Settings

If you want to use settings in your model, we recommend to **always** use one or more classes and mark the attributes as settings. It is not recommended to use the inline style definition of settings within action script.

This approach has the following advantages:

- It is easy to find all settings of a service.
- The same settings can be re-used in several diagrams.

Only exception of this rule: Use the inline style definition when having a single setting in a specific context.

On this Page:

Related Pages:

- Project Organization
- Naming Conventions and Containment Tree Organisation
- Model Documentation
- Settings
- Mappings
- Sub-activities
- Logging
- Error Handling

- Attribute Specification
- setting() Macro
- xUML Service Settings

Naming Conventions for Settings

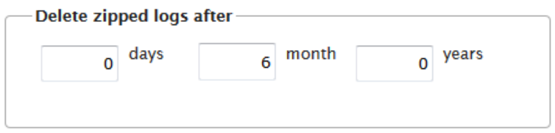
For	Use	Example
Single Settings	one class named **Settings**	
Multiple Settings	multiple classes: **SettingsSuffix**	SettingsCustomer
Documentation	use tagged value **Setting Name** to document how to use this setting.	Put in timeout in seconds:

Setting vs. User Interface

Usage of a settings class is to be preferred, if the setting is a simple one, e.g "Customer:".

In place of settings, you could also implement a user interface. The usage of a user interface is preferable, in the following cases:

- The settings need enhanced validation (enumeration etc.).
- The settings are complex, e.g. three input fields which belong together.



Delete zipped logs after

0 days 6 month 0 years

If you want to use a user interface to change the settings, they should be adjustable without service stop.