

---

## CREDO-2003-025

---

# Style Guide For Authoring *PROforma* Protocols

---

Rory Steele

<b>Creation Date</b>	08 October 2003
<b>Last Modification</b>	09 October 2003
<b>Revision</b>	9
<b>Circulation</b>	Unrestricted
<b>Status</b>	Draft

<b>1. Rationale</b>	<b>2</b>
<b>2. General style rules</b>	<b>2</b>
2.1 Use descriptive names	2
2.2 Acronyms	2
2.3 Text-based data definitions and quoting	2
2.4 Boolean data definitions	2
<b>3. Task names</b>	<b>3</b>
3.1 Append task type	3
3.2 Use UpperCamelCase	3
<b>4. Candidate names</b>	<b>3</b>
4.1 Use verb-noun pattern	3
4.2 Use UpperCamelCase	3
<b>5. Data definition names</b>	<b>3</b>
5.1 Use lowerCamelCase	3

## 1. Rationale

The rationale for a style guide for authoring *PROforma* protocols is to facilitate interoperability of guidelines. A consistent naming scheme also enhances the readability of the guideline and provides the author with semantic clues during authoring.

## 2. General style rules

### 2.1 Use descriptive names

Obviously, the more descriptive the name, the easier it is to understand the original intention of the author concerning the usage of that component within the guideline. Calling a task 'Request' provides little information to a future author wishing understand whether they would wish to incorporate that task within their guideline. Calling the task 'BloodPressureEnquiry' is much more informative.

### 2.2 Acronyms

When using acronyms within a name, capitalise *only* the first letter of the acronym to enhance readability. For example, use 'SendMessage' rather than 'SendXMLMessage'.

### 2.3 Text-based data definitions and quoting

When using text-based data items within an expression, and the expression is a comparison of the textual value of that data item, always quote the text that the data item is being compared against.

For example, use expressions of the form: `dataValue = "yes"`

rather than: `dataValue = yes`

### 2.4 Boolean data definitions

Never use boolean-based data definitions. **ALWAYS** use a text-based data definition and explicitly add the two range values for that definition. In a future release, boolean types will actually refer to 'true' Booleans rather than a bivalent text definition.

**NB** boolean data definitions actually are only a bivalent text definition, with the addition facility to use them as pseudo-booleans within a *PROforma* expression. Explicitly performing the text comparison within the expression greatly clarifies the information the expression is attempting to capture.

## 3. Task names

### 3.1 Append task type

Appending the task type to the name provides an obvious cue as to the task's function. For example, the task 'SendFax' could be describing the decision whether to send a fax, an enquiry whether the user wishes to send a fax or an action to actually schedule the sending of the fax. All ambiguity is removed by appending the correct task suffix to the task name ('SendFaxAction', 'SendFaxDecision', 'SendFaxEnquiry').

### 3.2 Use UpperCamelCase

Use upper camel-case to name tasks. This enhances the readability of the name by separating naming cues with capital letters, whilst keeping the name length to a minimum (as opposed to using the underscore character). Examples include 'BloodResultsEnquiry' and 'ConfirmAppointmentByFaxAction'.

## 4. Candidate names

### 4.1 Use verb-noun pattern

Use a consistent verb-noun pattern to capture the notion of choice between the candidates of a decision task.

Examples include 'GoRight', 'GoLeft', 'GoStraightOn' and 'GoBackwards'.

### 4.2 Use UpperCamelCase

Use upper camel-case to name candidates. This enhances the readability of the name by separating naming cues with capital letters, whilst keeping the name length to a minimum (as opposed to using the underscore character).

Examples include 'ReferPatient', 'ManageInPractice' and 'AdministerChemo'.

## 5. Data definition names

### 5.1 Use lowerCamelCase

Use lower camel-case for dataitem names. This enhances the readability of the name by separating naming cues with capital letters, whilst keeping the name length to a minimum (as opposed to using the underscore character). Also, using lower camel-case differentiates data definitions from task and candidate names.

Examples include 'bloodPressure', 'gender' and 'siblingNumber'.