

Tag	Description	Attributes (bold parameters are required, bold values are default)	Example(s)
object	manipulates a single object. Query must return at most 1 record. If no object matches countVar=maxCountVar=0	<b>from</b> ="OQL list of types with aliases" <b>where</b> ="strict OQL criteria, \$attributes" <b>countVar</b> ="newJavaVariableName" <b>maxCountVar</b> ="newJavaVariableName" <b>db</b> ="nameOfDBconfigFile"	<mak:object from="best.Student bs, bs.person p" where="bs=\$student" maxCountVar="foundIt"> ...JSP... <mak:value expr="p.name"/> </mak:object> <%if(foundIt.intValue()==0){%>No student found<%}>
list	Iterates over a series of matching objects	<b>from</b> ="OQL list of types with aliases" <b>where</b> ="OQL criteria, \$attributes" <b>orderBy</b> ="list of OQL fields, \$attributes" <b>groupBy</b> ="list of OQL fields, \$attributes" <b>countVar</b> ="newJavaVariableName" <b>maxCountVar</b> ="newJavaVariableName" <b>separator</b> ="string <%=expr%>" <b>db</b> ="nameOfDBconfigFile"	<mak:list from="best.Student bs, bs.person p" where="p.gender=1" orderBy="p.name, p.surname" groupBy="bs.lbg" countVar="count" maxCountVar="numberOfGirls" separator="," db="localhost_mysql_makumbadb"> ...JSP...<mak:value expr="p.name"/> </mak:list>
value	Computes a value, then formats it and displays it or puts into a java variable	<b>expr</b> ="OQL expression <%=expr%>" <b>format</b> ="format String <%=expr%>" <b>urlEncode</b> ="true false <%=expr%>" <b>html</b> ="auto true false <%=expr%>" <b>lineSeparator</b> ="<p><p> string <%=expr%>" <b>longLineLength</b> ="30 integer <%=intExpr%>" <b>var</b> ="newJavaVariableName" <b>printVar</b> ="newJavaStringVariableName" <b>display</b> ="true false"	<mak:value expr="s.person.name"/> <mak:value expr="count(s)"/>  Text formatter checks whether there are html tags inside the text and formats it accordingly. If longest line is longer than longLineLength it is formatted as <p>...</p> (or whatever the lineSeparator), otherwise (short lines) as <pre> This can be overridden with html parameter  For format parameters see: <a href="#">java.text.SimpleDateFormat</a>
attribute	Finds and manipulates an attribute	<b>name</b> ="attributeName" <b>var</b> ="newJavaVariableName" <b>exceptionVar</b> ="newJavaVariableName" <b>db</b> ="nameOfDBconfigFile <%=expr%>"	<mak:attribute name="student"/>
form	A generic form (action must, method, name & message can be used on other form types)	<b>action</b> ="actionPage.jsp <%=expr%>" <b>handler</b> ="javaMethodName" <b>method</b> ="GET POST <%=expr%>" <b>name</b> ="formName" <b>message</b> ="message text"	<mak:form handler="newAccountCheck" method="post"> <input name="pass2" type="password"><input type="submit"> </mak:form>
newForm	Form for creating new object	<b>type</b> ="OQL type" <b>db</b> ="nameOfDBconfigFile" +[ <b>action</b> , method, name, message]	<mak:newForm type="best.Student" action="studentList.jsp"> <mak:input field="person.name"/><input type="submit"> </mak:newForm>
editForm	Form for editing object	<b>object</b> ="OQL reference" +[ <b>action</b> , method, name, message]	<mak:editForm object="s" action="studentList.jsp"> <mak:input field="person.name"/><input type="submit"> </mak:editForm>
addForm	Creates new element for internal set	<b>object</b> ="OQL reference" <b>field</b> ="fieldname" +[ <b>action</b> , method, name, message]	
deleteLink	Creates a link to a regular makumba page that deletes the object	<b>object</b> ="OQL reference" <b>action</b> ="actionPage.jsp <%=expr%>" <b>message</b> ="message text"	<mak:deleteLink object="s" action="welcome.jsp" message="Student deleted"> Delete this Student</mak:deleteLink>
input	Field input control for forms	<b>field</b> ="fieldname <%=expr%>" <b>name</b> ="controlName <%=expr%>" <b>value</b> ="value \$attribute" <b>dataType</b> ="char text ptr int" <b>display</b> ="true false" <b>format</b> ="format" (See: <a href="#">java.text.SimpleDateFormat</a> ) <b>type</b> ="type" <b>size</b> , <b>maxLength</b> , <b>rows</b> , <b>cols</b> ="int <%=expr%>"	<mak:input field="name"/> <mak:input field="citizenship"/> <mak:input field="birthdate" format="d' of 'MMMM', year 'yyyy"/>
response	Prints form messages (in action page)	none	<mak:response/>
login	Form takes care of passing the existing arguments	none, but form should contain controls named 'username' and 'password'	<mak:login> <input type="text" name="user"> <input type="password" name="pass"> <input type="submit"> </mak:login>
logout	Removes a secret session attribute	<b>actor</b> ="name <%=expr%>"	<mak:logout actor="member"/>
version	Prints a Makumba version	none	<mak:version/>

Format:

fieldName= [attributes] fieldType [;Description or friendly field name]

Type	Description	Example
char char[x]	Character field (one line) ...length limited to x (1 to 255) characters Character enumerator (chooser interface)	title=char name=char[30] ;Name title=char{"Mr", "Mrs", "Miss", "Dr"} ;Title
text	A (long) text field, can contain binary data	comments=text ;Comments
ptr	Pointer to any object (of any type, even self)	citizenship=ptr general.Country ;Citizen of mother=ptr general.Person
int	Integer value Integer enumerator shows user friendly strings in interface, but stores integers in the database	height=int ;Height [cm] tShirtSize=int{"M"}=0, "L"}=1, "XL"}=2, "XXL"}=8, "S"}=3, "XS"}=4}
set	Internal set	addresses=set ;Addresses addresses->streetNameAndNumber=char[200] ;Street address addresses->zipCode=int ;ZIP addresses->city=char[100] ;City addresses->country=ptr general.Country ;Country addresses->description=text ;Description citizenships=set ptr general.Country ;Citizenships

Data type attributes	Description	Example
not null	Makumba new/edit form responders check before update and rejects null values	in general.Person: name=not null char[30] ;First name citizenship=not null ptr general.Country ;Citizen of
fixed	The value can't change (field edit controls are rendered as read only, edit form responder checks value before saving) Commonly combined with 'not null' for fixed object hierarchy	in general.Person: mother=fixed general.Person  in best.Student: person=fixed not null ptr general.Person

Data type	Default user interface	Relational database type (orientative)	Java type (orientative)
int	Text field <i>or slider</i>	Integer	<a href="#">java.lang.Integer</a>
int{}	Simple choice <i>or radio</i>		
char[]	Text field	Limited char	<a href="#">java.lang.String</a>
char{}	Simple choice <i>or radio</i>		
text	Text area <i>or upload</i>	Unlimited byte	<a href="#">org.makumba.Text</a>
date	Date	Date and time	<a href="#">java.util.Date</a>
ptr	Chooser <i>or radio</i> of titles or pointer	Preferably a long	<a href="#">org.makumba.Pointer</a>
xx=ptr xx->...	The record, if any	Preferably a long + table	
Index field	(Nothing)	Preferably long, auto-increment, or other unique	
modification date	Date, not editable, default field names: TS_create and TS_modify	Timestamp	<a href="#">java.sql.Timestamp</a>
set	Multiple choice of record titles	Table with ptrRel to the parent table and ptr to the foreign table	<a href="#">java.util.Vector</a> of <a href="#">org.makumba.Pointer</a>
xx=set xx-> ...	Like a table, under current record	Table with ptrRel to the parent table, and the rest of the fields	
set int{ ...}	Multiple choice from values	Table with ptrRel to the parent table, and an int field	<a href="#">java.util.Vector</a> of <a href="#">java.lang.Integer</a>
set char{ ...}	Multiple choice from values	Table with ptrRel to the parent table, and an int field	<a href="#">java.util.Vector</a> of <a href="#">java.lang.String</a>
Type (table)	<i>List of records, with add, delete, edit controls; titles emphasized</i>	Table	<a href="#">java.util.Vector</a> of <a href="#">java.util.Dictionary</a>
Record	<i>All fields, as shown above</i>	Record	<a href="#">java.util.Dictionary</a>

Note: Options in italics are not implemented yet

**Business Logic discovery**e.g. with the page path `/student/mycv/indexPersonal.jsp`, the following criteria are applied, in order:

criterion	Java classes checked for
page name	StudentMycvIndexPersonalLogic
caps parts of page/directory name	StudentMycvIndexLogic
directory name	StudentMycvLogic
ent directory name(s)	StudentLogic Logic