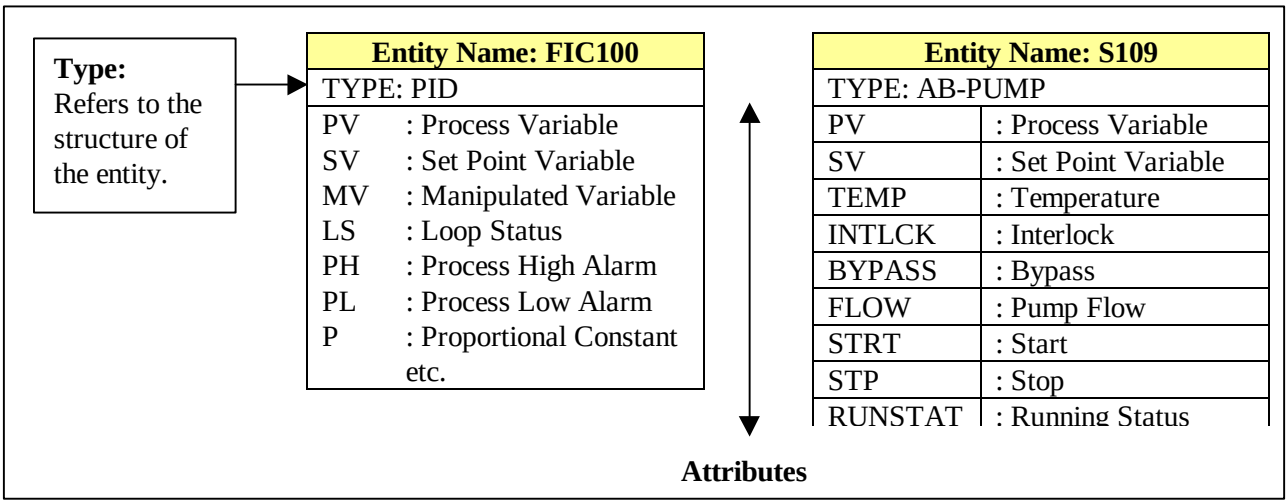
Table of Contents

List of Tables

8 Types

8.1 All About Types

In the Entities Chapter we discussed the concepts of Entities, Attributes and Types. The diagram



shows an example of two entities and the relationship between these three terms.

Summary

The example above shows two entities (FIC100 and S109).

- An entity is a group of attributes associated with a given physical function and assigned a single entity name.
- The **attributes** are the individual data elements that make up the entity. In this example, some of the attributes of **FIC100** are similar to the attributes of S109, e.g. PV, SV while others are different. **TEMP, INTLCK, BYPASS** are obviously special attributes for the speed control entity **S109**.
- The **type** or type definition is the structure of each entity.
- The **type** defines the number of attributes, what each attribute is called, the format of each attribute, etc.

In the above example, by saying that the entity **FIC100** has a type **PID** we are immediately aware of all of its attributes. i.e. If it is a **PID** type, then it must have a **PV, SV, MV, LS**, etc.

Once a type has been defined, you can have as many entities of that type as you require.

For example, if you have 200 motors on a site and each motor has seven attributes such as **START, STOP, RUN, OTEMP**, etc., to configure the motors, you would configure 200 entities all of type MOTOR.

In ***MacroView***, you can use the standard type definitions delivered with the drivers.

Alternatively, as this chapter explains, you can configure your own types.

In essence, this is a process of:

i. Choosing a type name.

ii. Making up the type by adding attributes.

8.2 Why Create Your Own Types?

In Process Control and SCADA, you can very often break down the process into a number of manageable functions.

- These functions can be "industry standard" functions like PID controllers, Analogue Inputs, etc. and, particularly with PLC systems, they can also be specific to your operation.

  For example, your corporation may have a very definite way of controlling the starting and stopping of motors, what interlocks are involved, etc. and this methodology is reflected in the PLC ladder logic.

- The conventional way of representing this site-specific motor is to assign a separate tag and a name to each of the individual inputs and outputs of every motor. I.e. for 200 motors, 7 inputs and 3 outputs you have 2000 tags.
- However, with ***MacroView*** you define a single motor type, which represents the motor. The type defines the inputs and outputs associated with the motor and then creates one entity per motor. I.e. 200 entities instead of the 2000 tags.

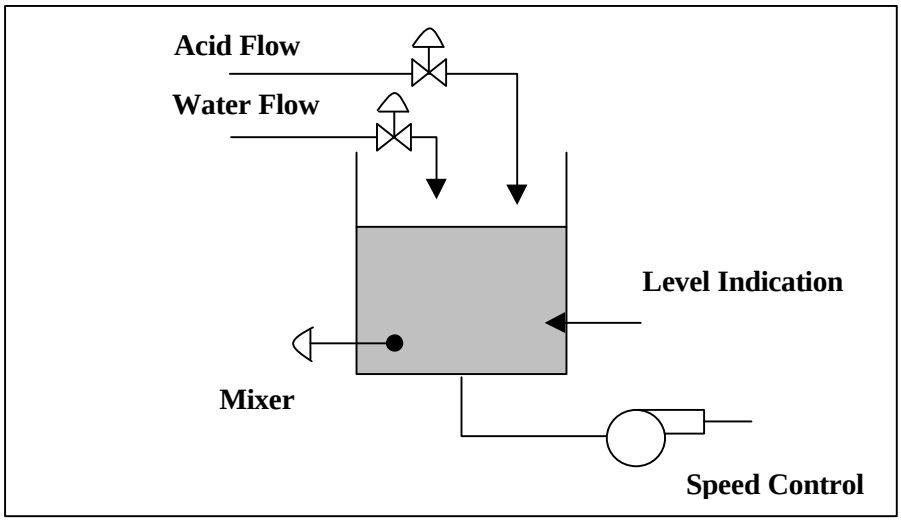
  Additionally, you may design a single group faceplate and a single detail faceplate, which can now be used to represent all 200 motors in the same format.

  This means that your operators always look at the motors the same way and control the motors in the same standard way.

- Of course, you can have as many types as you need to accurately reflect the functions in your process. For example of the 200 motors, there may be 5 different kinds. In this case, you need to define 5 types.
- You may also have as many attributes as required to make up the type. In general, to keep the types manageable, it is a good idea to keep the number of attributes per type below 20.

8.3 Designing Types

To design the types needed for your process, first look at the process and break it down into manageable groups of inputs and outputs that are associated with a given function.



In this example, the functions are Flow Control, Mixer Control, Level Indication and Speed Control.

The Flow Control and Level Indication are standard PID and Analogue Input types but the Mixer Control and Speed Control are corporate specific. I.e. we need two new types to cover all the entities of these types in the corporation.

Now, for the Speed Control type you associate each input or output to an attribute name, e.g. the trip digital input will be given the name TRIP, the interlock input will be given the name INTRLCK etc.

Note: Internal (non-I/O) PLC registers are also given names. Once you have decided on the attributes of the type, you are ready to configure the type into the system.

8.4 Configuring the Type

You configure the type by filling in the blanks in the engineering configurator. This is essentially a two-stage process.

i. Adding the `type name` to the system.

ii. Adding and defining each attribute of the new type.

To define the attribute, you need to specify:

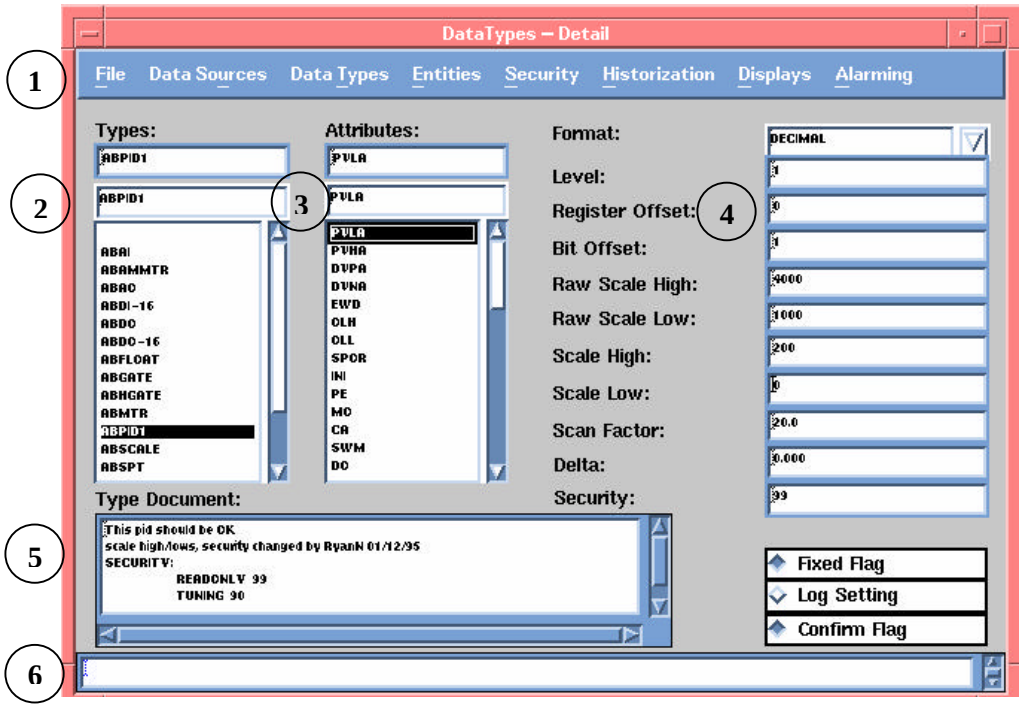
- How ***MacroView*** is to locate the attribute information. (This is basically a register/bit offset from the entity address.)
- How ***MacroView*** is to **convert** the data into Engineering Units. (E.g. is the raw data in floating point, digital, or scale format.)
- Some other characteristics of the attribute. (E.g. the security of the attribute, whether you want changes logged, etc.)

All this information is entered directly into the blanks in the configurator entry screens.

8.5 The Types Configuration Screen

To get to the types configuration screen, just select ***Data Types:Detail***.

Types Configuration Options



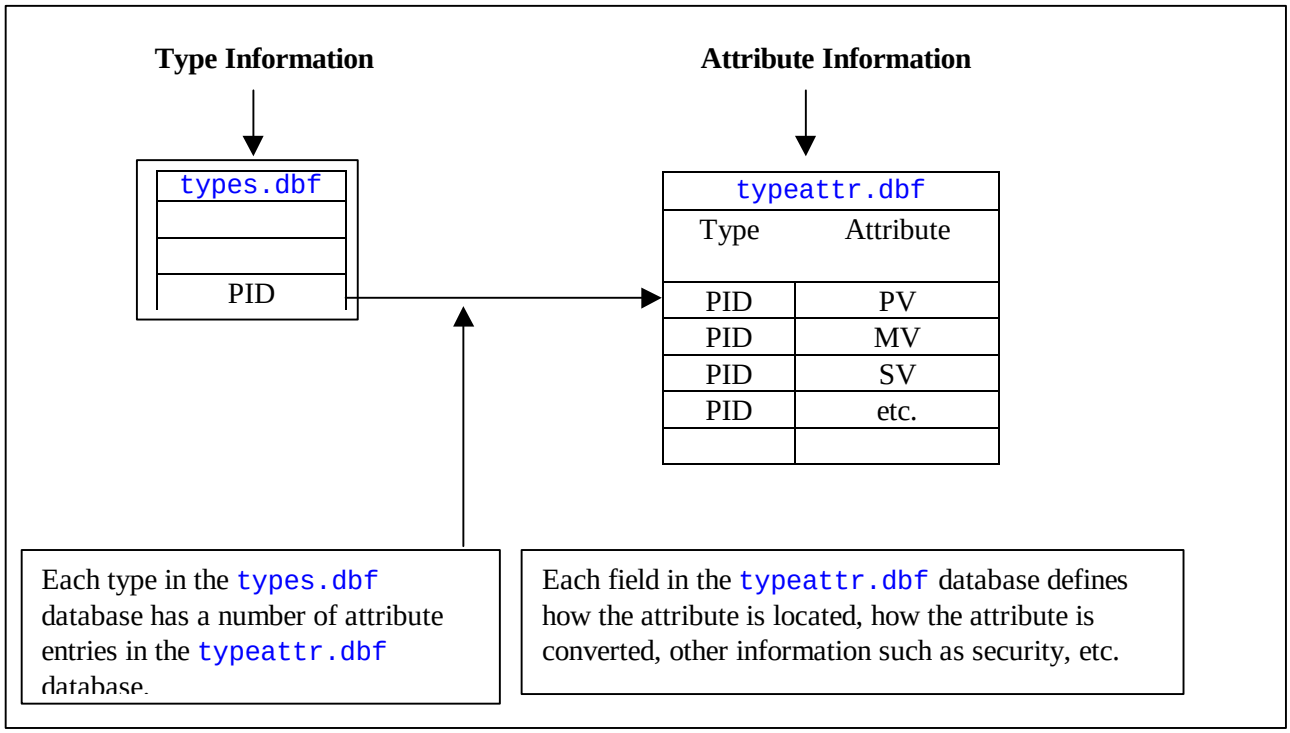
Types Configuration Options

1. **Menu:** Provides a means of Adding Blank types or attributes, Adding Like (cloning), and Deleting Types and Attributes.
2. **Types Selection List:** Click on the type with which you wish to work. The list shows the available list in the `types.dbf` database. Once you have selected a type, the attributes will appear in the attribute list.
3. **Attributes List:** The attribute list shows the attributes that are configured for the selected type. Each entry in the list corresponds to a record in the typeattr.dbf database. To add an attribute, just use the menu option.

   Select the attribute you wish to configure by clicking on the attribute in the list. The items associated with the selected attribute will appear in the detail area.
4. **Attribute Detail Area:** The attribute detail area shows the various characteristics assigned to the attribute. The next sections discuss each of these characteristics in detail.
5. **Type Document:** Enter any comments that you feel are of interest in this area. This information is not used by the system.
6. **Message Area:** Suggestions, Help and Error messages are sent to this area to assist you in your configuration.

8.6 How *MacroView* Uses the Type Information

When you are entering the type and attribute information into the configurator, you are basically adding records to two databases, each type has an entry in the `types.dbf` dBase file and each attribute for that type has an entry in the `typeattr.dbf` dBase file.



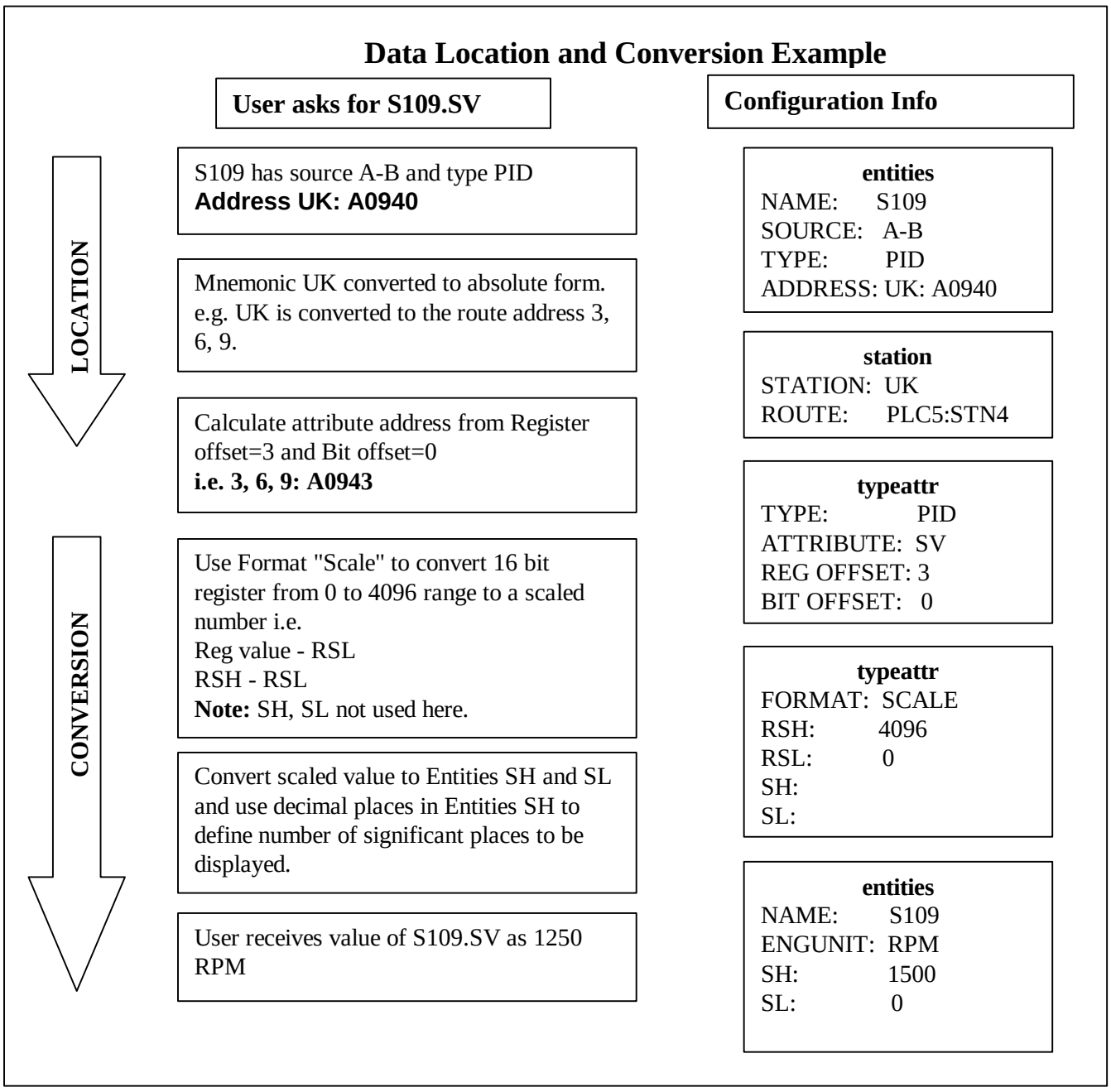
Attribute Characteristics Summary

Each characteristic of the attribute is discussed in more detail in the next sections. The table below shows a brief summary of these characteristics.

Table 1: `typeattr.dbf` Database

Typeattr.dbf Field	Field	Description
	TYPE	The name of the type associated with this attribute.
	ATTRIBUTE	The name of this attribute.
LOCATION	LEVEL	Which start address this attribute is related to.
	REGISTER OFFSET	How many bits from the start address the attribute is positioned.
	BIT OFFSET	How many bits from the start address the attribute is positioned.
CONVERSION	FORMAT	The format of the raw data.
	RAW SCALE HIGH	The raw or "native" scale high.
	RAW SCALE LOW	The raw or "native" scale low.
	SCALE HIGH	The high scale overrides to get to the engineering units
	SCALE LOW	The low scale overrides to get to the engineering units.
OTHER CHARACTERISTICS	DELTA	The amount the value should be incremented when the fast ramp function is used.
	SECURITY	The security level of the attribute.
	SCAN FACTOR	For DCS systems, the multiple of the entity scan time at which this attribute will be scanned.
	FIXED FLAG	In a simulation, whether this attribute is to be varied or not.
	LOG SETTING	If an operator changes the value, do you want to log this fact?
	CONFIRM FLAG	Do you require that an operator confirms a change before it is carried out?

8.7 Type and Attribute Configuration Specifics



Note: The configuration of the attributes is related not only to the process requirements, but also to the nature of the source.

For example, a PLC image type driver requires register location information, whereas a sorted image driver does not need this information because the driver asks for the data by name and not register number.

8.8 Type and Attribute Configuration.

The Type and Attribute Configuration Pages on the following pages explain in detail just how the types and their attributes are configured.

Once the attributes have been added, you need to:

- Configure the **location** details.
- Configure the **conversion** details.
- Configure the **additional information** such as security.

Adding Type and Attributes

This task is carried out by:

- First adding the Type and then adding each of the Attributes after that.

Adding Types

What you type

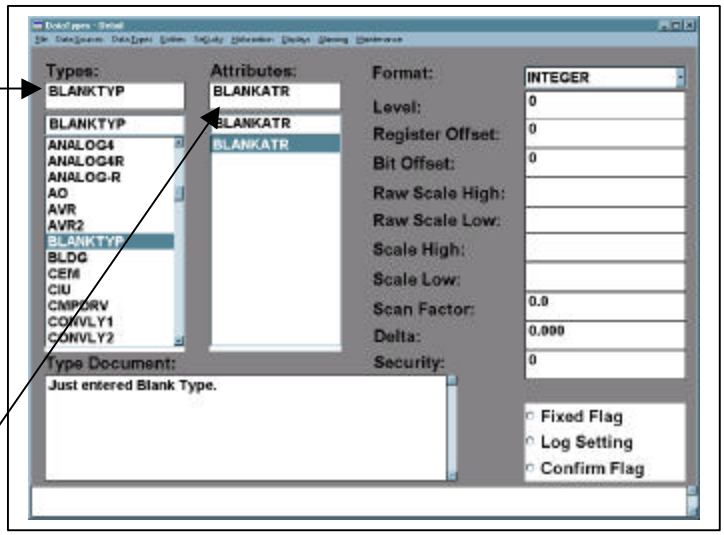
Using the ***Data Types:Detail*** then ***Add Blank:Types*** menu option will add a new Type called "**BLANKTYP**" to the Type list.
Select **BLANKTYP** from the Type list and, using the top editline, edit the name "**BLANKTYP**" to that of the new Type you are creating.

How it Works

The new Type will have one attribute called "**BLANKATR**" and you can use the ***Data Types:Detail*** then ***Add Blank:Attributes*** menu option to add more attributes to this Type.

Things to Note

The configurator will only allow you to add one Type at a time, as the new Type is given the name "BLANKTYP" and you must edit this before you are able to add a new one.



Adding Attributes

What you type

Using the ***Data Types:Detail*** then ***Add Blank:Attributes*** menu option will add a new Attribute called "**BLANKATR**" to the Attribute list.
Select **BLANKATR** from the Attribute list and, using the top editline, edit the name "**BLANKATR**" to that of the new Attribute you are creating. With this new Attribute selected, you may now use the editing options to set up the details of the attribute, as described in the following part of this chapter.

Things to Note

While the configurator will only allow the addition of one Type at a time, it will allow the adding of multiple "BLANKATR" and you would then edit each of them. You may be able to save configuration time by using the ***Data Types:Detail*** then ***Add Like:Attributes*** menu option and then making the minor editing changes to each Attribute.

Main Type Configuration (Type Name, Documentation)

Here we add the type name and documentation of the entity types.

Type Name

What you type

Enter the name of the new TYPE.

How it Works

The type name is used:
1. To cross reference all references to the name appearing in the entity.
2. As the name for the group and detail faceplate graphics.

Example

A-BPID, SPDCTL
Use a name that is easy to remember and a good representation of the type.

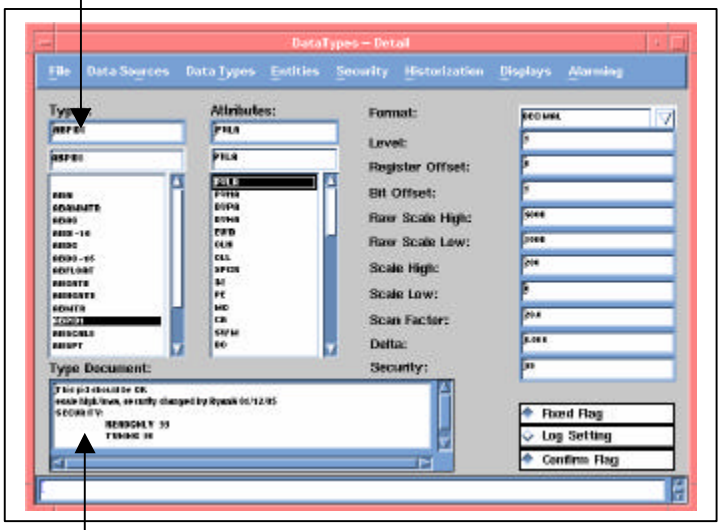
How to get there

1 ***Data Types:Detail*** This brings up the Types Detail screen.

2 Click on the Type to be modified: The attributes will appear in the central window.

3 Click on the *attribute* to be modified. The attribute items will appear in the detail area.

4 Alternatively, you may add a blank type or attribute using: ***Data Types:Add Blank:Types or Attributes*** and edit the new record.



Documentation

What you type

Type in any documentation in the documentation area.
You may want to document how the **TYPE** is to be used, what to watch out for etc.
Now that the main **TYPE** information is entered, you need to define the attributes.

Edit Attribute Screen (Attribute, Level)

The next two pages shows how you configure the items used to ***locate*** the attribute raw value for a PLC image Source.

Attribute

What you type

If using the ***Add Like*** option you type in the new Attribute name, or edit the "BLANKATR" name to the new name of the attribute. You may use up to 8 characters. This name together with the entity name is all a user need know to get to the attribute.

Example

PV, LS, OVERTEMP
Choose an attribute name that is pronounceable and accurately represents the function of the attribute.

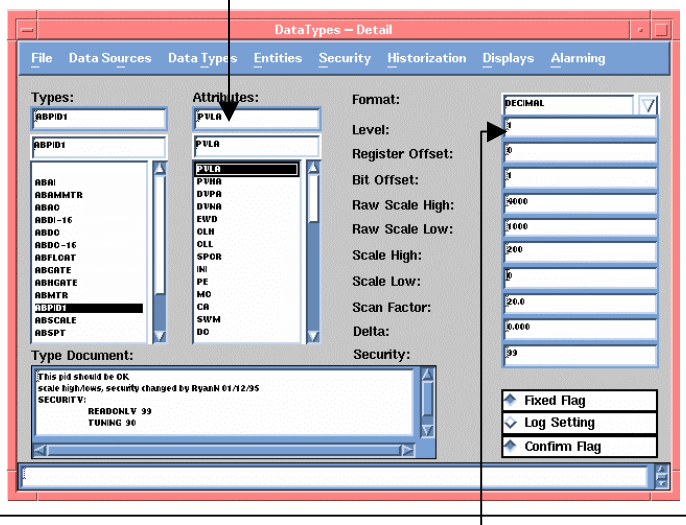
How to get there

1 ***Data Types:Detail*** This brings up the Types Detail screen.

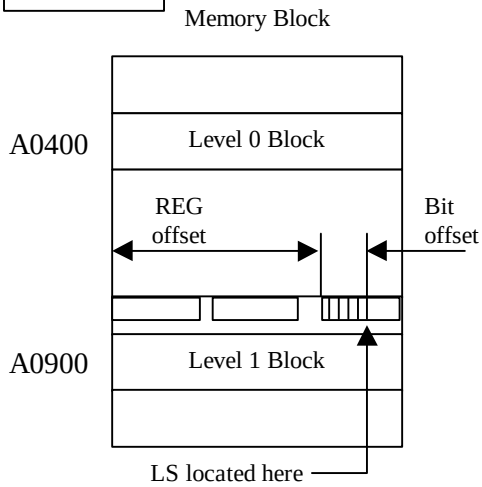
2 Click on the Type to be modified: The attributes will appear in the central window.

3 Click on the *attribute* to be modified. The attribute items will appear in the detail area.

4 Alternatively, you may add a blank type or attribute using : ***Data Types:Add Blank:Types or Attributes*** and edit the new record.



Example



Level

What you type

Choose the level or address in the entity address configuration that corresponds to the block of memory that holds this attribute.

Example

For example, if the **Entity** address is UK:A0400;UK:A0900 and the attribute is in the second address, set Level to 1.
In this example, the LS attribute is located in the second block of registers, i.e. LEVEL 1.

How it Works

Entity data is often located in two or more blocks of the memory

Edit Attribute (Raw Scales, Scales and Format)

This page shows how to configure the conversion characteristics of the attribute so that the raw register form is converted into a user recognisable attribute.

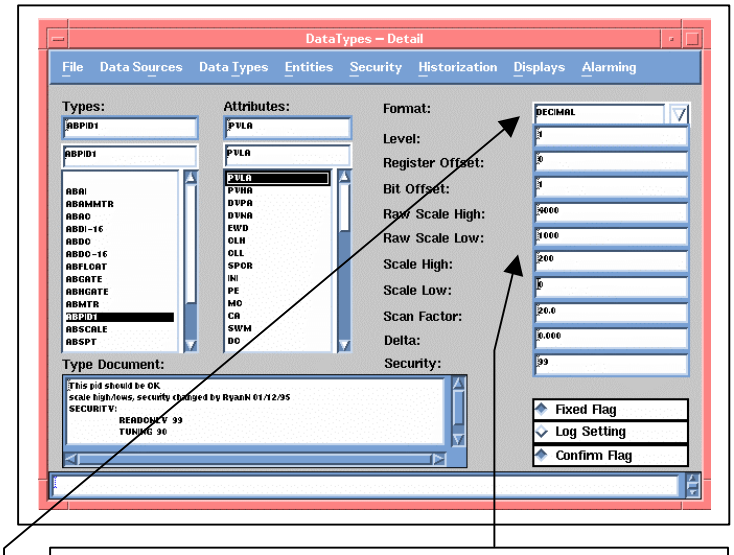
How to get there

1 ***Data Types:Detail*** This brings up the Types Detail screen.

2 Click on the Type to be modified: The attributes will appear in the central window.

3 Click on the *attribute* to be modified. The attribute items will appear in the detail area.

4 Alternatively, you may add a blank type or attribute using: ***Data Types:Add Blank:Types or Attributes*** and edit the new record.



Raw Scale High and Low, Scale High and Low

After you have chosen the FORMAT field, use the information in Table 2: Attribute Format Overview of this Chapter, which details the meaning of the scales for this format.
Remember the scales have different meanings for different FORMATS. In general, the Raw scales are used for the first level of conversion and the scales for the next level of conversion.

Format

Select one of the Format options. The Format field tells the system how to convert the raw image into a user recognizable form. The TABLE on the next page gives a more detailed description of these formats and how they use the raw scales and scales in the conversion process.

SCALE	:	Scales a raw PLC value.
DIGITAL	:	Converts a bit to a digital 0 or 1.
FLOAT	:	Converts a raw value to an integer.
INTERGER:		Converts a raw value to an integer.
DECIMAL:		Converts to an integer and then scales it by a factor of 10.
STATUS	:	Converts a bit to one of two strings.
STATUS2	:	Converts two bits to one of four strings.
PULSE	:	Converts a bit to a zero or one. Setting of the bit always returns to zero.
BCD	:	Converts a raw 16 bit BCD word.
IMAGE	:	Remains unchanged. (used for sorted image drivers like DCS systems.)
TEXT		Used to identify an ASCII string in a

Example

In the example above, the data is first scaled between 0-4096 with the raw scales to get a number between 0 and 1. Because it is the MV it is then scaled to a number between 0 and 100%.
Note: If you use a TEXT format, then the scales have entirely different meanings. Check the example on TEXT attribute configuration

Edit Attribute (Reg & Bit Offset)

The register and bit offsets define the offset (distance) from the start address and the position of the attribute in the memory map.

Reg Offset

What you type

Enter the number of registers further on from the entity block start address.

Example

In this example, the attribute LS is in the second address Level=1, and is 2 registers on from this address. i.e. A902

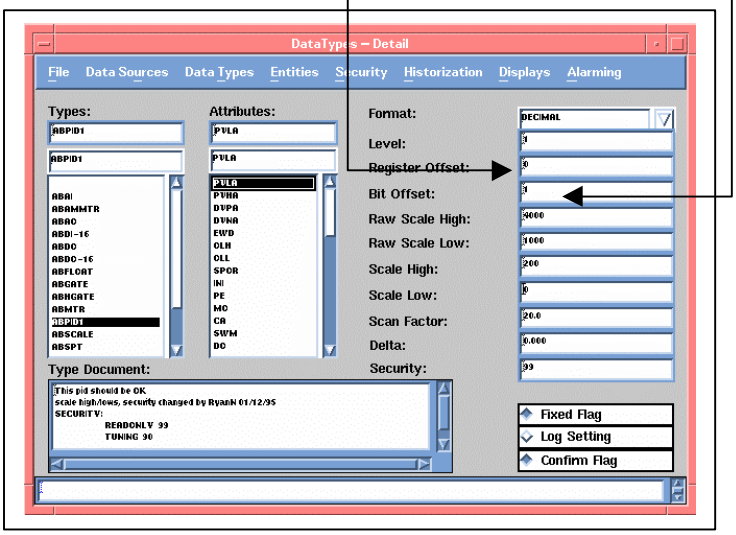
Bit Offset

What you type

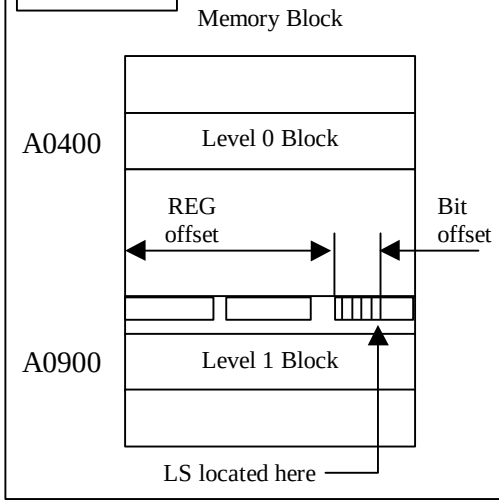
Enter the number of bits further on into the register block at the start of the attribute data.

Example

In this example, the LS bit is located in the 5th bit, 4 bits on from the start bit of the register.



Example



Note: If you have selected your format field to be image, for a sorted image source, these fields have no meaning.
For TEXT type sources, these fields (LEVEL, REG OFFSET, BIT OFFSET) have very different meanings. See the section on TEXT Attribute configuration.

Table 2: Attribute Format Overview

Format	Converts From	Converts To	Raw Scale Low (RSL)	Raw Scale High (RSH)	Attribute Scale Low (SL)	Attribute Scale High (SH)	Comments
Scale (e.g. MV)	Raw 16 bit register (Reg) scaled to PLC scale 1432	 34.96	First Conversion Low 0	First Conversion High 4096	Second conversion Low 0	Second conversion High 100	Uses (Reg Val-RSL)*(SH-SL) +SL (RSH-RSL) If the attribute SH and SL are blank, the entity SH and SL are used in their place. (See Note on SCALE)
Status (otemp)	Raw PLC bit 1	0 = RSL 1 = RSH HOT	0 String COOL	1 String HOT	N/A	N/A	Uses the Raw scales to determine the strings that are displayed.
Status2 (e.g. LS)	2 Consecutive bits 10	00 = RSL 01 = RSH 10 = SL 11 = SH CAS	00 String MAN	01 String AUT	10 String CAS	11 String COMP	The two bits must occur in the same register.
Integer (e.g. MV)	Raw 16 bit Register 84	2s compliment 2 byte integer 65	Used as Low clamp 0	Used to high clamp 100	N/A	N/A	See Note 2.
Decimal	Raw Integer Value 8734	Decimal value scaled by factor of 10 87.34	N/A	N/A	Used as Low clamp 0.00	Used as High clamp & scale factor 100.0	The scale factor is taken from the number of decimal places of the attribute SH if present, otherwise by the entity SH. Also see Note 2.
Float	IEEE 4 byte floating value	Real value 4269.3	N/A	N/A	Low clamp 200.0	High clamp # of significant figures 1500.0	See Notes 2 and 3.
Digital	Raw PLC bit 1	Value 0 or 1 1	N/A	N/A	N/A	N/A	
Pulse	Raw PLC bit 1	Value 0 or 1 1		N/A Number of milliseconds Pulse is high 1000	N/A	N/A	If an operator sets the value to a 1, then the value will be reset to 0 after a time RSH milliseconds (If RSH is blank, it will reset after 500 milliseconds). See Note on pulse format.
BCD	4 sets of 4 bit BCD values 8196	Real world value	N/A	N/A	Used as Low clamp	Used as High clamp	See Note 2.
Image (e.g. LS)	Image Value CASCADE	Image value (No change) CASCADE	N/A	N/A	N/A	N/A	The sorted image source just passes the real world value on to the uses.
Text	TEXT STRING 43.9	TEXT STRING (No change)	N/A	Search string SV	Used as Low clamp 0	Used as High clamp 200.0	See Note 2. Note the number of characters in the string is held in the LEVEL field. See note on TEXT attribute configuration.

Note 1: In general, the conversion process uses the SH and SL configured at the attribute level if present. If not present, it uses the SH and SL at the entity level.

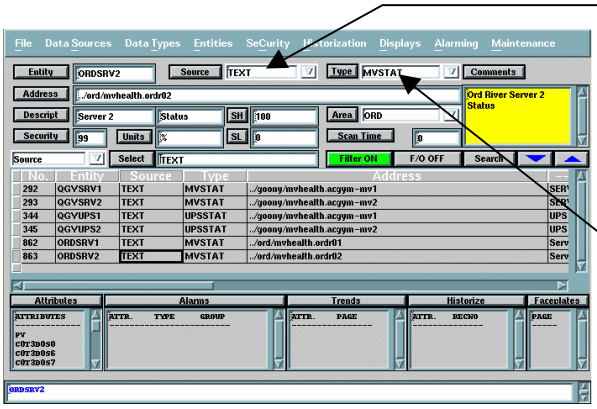
Note 2: The **attribute** SH and SL, if present defines the operator settable limits. If not present, then the **entity** SH and SL are used to define the limits.

Note 3: The format of the value (i.e. Number of decimal places) is defined by the attribute SH if present. If not present, the **entity** SH number of decimal places is used.

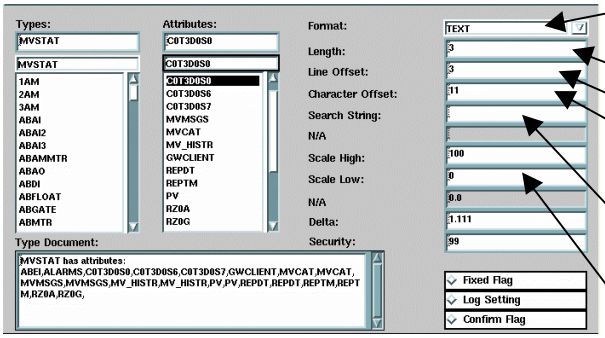
Text Attribute Configuration Example

The diagram shows how the attribute `Format`, `Length`, `Line Offset`, `Search String`, and `Scales` are used to locate the attribute in a ***text*** file

Entity Configuration



Type Configuration



Entity Data Block

The graphic asks for the Space left on disk c0t3s0d0 in server ORD2. This corresponds to entity ORDSRV2, attribute C0T3S0D0.

The entity configuration of ORDSRV2 indicates the source is TEXT.

The type (attribute information) is MVSTAT.

The location of the text file is in the directory "ord", filename "mvhealth.ordr02".

The Type information holds the key to locating and formatting the attribute value.

The **TEXT "driver"** reads the attribute C0T3S0D0 of the type MVSTAT.

The attribute value is 3 characters in **Length** and is situated 3 lines, (**Line Offset** = 3), and 11 characters, (Character Offset = 11), from the start of the file or the start of the **search string** if a **search string** is being used.

There is no **Search String** in this example. The **Search String** may be used to find a certain line in the file.
The **SH and SL** may be used as clamps

Text File Name
../ord/mvhealth.ordr02

```
19/10/95
14:45:01
Processor load (%)
21
c0t3d0s0: 67
c0t3d0s6: 86
c0t3d0s7: 48
```

3 lines below the top of the file (or attribute search string).
11 characters offset to the actual value (Attribute Character Offset)
Length of field is 3 characters (e.g. 0 to 100 percent).

Edit Attribute (Delta, Fixed, Log Setting, Confirm)

These are additional control features that can be configured for each attribute.

Ramp Delta

Enter the percentage amount of Entity scale that the attribute value will increase by when the operator presses the up key (K).
The attribute will be reduced by the same percentage when the down key (L) is pressed.
The fast ramp keys (Control-K and Control-L) work on a fixed multiple of the range key. For example, if the multiple is 10 then all the entities fast ramps will be 10 times the normal ramp figure. You can set this multiple by setting the environment variable FASTRAMP to a numeric scaling value. (See Environment Variables in the Maintenance Chapter).

Fixed Flag

What you type

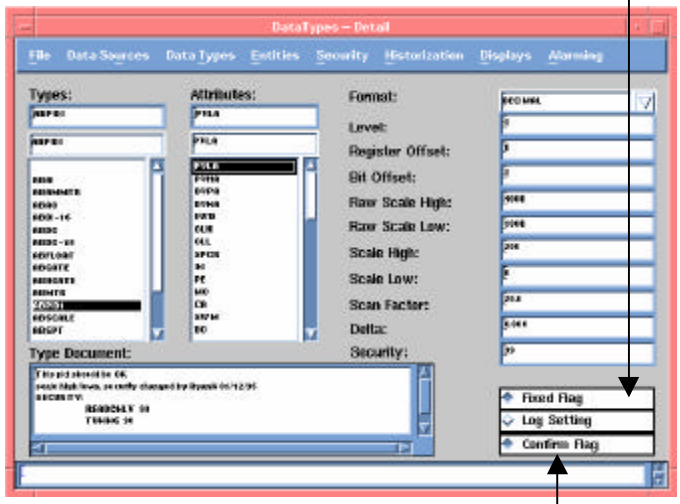
This is used only when a source is set to "simulate". If set, the value will be fixed i.e. it will not continuously vary in simulate mode.
E.g. For a PV, don't set the flag. For a HI alarm do set the flag.

How it Works

This flag can be used to make the simulation a little more realistic. However to make the simulation behave in a truly realistic mode, you should use something like meta scripts. (See the meta script Chapter.)

How to get there

1. ***Data Types:Detail*** This brings up the Types Detail screen.
2. Click on the Type to be modified: The attributes will appear in the central window.
3. Click on the *attribute* to be modified. The attribute items will appear in the detail area.
4. Alternatively, you may add a blank type or attribute using: ***Data Types:Add Blank:Types or Attributes*** and edit the new record.



Log Setting

If you want a log of any changes made on this attribute by an operator, set the Log Setting Flag. If set, a message will be:
1. Printed on the printer.
2. Stored in the historical messages file.

The message stores the operators' user name, the attribute and entity, and the new value of the attribute.

Confirm Flag

What you type

If you are configuring an attribute that has important control consequences, you can specify that the control change must be confirmed before it is actioned.

How it Works

If the confirm flag is set, when the operator makes a change, a message like "Confirm SV change to 90" will appear in the message area.
If the operator presses the Y or Yes key, the value will then be written to the field.

Edit Attribute Screen (Security, Scan Factor)

The directions below describe how to specify the Security characteristics of the attribute. Also, if your driver is a sorted image type, you can specify the scan factor, which is used to determine how often the point is updated.

Security

What you type

Enter the security number of this attribute (Between **0** and **99**).
0 means there are no restrictions to anyone changing the attribute, **99** means only high security personnel can change the attribute.

Example

The attribute security field is one component of a calculation carried out to determine whether this attribute can be changed at a particular moment in time.
The **composite** security number of an attribute is calculated as the higher of the **attribute** security number, the entity security number, and the security number for the area as assigned to that console.

How it Works

If this number is higher than or equal to the **composite** access number (calculated from the lower of the user access number and the console access number) then the user cannot change the value of the attribute. (See the Security Chapter.)

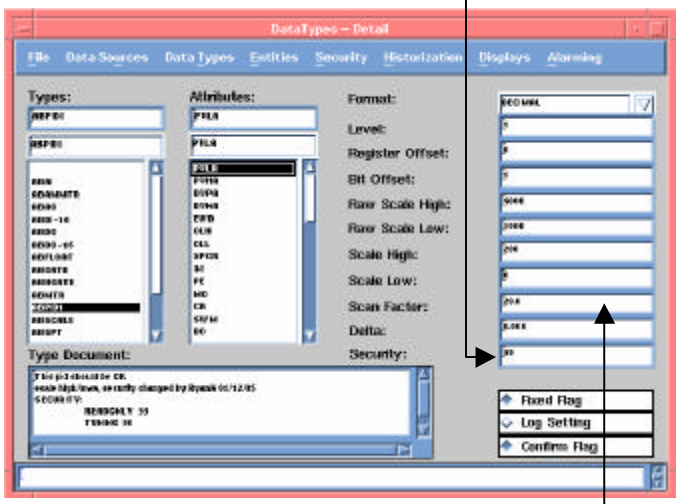
How to get there

1 ***Data Types:Detail*** This brings up the Types Detail screen.

2 Click on the Type to be modified: The attributes will appear in the central window.

3 Click on the *attribute* to be modified. The attribute items will appear in the detail area.

4 Alternatively, you may add a blank type or attribute using: ***Data Types:Add Blank:Types or Attributes*** and edit the new record.



Scan Factor

What you type

The Scan Factor is only used for sorted image sources (e.g. for DCS systems), and not for drivers which perform block transfers of registers such as PLC drivers.
Enter the scan factor, (or scan multiplier) for this attribute. This number is used together with the entity scan time to determine the attribute scan time.

Example

For example, if the attribute PH (Process High Alarm limit) has been set to 10 and for an entity FIC100, the scantime is set to 10 seconds, then the PV of FIC100 will be scanned every 10 seconds and the PH (this attribute) will be scanned every 100 seconds (scan factor x scan time).
Note: See the notes on scan time in the sorted image section of the Sources chapter. Also have a look at the notes on scan time in the Entities chapter.
Important: When used, The Scan Factor should never be 0. (This causes constant scanning.)

8.9 Advanced Notes on Scales

Essentially, the conversion takes place in two stages.

The first stage converts the scaled number from the PLC raw scale to a 0 to 1 scale.

For example: A-B analogues are scaled 0 to 4096. The first stage converts this to a 0 to 1 level i.e.

```
(Raw_Val - RSL)/(RSH - RSL)
```

Then **the second stage** scales the 0 to 1 figure to a "real world" number. The second stage uses either the **attribute** scale low and scale high (if they exist) or the entity scale low and scale high (if the attribute scale low and scale high are blank).

For example, the MV of a PID controller must move between 0 and 100% irrespective of the PV scales -- here we use the **attribute** SH at 100, and the **attribute** scale low as 0. The PV, however, could be for a Temperature Controller which moves between 200 DEGF and 1500 DEGF. Therefore, for the PV, we leave the attribute Scale Low and High blank and allow the **entity** Scale Low and high to dictate the scaling.

In summary, the value is calculated as:

```
Val = (((Raw_Val - RSL)/(RSH-RSL)) * (SH-SL)) +SL
```

where SH and SL are the attribute SH and SL, if present. Otherwise, they are the entity SH and SL.

Note: The number of decimal points is determined by the format you set the Scale High value either in the attribute Scale High, if present, or the entity Scale High. (E.g. If the attribute Scale High is set at 100.00, then an MV would read 34.96.)

8.10 Advanced Notes on Pulse Format

You should only use the pulse format for non-critical applications. (E.g. for silencing a horn). The reasons for this are as follows:

i. The time in milliseconds is not accurate.

ii. The mechanism of creating the pulse involves two separate protocol messages. First a set message and then a reset message.

If for any reason there is an error, subsequent retries of the second message (to reset) may cause the bit to be left on.

Even though the number of retries can be increased, there is always the possibility of the bit being left on.

Critical pulse type operations are therefore best carried out at the lower control level. (E.g. the PLC itself resets a bit to 0.)

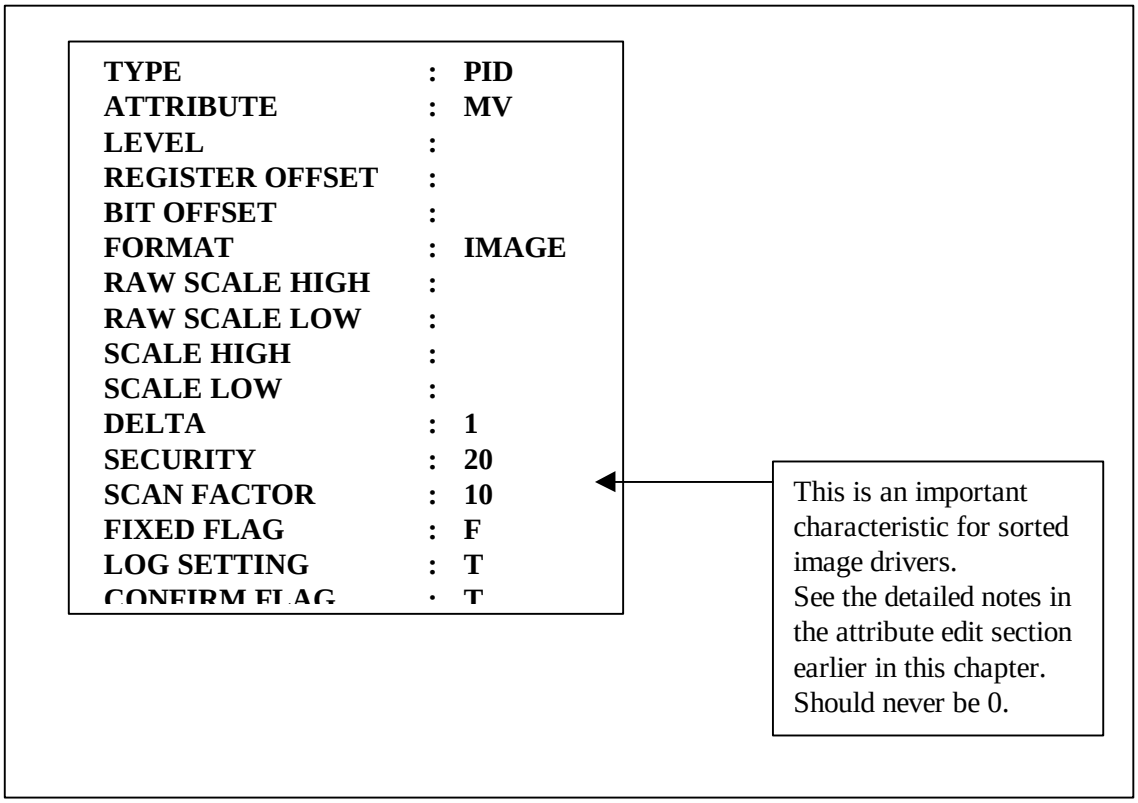
8.11 Sorted Image Based Sources

The configuration of types for sorted image based drivers such as for DCS systems is comparatively simple because a large amount of the location and conversion functions are carried out by the source equipment.

In contrast to a PLC Image source, where you have to specify the register offset and format type information, in a sorted image system this location and conversion process is carried out at the lower level.

You still have to configure the attributes, however, if you are not using the standard types. This is to tell ***MacroView*** such information as which attributes to allow for in the type and what security, logging and ramp functions to use.

A typical sorted image attribute configuration will look like this.



8.12 Checking Out the Type

Once you have configured the type, to check its functionality in the Navigator, you need to configure at least one entity.

i. First, enter the entity name in the scratchpad and call up the detail faceplate by pressing the Detail (D) key. The attribute will come up in the generic faceplate format. (Unless you have already designed a faceplate for it.) Check that each of the attributes match the real world values.

ii. ***Optionally***, call up the group faceplate by pressing the Group (G) key. The attribute will come up in the generic faceplate format. (Unless you have already designed a faceplate for it.) Check that each of the attributes match the real world values.

iii. ***Optionally***, you may now want to design a custom detail faceplate either by modifying an existing faceplate or by designing the faceplate from scratch (See the Graphics Chapter). (You may elect to stay with the generic design to reduce your engineering effort.)

You give the custom detail faceplate the same name as the type and, once converted, the template metafile is stored in the following directory:

For UNIX Systems: `$HOME/detail/<source name>` (e.g. for an A-B source, the faceplate will be stored in the `$HOME/detail/a-b` directory).

For Windows NT Systems: `%HOME%\detail\<source name>` (e.g. for an A-B source, the faceplate will be stored in the `%HOME%\detail\a-b` directory).

You should now be able to see all the attributes of the new type in the custom detail format by pressing the Detail key (once you have the entity name in scratchpad).

iv. ***Optionally***, you may also want to design a group faceplate and check this out as well.