

---

**CREDO-2002-006**

---

# How to Use Parameters in PROforma

---

David Sutton

<b>Creation date</b>	2002
<b>Last modification</b>	06 September 2005
<b>Last edited by</b>	Ayelet Oettinger
<b>Revision</b>	1
<b>Version</b>	1

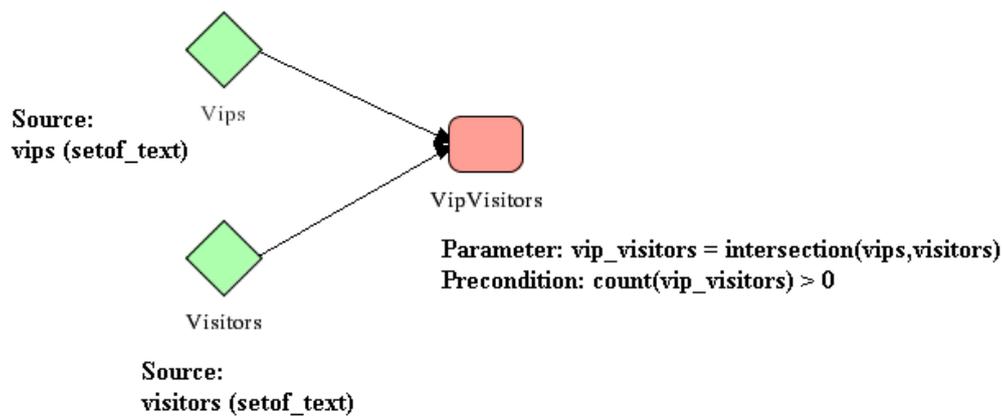
1. **What is a Parameter?** \_\_\_\_\_ 2
2. **What is the Scope of a Parameter?** \_\_\_\_\_ 2
3. **What is the Value of a Parameter?** \_\_\_\_\_ 3
4. **Are Parameters Useful?** \_\_\_\_\_ 3

## 1. What is a Parameter?

A Parameter is essentially a data item that is “local” to a particular task and is assigned a value when that task is started or when its precondition is evaluated.

### Example:

Consider the guideline below.



The enquiries **Vips** and **Visitors** have as sources the two setof\_text data items **vips** and **visitors**. The plan **VipVisitors** has a parameter **vip\_visitors** whose value is the intersection of the sources of the two intersections and whose precondition ensures that it can only start if that intersection is not empty. (Note: the intersection function is not yet implemented, but will be soon).

## 2. What is the Scope of a Parameter?

Certain properties of tasks are defined as expressions or conditions. These properties may refer to the parameters of the task. So for instance the parameter(s) of a task could be referred to in that task's precondition, postcondition, its procedure (if it is an action) and so on.

In addition the parameter(s) of a decision may be referred to in the recommendation rules and arguments associated with that decisions candidates.

The parameter(s) of a plan may also be referred to when assigning values to parameters of that plan's component tasks. So for instance the plan **VipVisitors** might have a subtask **ProcessFirst** with a parameter **first\_vip\_visitor** which was assigned a value by the expression

**first\_vip\_visitor = nth(1,vip\_visitors)**

Where the function **nth** (not yet implemented) returns the nth member of a set.

Note that the expression that assigns a value to a parameter of some task T may refer to the parameters of T's parent plan but may not refer to other parameters of T itself<sup>1</sup>

### 3. What is the Value of a Parameter?

When a parameter appears in an expression its value depends on whether the expression is being evaluated before or after the task has started (i.e. entered the **in\_progress** state).

If an expression is being evaluated after the task has started then any occurrence of a parameter evaluates to the value that the defining expression of the parameter had *at the time the task started*. If the expression is being evaluated at some time *t* before the task has started then occurrences of parameters evaluate to the value of the parameter at that time *t*.

The following example may make this a bit clearer:

Suppose that the following values are entered for the data items **vips** and **visitors**

**vips** = ["Elvis", "Sting", "Prince\_Charles"]

**visitors** = ["Elvis", "Arthur\_Smith", "Joe\_Bloggs"]

When the precondition of the plan **VipVisitors** is evaluated the occurrence of **vip\_visitors** that it contains is evaluated by evaluating its defining expression at that particular moment. In other words we evaluate the precondition as if it had been written as

**count(intersection(vips,visitors)) > 0**

rather than **count(vips\_visitors) > 0**

The precondition clearly is true since **intersection(vips,visitors) = ["Elvis"]**

Any subsequent reference to **vips\_visitors** will still evaluate to **["Elvis"]** even if the values of the data items **vips** or **visitors** have been changed since the task began.

The reason for evaluating parameters this way is that it allows them to be closely linked to preconditions (and to "suspend" conditions when they are implemented).

If a task has started then its precondition must have been true *at the moment that the task started* but is not necessarily true throughout the tasks enactment. The idea is that parameter evaluation will mirror this pattern and allow the enactment of the task to depend on *why* the precondition was true (in this case who the VIP Visitors where).

### 4. Are Parameters Useful?

IMHO parameters are of relatively little use at the moment but may come into their own as the expression language is expanded and "suspend" conditions are implemented.

---

<sup>1</sup> The reason is that such assignments are potentially ambiguous.