Gargantua 8 form design
best practices

Table of contents

Attribute design

Attributes are the first brick in the design of your EDM database; choosing the proper attributes and their properties is essential for later improvements. Attributes, besides the master role in organising data, play an important role in other EDM concepts such as statistics or views.

Workflow processes also make use of attributes for controlling the flow of a process.

Naming conventions

Attribute names have few constraints, beside the fact that they have to be unique.

However, for easy understanding of existing structures, naming conventions play an important role. There is actually no standardized naming convention; this document describes one such possible convention.

For instance, one naming convention may require the attribute name to follow these rules

All caps names

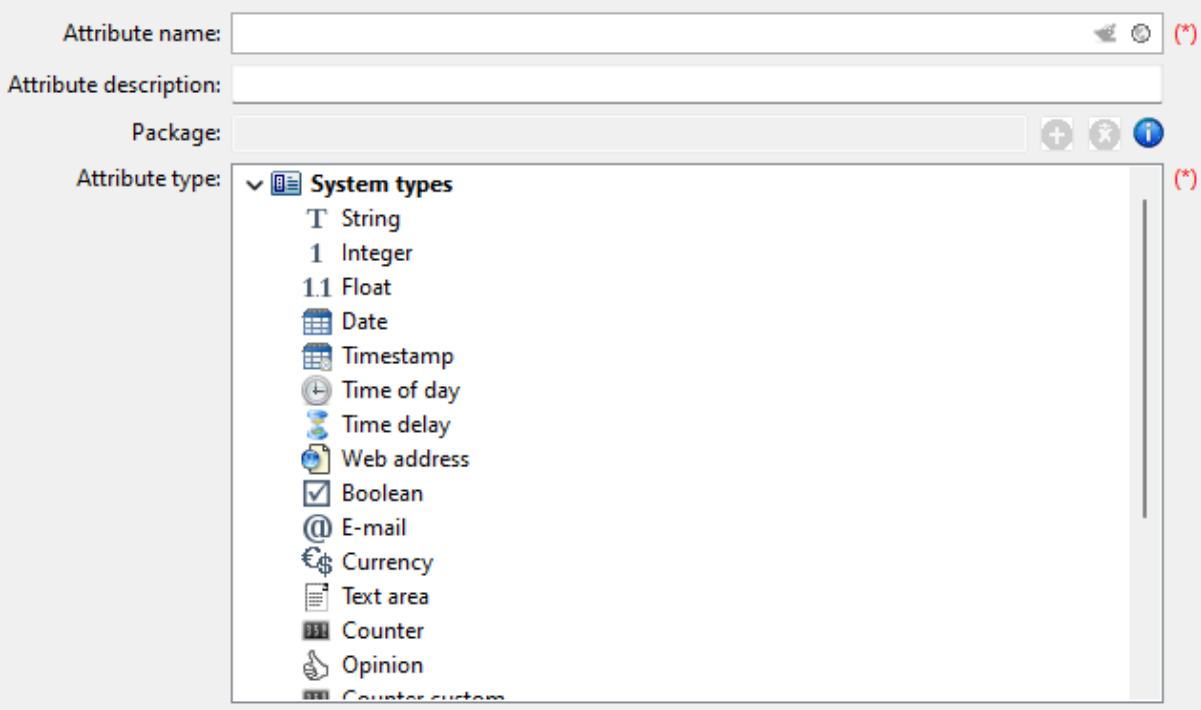
Use only capital letters when naming your attributes

Limited symbol set

Besides letters, use only a reduced symbol set; for instance use . (dot / full stop) for separating elements and _ (underscore) for separating words when required

Naming format

Use a consistent naming scheme throughout all your attributes; for instance, `<COMPONENT>.<TYPE>.<ATTRIBUTE NAME>` is a practical naming scheme.



EXAMPLES

Here are a few examples that follow the naming conventions stated in the previous section

CRR.ALP.ADDRESS

An alphanumeric attribute that will hold a street address for a mailing database

Breakdown and explanation:

Based on the naming format `<COMPONENT>.<TYPE>.<ATTRIBUTE NAME>`

CRR

The component; `CRR` as abbreviation for 'Courrier'; usually the component part of the naming scheme has exactly 3 letters

ALP

`ALP` is the abbreviation for 'alphanumeric'; also possible abbreviations are `DAT` for 'Date', `BLN` for 'Boolean' and so on.

ADDRESS

The effective attribute name; here we need an address, so we name our attribute `ADDRESS`

Here are a few more examples :

CRR.ALP.FIRST_NAME

An alphanumeric attribute that will hold the first name of a person for a mailing database

CRR.DAT.REGISTRATION

A date attribute that will hold the registration date of a document for a mailing database

Forms are used for several purposes

- they define the collection of attributes that are set on a specific object (document, folder, process, etc.)
- they define some attribute properties that will have an impact on the user input
- they define the layout of the page that is used for data visualisation and input
- they provide means for user interaction

This section provides some best practices in the context of the Gargantua application that can be used when designing forms.

Naming conventions

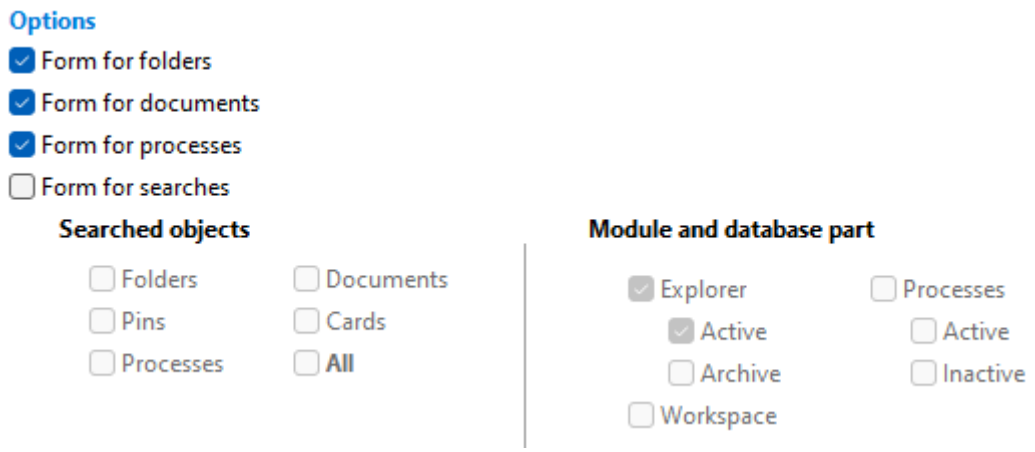
As well as for attributes, form names have few constraints, beside the fact that they have to be unique.

However, for easy understanding of existing structures, naming conventions play an important role. There is actually no standardized naming convention; this document describes one such possible convention.

For instance, one naming convention may require the form name to follow these rules

Usage prefix

Use GED or WF as a prefix to easily identify forms designed for the document management from workflow forms; generally, the workflow forms are more complex due to a richer interaction with users



Separate popup forms

Some forms may be displayed as a popup fron another form; is it not a bad idea to separate these forms from the general purpose ones; it is also possible to contain the popup pages within a multi-page form, too.

Naming format

Use a consistent naming scheme throughout all your forms; for instance, `<USAGE> - <FORM NAME>` is a practical naming scheme.

Examples

Here are a few examples that follow the naming conventions stated in the previous section

WF - Dashboard

A form used for dashboard purposes in a workflow process

Breakdown and explanation:

Based on the naming format `<USAGE> - <FORM NAME>`

WF

This form is designed for a workflow process; other possible usage is GED for document management.

Dashboard

The effective form name; here we need an dashboard, so we name our form `Dashboard`

Here are a few more examples :

GED - Generic folder

A form used in the document management for a general folder

WF - POPUP

A form used for popup display in workflow processes

Form scripts

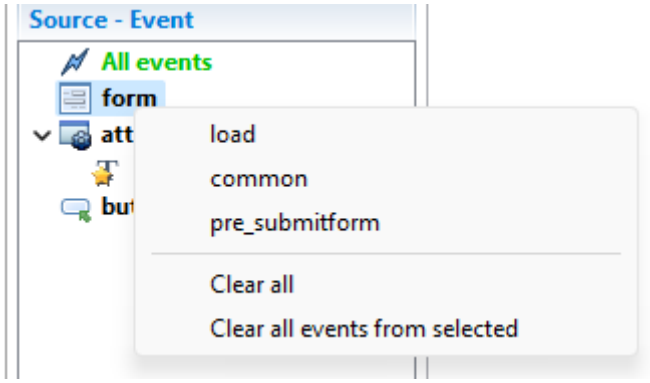
Scripts play an important role in customizing form behavior and extend standard behavior. The main goal is to provide better user experience and improved ergonomy for the user.

Embedded scripts

Embedded scrippts are scripts that are defined with the form itself; there are two types of scripts

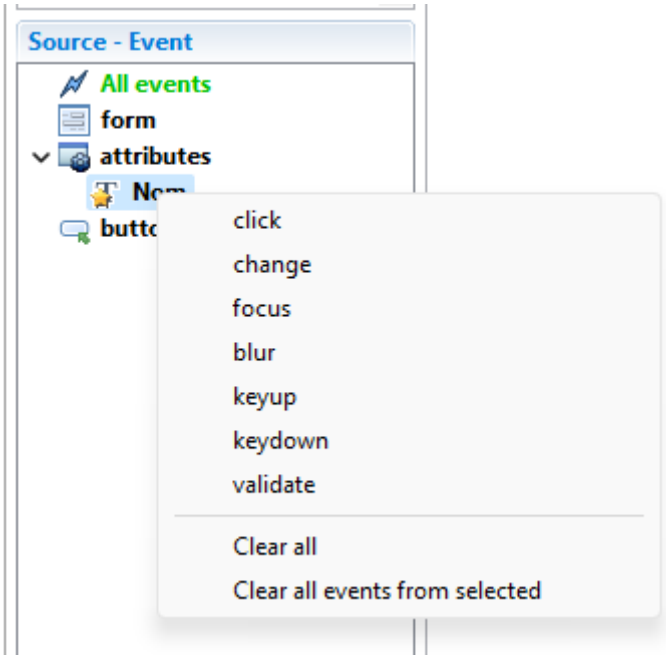
general scripts

these scripts are included as they are written in the form



event scripts

these scripts are wrapped in a proper event function that is attached to an element



The main difference between the two types above is that the event scripts are always internally wrapped in a function which is automatically bound to the event.

For instance, for an event script one would only write something like this :

```
var x = gfxVal('myAttribute');
```

While the general scripts (the 'common' script) would contain something like this :

```
function myFunction(myParameter) {
    var x = gfxVal('myAttribute');
}
```

Included scripts

Included scripts are scripts defined in the Scripts branch of the Admnistrator tool; these are standalone scripts thay may contain common behavior, and thus can be included in different forms.

These scripts are also included as they are written in the form itself.

To be available for inclusion, the script must be defined as a form script.

Embedded scripts vs included scripts

Common scenarios

Conditional mandatory attributes

Form extensions

3rd party web services

Retrieving data from Gargantua

Customizing controls

Customizing appearance

Script design

Naming conventions

Script aggregation