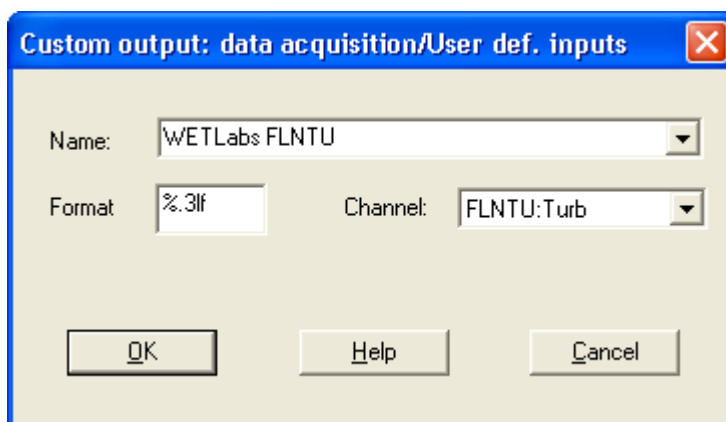
### 1.2.7 Gyro item

It can be selected which gyro to use and how the gyro values should be formatted.



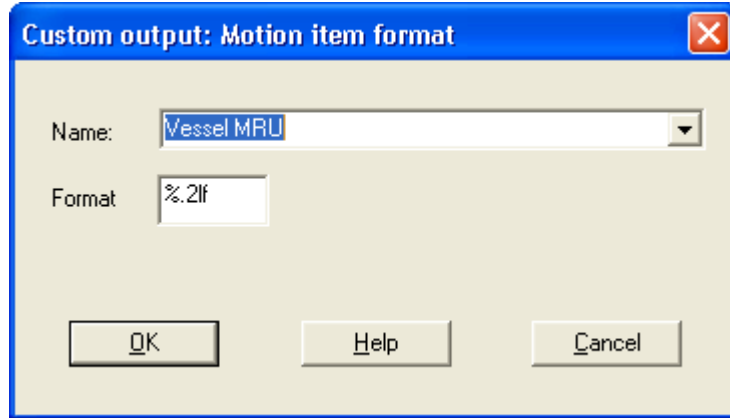
### 1.2.8 Data acquisition data item

It can be selected which echo sounder and channel should be output and how the format should be.



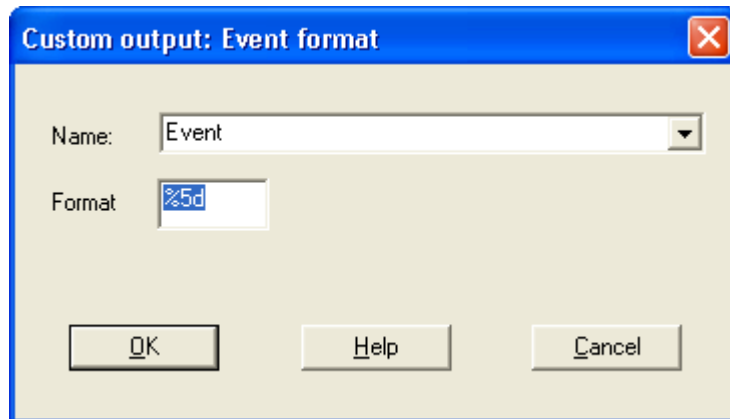
### 1.2.9 Motion item

If more motion sensors the sensor can be selected and which items (roll, pitch, heave) should be output and how the format should be.



### 1.2.10 Event item

It can be selected how the event number should be formatted.

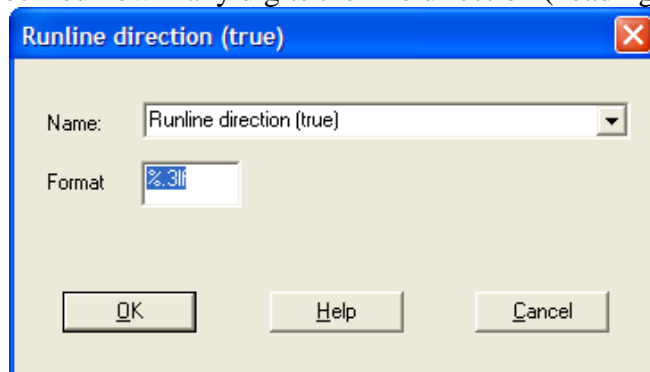


To output the event number enter eg. %05d to use 5 digits with leading zeros. It's also possible to include event text (manual event and userdefined events) by entering %05d,%s where %s represents the string with event text.

### 1.2.11 Run line direction

The actual bearing of the active line segment (or tangent of curve intersection point) can be output as either true heading or grid heading (the latter is compensated for meridian convergence)

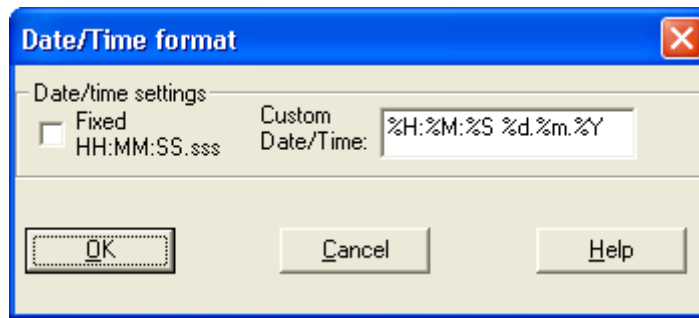
It can be specified how many digits the line direction (heading) must contain:



To output heading with leading zeros and two ciphers, then just set %06.2lf

### 1.2.12 Date & Time item

Date and time can also be defined (the same in LogData program).



Time is default set to the fixed format. If *Custom Date/Time* is selected, the user can define the format. See 2.6 for a detailed description on format definition.

**NOTE:**

The fixed format output's time as hh:mm:ss.sss - this will always be given as UTC time

The custom format is freely configured - and this shall be based on the local time - ie. reading whatever settings you may have on the pc

**Example:**

Specifying: %H:%M:%S %d.%m.%Y

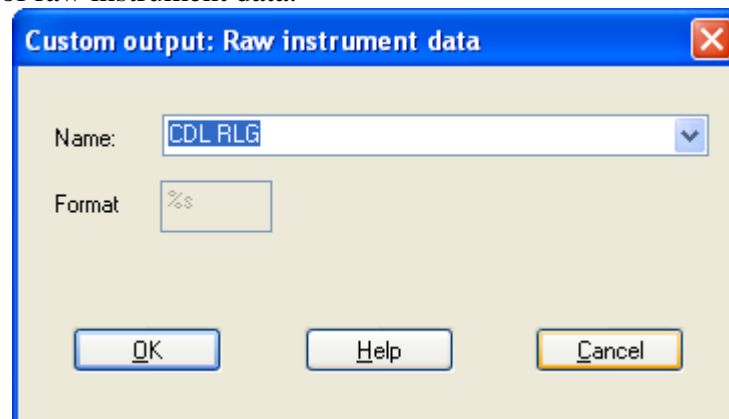
Will result in the following output string:

10:06:01 12.01.1998

If *Fixed* is selected the format will be HH:MM:SS

**1.2.13 Raw instrument output**

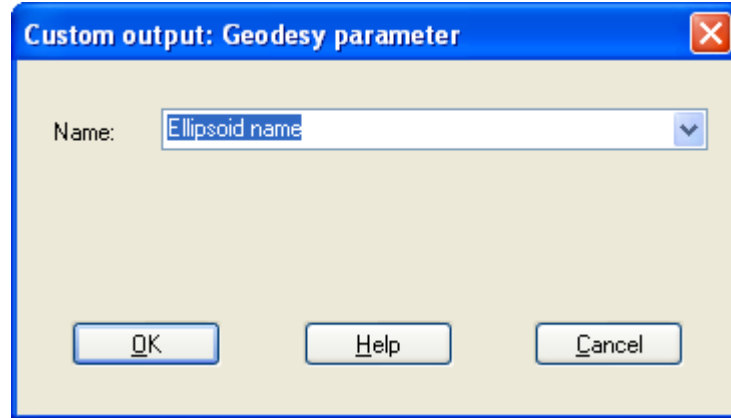
To output copy of raw instrument data:



Just select the instrument in action. It will not work for binary instruments.

**1.2.14 Geodesy information**

It's possible to output information on the selected geodesy:

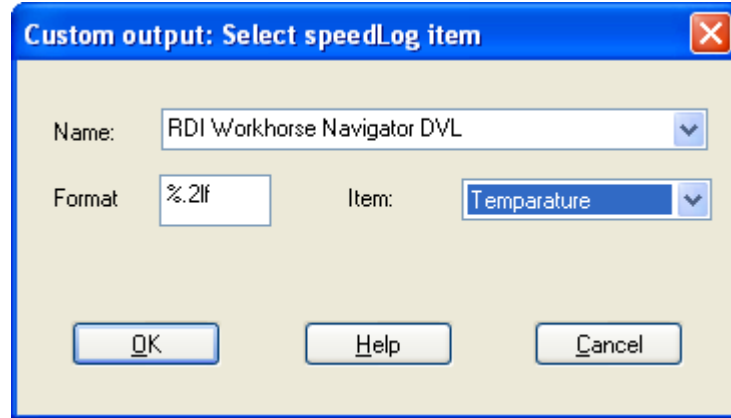


The following parameters will be available:

- Ellipsoid name  
Character string
- Semi major axis  
Floating point number
- Inverse flattening  
Floating point number
- Projection name  
Character string
- Scale at Origin                   /\* Point scale factor at the ellipsoid origin \*/  
Character string
- First parallel                    /\* First parallel in Lamberts conical - degree \*/  
ddd mm ss.ssss
- Second Parallel                 /\* Second parallel in Lamberts conical - degree \*/  
ddd mm ss.ssss
- Longitude at Origin           /\* Origin lambda (lon) on the reference ellipsoid [degree] \*/  
ddd mm ss.ssss
- Latitude at Origin             /\* Origin phi (lat) on the projection ellipsoid [degree] \*/  
ddd mm ss.ssss
- False Easting;                 /\* Origin easting on the projection plane [meters] \*/  
Floating point
- False Northing;               /\* Origin northing on the projection plane [meters] \*/  
Floating point
- UTM zone;                      /\* Derived from "Original Longitude" [1..60] \*/  
Integer

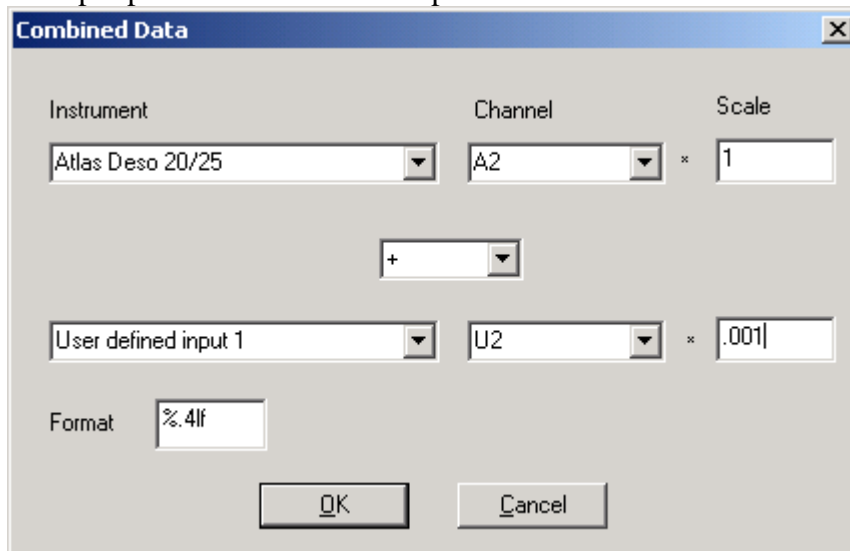
### 1.2.15 Speed log data

If a RDI speed log is attached, the system can output various parameters from the PDO data from the Doppler log:



### 1.2.16 Combined data acquisition

Via the Combined Data Acquisition item, you may combine two data values like depth and altitude or depth plus constant to one output.



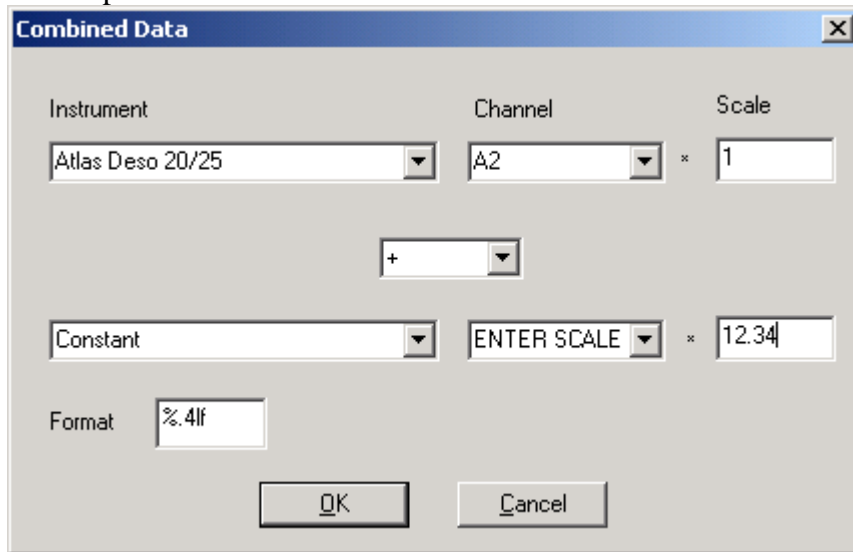
You must here select one or two instruments (the lists to the left) and channels for each of them. Finally you may scale the inputs if they are in different data units.

The middle part allows selection of operator

- + Plus
- Minus
- \* Multiply
- / Divide by
- NO None – item number two is ignored

The above example gives (Deso20 channel 2) plus (User defined input channel 2 divided by 1000).

A special instrument called Constant is defined in the lists:



The use of this result in just the value entered in Scale – eg. (Deso20 channel 2) plus 12.34.

### 1.3 Definition file

It must be noted that the final definition of user defined output is handled in some separate files (that is outside the normal definitions in the *gensetup.db* file. These files are located on `\eiva\navipac\setup` and named `userDef<index>.out` (eg. `userDef611.out`)

## 2. Custom output format strings

The format field for positions, depths etc follow the ANSI C syntax:

### Format Specification Fields: printf and wprintf Functions

A format specification, which consists of optional and required fields, has the following form:

**%**[*flags*] [*width*] [*.precision*] [{*h* | *l* | **I64** | **L**}]*type*

Each field of the format specification is a single character or a number signifying a particular format option. The simplest format specification contains only the percent sign and a *type* character (for example, `%s`). If a percent sign is followed by a character that has no meaning as a format field, the character is copied to **stdout**.

For example, to print a percent-sign character, use `%%`.

The optional fields, which appear before the *type* character, control other aspects of the formatting, as follows:

*type*

Required character that determines whether the associated *argument* is interpreted as a character, a string, or a number (see [Table R.3](#)).

*flags*

Optional character or characters that control justification of output and printing of signs, blanks, decimal points, and octal and hexadecimal prefixes (see [Table R.4](#)). More than one flag can appear in a format specification.

*width*

Optional number that specifies the minimum number of characters output. (See [printf Width Specification](#).)

*precision*

Optional number that specifies the maximum number of characters printed for all or part of the output field, or the minimum number of digits printed for integer values (see [Table R.5](#)).

**h | l | I64 | L**

Optional prefixes to *type*-that specify the size of *argument* (see [Table R.6](#)).

## 2.1 printf Type Field Characters

The *type* character is the only required format field ; it appears after any optional format fields. The *type* character determines whether the associated argument is interpreted as a character, string, or number. The types **C** and **S**, and the behaviour of **c** and **s** with **printf** functions, are Microsoft extensions and are not ANSI-compatible.

**Table R.3 printf Type Field Characters**

Character	Type	Output Format
<b>c</b>	<b>int</b> or <b>wint_t</b>	When used with <b>printf</b> functions, specifies a single-byte character; when used with <b>wprintf</b> functions, specifies a wide character.
<b>C</b>	<b>int</b> or <b>wint_t</b>	When used with <b>printf</b> functions, specifies a wide character; when used with <b>wprintf</b> functions, specifies a single-byte character.
<b>d</b>	<b>int</b>	Signed decimal integer.
<b>i</b>	<b>int</b>	Signed decimal integer.
<b>o</b>	<b>int</b>	Unsigned octal integer.
<b>u</b>	<b>int</b>	Unsigned decimal integer.
<b>x</b>	<b>int</b>	Unsigned hexadecimal integer, using "abcdef."
<b>X</b>	<b>int</b>	Unsigned hexadecimal integer, using "ABCDEF."
<b>e</b>	<b>double</b>	Signed value having the form [ - ] <i>d.ddd e</i> [ <i>sign</i> ] <i>ddd</i> where <i>d</i> is a single decimal digit, <i>ddd</i> is one or more decimal digits, <i>ddd</i> is exactly three decimal digits, and <i>sign</i> is + or -.
<b>E</b>	<b>double</b>	Identical to the <b>e</b> format except that <b>E</b> rather than <b>e</b> introduces the exponent.
<b>f</b>	<b>double</b>	Signed value having the form [ - ] <i>ddd.dddd</i> , where <i>ddd</i> is one or more decimal digits. The number of digits before the decimal point depends on the magnitude of the number, and the number of digits after the decimal point depends on the requested precision.
<b>g</b>	<b>double</b>	Signed value printed in <b>f</b> or <b>e</b> format, whichever is more compact for the given value and precision. The <b>e</b> format is used only when the exponent of the value is less than -4 or greater than or equal to the precision argument. Trailing zeros are truncated, and the decimal point appears only if one or more digits follow it.
<b>G</b>	<b>double</b>	Identical to the <b>g</b> format, except that <b>E</b> , rather than <b>e</b> , introduces the exponent (where appropriate).
<b>n</b>	Pointer to integer	Number of characters successfully written so far to the stream or buffer; this value is stored in the integer whose address is given as the argument.
<b>p</b>	Pointer to <b>void</b>	Prints the address pointed to by the argument in the form <i>xxxx:yyyy</i> where <i>xxxx</i> is the segment and <i>yyyy</i> is the offset, and the digits <i>x</i> and <i>y</i> are uppercase

		hexadecimal digits.
<b>s</b>	String	When used with <b>printf</b> functions, specifies a single-byte-character string; when used with <b>wprintf</b> functions, specifies a wide-character string. Characters are printed up to the first null character or until the <i>precision</i> value is reached.
<b>S</b>	String	When used with <b>printf</b> functions, specifies a wide-character string; when used with <b>wprintf</b> functions, specifies a single-byte-character string. Characters are printed up to the first null character or until the <i>precision</i> value is reached.

## 2.2 Flag Directives

The first optional field of the format specification is *flags*. A flag directive is a character that justifies output and prints signs, blanks, decimal points, and octal and hexadecimal prefixes. More than one flag directive may appear in a format specification.

**Table R.4 Flag Characters**

Flag	Meaning	Default
-	Left align the result within the given field width.	Right align.
+	Prefix the output value with a sign (+ or -) if the output value is of a signed type.	Sign appears only for negative signed values (-).
0	If <i>width</i> is prefixed with <b>0</b> , zeros are added until the minimum width is reached. If 0 and - appear, the <b>0</b> is ignored. If <b>0</b> is specified with an integer format ( <b>i</b> , <b>u</b> , <b>x</b> , <b>X</b> , <b>o</b> , <b>d</b> ) the <b>0</b> is ignored.	No padding.
<i>blank</i> (' ')	Prefix the output value with a blank if the output value is signed and positive; the blank is ignored if both the blank and + flags appear.	No blank appears.
#	When used with the <b>o</b> , <b>x</b> , or <b>X</b> format, the # flag prefixes any nonzero output value with 0, 0x, or 0X, respectively.	No blank appears.
	When used with the <b>e</b> , <b>E</b> , or <b>f</b> format, the # flag forces the output value to contain a decimal point in all cases.	Decimal point appears only if digits follow it.
	When used with the <b>g</b> or <b>G</b> format, the # flag forces the output value to contain a decimal point in all cases and prevents the truncation of trailing zeros. Ignored when used with <b>c</b> , <b>d</b> , <b>i</b> , <b>u</b> , or <b>s</b> .	Decimal point appears only if digits follow it. Trailing zeros are truncated.

## 2.3 printf Width Specification

The second optional field of the format specification is the width specification. The *width* argument is a nonnegative decimal integer controlling the minimum number of characters printed. If the number of characters in the output value is less than the specified width, blanks are added to the left or the right of the values — depending on whether the - flag (for left alignment) is specified — until the minimum width is reached. If *width* is prefixed with 0, zeros are added until the minimum width is reached (not useful for left-aligned numbers).

The width specification never causes a value to be truncated. If the number of characters in the output value is greater than the specified width, or if *width* is not given, all characters of the value are printed (subject to the [precision](#) specification). If the width specification is an asterisk (\*), an **int** argument from the argument list supplies the value. The *width* argument must precede the value being formatted in the argument list. A nonexistent or small field width does not cause the truncation of a field; if the result of a conversion is wider than the field width, the field expands to contain the conversion result.

## 2.4 How Precision Values Affect Type

Table R.5

Type	Meaning	Default
<b>c, C</b>	The precision has no effect.	Character is printed.
<b>d, i, u, o, x, X</b>	The precision specifies the minimum number of digits to be printed. If the number of digits in the argument is less than <i>precision</i> , the output value is padded on the left with zeros. The value is not truncated when the number of digits exceeds <i>precision</i> .	Default precision is 1.
<b>e, E</b>	The precision specifies the number of digits to be printed after the decimal point. The last printed digit is rounded.	Default precision is 6; if <i>precision</i> is 0 or the period (.) appears without a number following it, no decimal point is printed.
<b>f</b>	The precision value specifies the number of digits after the decimal point. If a decimal point appears, at least one digit appears before it. The value is rounded to the appropriate number of digits.	Default precision is 6; if <i>precision</i> is 0, or if the period (.) appears without a number following it, no decimal point is printed.
<b>g, G</b>	The precision specifies the maximum number of significant digits printed.	Six significant digits are printed, with any trailing zeros truncated.
<b>s, S</b>	The precision specifies the maximum number of characters to be printed. Characters in excess of <i>precision</i> are not printed.	Characters are printed until a null character is encountered.

If the argument corresponding to a floating-point specifier is infinite, indefinite, or NaN, **printf** gives the following output.

Value	Output
+ infinity	<b>1.#INF</b> <i>random-digits</i>

– infinity	<b>-1.#INF</b> <i>random-digits</i>
Indefinite (same as quiet NaN)	<i>digit.#IND</i> <i>random-digits</i>
NAN	<i>digit.#NAN</i> <i>random-digits</i>

## 2.5 Size Prefixes for printf and wprintf Format-Type Specifiers

Table R.6

To Specify	Use Prefix	With Type Specifier
long int	<b>l</b>	<b>d, i, o, x, or X</b>
long unsigned int	<b>l</b>	<b>u</b>
short int	<b>h</b>	<b>d, i, o, x, or X</b>
short unsigned int	<b>h</b>	<b>u</b>
<b>__int64</b>	<b>I64</b>	<b>d, i, o, u, x, or X</b>
Single-byte character with <b>printf</b> functions	<b>h</b>	<b>c or C</b>
Single-byte character with <b>wprintf</b> functions	<b>h</b>	<b>c or C</b>
Wide character with <b>printf</b> functions	<b>l</b>	<b>c or C</b>
Wide character with <b>wprintf</b> functions	<b>l</b>	<b>c or C</b>
Single-byte – character string with <b>printf</b> functions	<b>h</b>	<b>s or S</b>
Single-byte – character string with <b>wprintf</b> functions	<b>h</b>	<b>s or S</b>
Wide-character string with <b>printf</b> functions	<b>l</b>	<b>s or S</b>
Wide-character string with <b>wprintf</b> functions	<b>l</b>	<b>s or S</b>

Thus to print single-byte or wide-characters with **printf** functions and **wprintf** functions, use format specifiers as follows.

To Print Character As	Use Function	With Format Specifier
single byte	printf	<b>c, hc, or hC</b>
single byte	wprintf	<b>C, hc, or hC</b>
wide	wprintf	<b>c, lc, or lC</b>
wide	printf	<b>C, lc, or lC</b>

To print strings with **printf** functions and **wprintf** functions, use the prefixes **h** and **l** analogously with format type-specifiers **s** and **S**.

## 2.6 Date & Time format

The format argument consists of one or more codes; based on the c/c++ printf, the formatting codes are preceded by a percent sign (%). Characters that do not begin with % or # are copied unchanged to logfile or output. The formatting codes for date and time in custom logfiles are listed below:

*%a Abbreviated weekday name*

*%A Full weekday name*

*%b Abbreviated month name*

*%B Full month name*

*%c Date and time representation appropriate for locale*

*%d Day of month as decimal number (01 - 31)*

*%H Hour in 24-hour format (00 - 23)*

*%I Hour in 12-hour format (01 - 12)*

*%j Day of year as decimal number (001 - 366): (=Julian day)*

*%m Month as decimal number (01 - 12)*

*%M Minute as decimal number (00 - 59)*

*%p Current locale's A.M./P.M. indicator for 12-hour clock*

*%S Second as decimal number (00 - 59)*

*#s Seconds past midnight*

*%U Week of year as decimal number, with Sunday as first day of week (00 - 51)*

*%w Weekday as decimal number (0 - 6; Sunday is 0)*

*%W Week of year as decimal number, with Monday as first day of week (00 - 51)*

*%x Date representation for current locale*

*%X Time representation for current locale*

*%y Year without century, as decimal number (00 - 99)*

*%Y Year with century, as decimal number*

*%z, %Z Time-zone name or abbreviation; no characters if time zone is unknown*

*%% Percent sign*