

Extension of: Kuhn et al. (1999a), S. 79

3. CLASS ATTRIBUTE & ATTRIBUTE TYPES

Basics: Definition of Attributes



- ▶ Attributes for classes and relation classes have to be defined in the definition section of the class/relation class with 'TYPE'.
- ▶ The following attribute types are possible:
 - ▶ INTEGER integer
 - ▶ DOUBLE floating number
 - ▶ STRING string – max. 3699 symbols
 - ▶ LONGSTRING string – max. 32000 symbols
 - ▶ TIME time
 - ▶ DATE date
 - ▶ DATETIME date and time
 - ▶ ENUMERATION enumeration for selecting a characteristic
 - ▶ ENUMERATIONLIST enumeration for selecting one or several characteristics
 - ▶ DISTRIBUTION statistical distribution
 - ▶ PROGRAMCALL enumeration for selecting a program
 - ▶ RECORD a table of attributes
 - ▶ EXPRESSION a formula
 - ▶ INTERREF reference on a model or an instance
 - ▶ ATTRPROFREF a preset set of attribute values

Attribute Types and their Appearance



Numerical Attributes: **Integer (INTEGER)**

1_Integer:

- ▶ An attribute of the type "Integer" is defined as an integer from -1,999,999,999 to 1,999,999,999.
- ▶ An ADOxx integer is limited to 10 digits plus an optional sign ('+' or '-')
- ▶ The standard value of attributes of this type is "0" or a value defined

Attribute Types and their appearance



Numerical Attributes: **Floating number (DOUBLE)**

2_Double:

0.000000

- ▶ The amount of decimal places is defined by the attribute definition
- ▶ An attribute of the type "Double" is defined for a float within +/-999,999,999,999,999 for an integer (without decimal places) or +/-999,999,999.999999 for figures with 6 decimals.
- ▶ The corresponding attribute value is displayed to 6 decimal places. That means that a double value should not exceed a total of 15 significant digits with at last 6 decimal digits!
- ▶ The standard value of attributes of this type is "0.000000" or a value defined in the application library.

Attribute Types and their Appearance



String attributes: **String (STRING)**

3_String:

- ▶ An attribute of the type "String" is defined for texts up to 3700 characters of any type.
 - ▶ Hint: The maximum number of characters is 250 for name. That concerns classes, relation, instances, attributes, application models, libraries and application libraries.
 - ▶ Model names have a special rule!
- ▶ The standard value of attributes of this type is "" (no entry) or a value defined in the application library.

Attribute Types and their Appearance



String attributes: **Longstring (LONGSTRING):**

- ▶ Some text attributes are already defined as „multi-line“. The parameter lines can be used to specify how many lines should be shown in the text field of the Notebook.
- ▶ The parameter dialog can be used to specify special input supports in place of the standard one.
- ▶ An attribute of type "Longstring" is defined for texts up to 32000 characters of any type.
- ▶ The standard value of attributes of this type is "" (no entry) or a value defined in the application library.

Attribute Types and their Appearance



Enumerations / Enumeration lists: **Enumeration (ENUMERATION)**

The screenshot shows a software interface with three different visual representations of an enumeration type named `8_Enumeration`:

- Radio buttons:** A group box titled `8_Enumeration` containing three radio buttons labeled `value-1`, `value-2`, and `value-3`. The `value-1` radio button is selected.
- Checkbox:** A single checkbox labeled `8_Enumeration`.
- Dropdown list:** A label `8_Enumeration:` followed by a dropdown menu. The dropdown is open, showing a list with three items: `value-1`, `value-2`, and `value-3`. The `value-1` item is currently selected and highlighted in blue.

- ▶ The parameter `ctrltype` sets how the enumeration should appear, as a drop down list, as radiobuttons or as checkboxes (only if two possible values).
- ▶ An attribute of the type "Enumeration" is characterised by a defined set of values. An "Enumeration" attribute has exactly one value of this set.
- ▶ The standard value of this type is specified in the library definition.

Attribute Types and their Appearance



Enumerations / Enumeration lists: **Enumeration list(ENUMERATIONLIST):**



- ▶ An attribute of the type "Enumeration list" is characterised by a defined set of values. An "Enumeration list" attribute has either none, one or several values of this set. The difference to an "Enumeration" attribute is, that an "Enumeration list" attribute can have more than one entry selected!
- ▶ The standard value of this type must specified in the library definition.

Attribute Types and their Appearance



Date / Time: **Date (DATE)**

5_Time:

The ADOxx format for date is YYYY:MM:DD

Date / Time: **Time (TIME)**

6_Date:

The ADOxx format time is YY:MM:DDD:HH:MM:SS

Date / Time: **Date and Time (DATETIME)**

7_DateTime:

The ADOxx format time is YYYY:MM:DD HH:MM:SS

- ▶ Time format YY:DDD:HH:MM:SS (years:days:hours:minutes:seconds). Valid day ranges are from 0 to 365, valid hours are between 0 and 23, valid minutes and valid seconds are between 0 and 59.
- ▶ The standard value of attributes of this type is "00:000:00:00:00" or a value defined in the application library.

Attribute Types and their Appearance



References / Program calls: **Intermodel reference (INTERREF)**



Syntax of the InterRef domain definition facet:

<i>InterRefDomain</i>	:	[REFDOMAIN [max:intValue]] <i>Refs</i> .
<i>Refs</i>	:	<i>ModRefs</i> <i>InstRefs</i> .
<i>ModRefs</i>	:	{ MODREF mt:modelTypeName } .
<i>ObjRefs</i>	:	{ OBJREF mt:modelTypeName c:className max:intValue } .

Syntax of InterRef attribute values:

<i>InterRefValue</i>	:	<i>ModRefs</i> <i>ObjRefs</i> .
<i>ModRef</i>	:	{ REF m:modelName mt:modelTypeName } .
<i>ObjRef</i>	:	{ REF m:modelName mt:modelTypeName c:className i:instanceName } .

Attribute Types and their Appearance



References / Program calls: **Programcall (PROGRAMCALL)**

- ▶ A PROGRAMCALL attribute is characterized by a fixed set of items. These items are related to AdoScripts which can be called via the user interface. A PROGRAMCALL attribute value consists of at most one of the defined items and an optional parameter.

10_Program Call

Executable: →

<automatically> ▼

Program arguments: 🔍

"C:\Programme\BOC\ADOxx 1.3\areena.exe"

<i>ProgramCallDomain</i>	:	{ <i>ItemDefinition</i> } .
<i>ItemDefinition</i>	:	ITEM <i>itemText</i> [<i>ParameterDefinition</i>] { <i>FdlgFilter</i> } <i>AdoScript</i> .
<i>ParameterDefinition</i>	:	param : <i>paramText</i> [: <i>defaultTextValue</i>] .
<i>FdlgFilter</i>	:	fdlg-filter <i> : <i>filterText</i> fdlg-type <i> : <i>filterDescriptionText</i> .

itemText, *paramText*, *defaultTextValue*, *filterText* and *filterDescriptionText* are string values.

Attribute Types and their Appearance



Table: **Table (TABLE)**

Tables will appear in Notebooks according to the definition of the table class.

Following adjustments can be done in AttrRep of the table class:

- ▶ which columns should be shown
- ▶ in what sequence
- ▶ Relative width - Parameter `width`

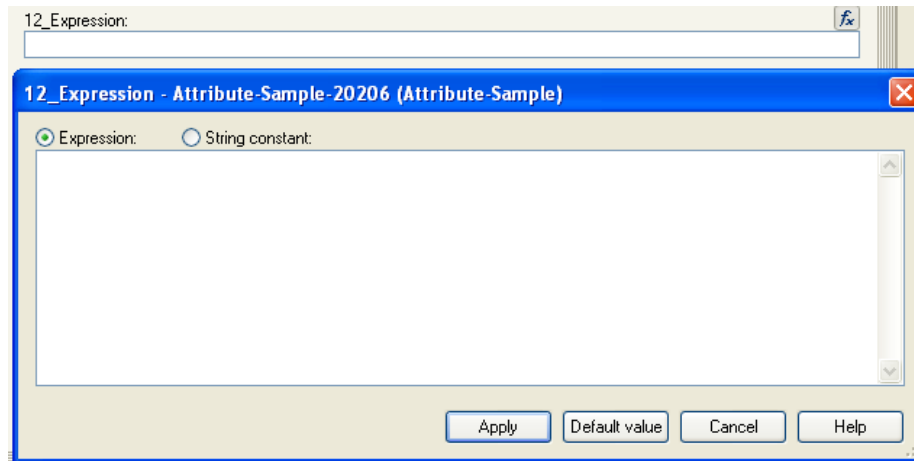
An Attribute of Type "Table" (RECORD) is defined by a flexible List-/Table-Administration of Attribute Types that are put together.

The standard Value for Attributes of this Type depends on the Attribute Types defined in the Table Class.

Attribute Types and their Appearance



Expressions / Attribute profile references: Expression(EXPRESSION)



- ▶ Every definition of expression attributes is started with the keyword `EXPR`. The result type is defined with the attribute `type` and the default formula is defined with the attribute `expr`. Every time you create an instance (a model, object, or connector), this formula will be used to compute the result value of the expression.
- ▶ By setting the modifier `fixed`, you make the expression attribute a fixed expression. The user will not be able to change the formula in the Modelling Toolkit.
- ▶ The formula itself (defined in the attribute `expr`) must never be longer than 3600 characters.
- ▶ For expressions with result type `double`, the attribute `format` can be used to specify the number of digits that should be displayed on the user interface. Note: the number of digits displayed on the user interface do not affect the internal precision of the expression result value.

Attribute Definition



<i>attribute-definition :</i>	<i>instanceattribute-definition classattribute-definition .</i>
<i>classattribute-definition :</i>	CLASSATTRIBUTE <i>identifier</i> TYPE <i>typeidentifier</i> CLASSATTRIBUTE <i>identifier</i> TYPE <i>typeidentifier</i> VALUE <i>val</i> CLASSATTRIBUTE <i>identifier</i> VALUE <i>val</i> CLASSATTRIBUTE <i>identifier</i> TYPE RECORD .
<i>instanceattribute-definition :</i>	ATTRIBUTE <i>identifier</i> TYPE <i>typeidentifier</i> ATTRIBUTE <i>identifier</i> TYPE <i>typeidentifier</i> VALUE <i>val</i> ATTRIBUTE <i>identifier</i> VALUE <i>val</i> ATTRIBUTE <i>identifier</i> TYPE RECORD .
<i>typeidentifier :</i>	INTEGER DOUBLE STRING DISTRIBUTION TIME ENUMERATION ENUMERATIONLIST PROGRAMCALL INTERREF EXPRESSION ATTRPROFREF .