KET specification for macromolecules

This specification defines only changes to KET format specification supporting macromolecules (big molecules). Full specification can be found <HERE>

The root structure of the KET file in terms of macromolecules support.

Macromolecules and their components are represented in KET format as a list of nodes.

- root
    - nodes
        - $ref: "monomer<N>"
        - $ref: "group<M>"
    - connections
        - array of connections
    - templates
        - $ref: "monomerTemplate-<id>"

The word "Template" in monomerTemplate and groupTemplate emphasizes that the objects can be utilized repeatedly in macromolecules construction.

Value	Type	Value constraints	Usage	Description
id for monomerTemplate	string	1. id is unique for rest of the file 2. Identifiers for inline monomers starts from '#' symbol.	$ref: "monomerTemplate-Ala" $ref: "monomerTemplate-#1" $ref: "monomerTemplate-any_user_utf8_string"	Use corresponding 3-letter designations for standard monomers, inline monomers start from "#".
N, M	integer	>=0, unique number of monomer, group or connection	$ref: "monomer0" $ref: "group1"	

Monomers' template definition in KET

In KET, a monomer can be defined outside the root block using the reference ***monomerTemplate-<id>*** or straight as element of nodes array inside the root.

There are following fields in a monomerTemplate object definition.

Field	Description	Required	Type	Value constraints	Usage	Comments
type	object type	yes	string	the value is always "monomerTemplate"	"type": "monomerTemplate"	
monomerClass	monomer class. can be unspecified.	no	string	can be one of the following enum values: AminoAcid, Sugar,Phosphate, Base, Terminator, Linker, Unknown, CHEM.	"monomerClass": "AminoAcid"	
id	monomer identifier	yes	string	any non empty unique string among monomers defined in the KET	"label": "Ala" "label": "#1"	• for the standard monomers corresponding 3-letter designations • inline monomers starts from '#' symbol. • for all other cases it should be non empty unique string among other monomers defined in the current KET.

fullName	monomer full name	no	string	any non empty string	"fullName": "alanine"	
alias	monomer display alias. If not specified, the value of **id** should be used instead.	no	string	any non empty string	"symbol": "A" "symbol": "meA"	• It is recommended to use corresponding 1-letter designations for standard monomers. • As this value is used for labeling the monomer icon, it should be as short as possible.
naturalAnalog	specified for monomers that are modified versions of standard ones. refers to unmodified monomer identifier. for a standard monomer the value matches with the **label**.	no	string	■ Ala,Arg,Asn,Asp, Cys, Gln,Glu,Gly,His,Ile, Leu,Lys,Met,Phe, Pro, Ser,Thr,Trp,Tyr,Val ■ Pi ■ Rib, dRib ■ Ade,Cyt,Gua,Thy, Ura ■ if not defined - the monomer has no natural analog	"naturalAnalog": "Ala"	In the current conventions, the value can only be a three-letter identifier for a standard monomer. However, when expanding the classes of monomers, longer identifiers can be used.
attachmentPoints	array of attachment points of the monomer	no	array of objects		"attachmentPoints":[{ "type": "right", "label": "Br", "attachmentAtom": 0, "leavingGroup : [{ "atoms": [9 ,10] }] }]	
■ **type**	attachment point type	no	string	the optional type which can be one of: "left", "right", "side". if the field not specified there are following defaults: **attachmentPoints[0]** → left **attachmentPoints[1]** → right **attachmentPoints[2..n]** → side		

- **label**	attachment point label	no	string	if defined, the value should be unique in the attachmentPoints array, if the value it's not defined the defaults are: **type =** left → **label = "**R1"**,** **type =** right → **label = "**R2", first attachment point of a "side" type appeared in attachmentPoints array → **label** = "R3", second, → **label = "**R4"**,** etc.	"label" : "R2"	
- **attachmentAtom**	index of an attachment atom in the atoms array of the monomer	yes	integer	>=0	"attachmentAtom" : 0	

- **leavingGroup**		no	object		"leavingGroup : {
					"atoms":[9 ,10]
					}	

○ **atoms**	array of indexes of the atoms to be removed when the corresponding attachment point is in use.	yes	array of integers	**atoms**[i] >= 0	"atoms": [9 ,10]	
molecule description fields: atoms, bonds, etc.	see common KET-specification					

Monomer instance definition in KET

Monomer instance appears as an entry ***monomer<N>*** as an element of nodes array inside the root, or inside a **group**.

"**monomer**" type refers to an instance of a **monomerTemplate** and has the following definition:

Field	Description	Required	Type	Value constraints	Usage	Comments
type	object type	yes	string	"type": "monomer″	"type": "monomer"	
id	unique monomer instance id	yes	string	any non-empty string	"id": "mon1"	
seqid	monomers numerations in the group. it's not always straight from left to right.	no	integer	seqid > 0, i.e. starts from 1	"seqid": 1	
origin	monomer coordinates	no	object		"origin": { "x": 1.5, "y": 0.5 }	
■ x	x coordinate	yes	number		"x": 1.5	
■ y	y coordinate	yes	number		"y": 0,5	
alias	it is possible to rename the display alias for a monomer instance	no	string	any non empty string	"symbol": "A1"	
regular monomer						

templateId	refers to id of a monomerTemplate	yes	string	same as for **id** in the monomerTemplate	"templateId": "Ala"	
variant monomer						
operator	logic operator. defines logic relationship between elements in the	yes	string	"enum": ["and", "or"]	"operator": "xor"	
monomer List	list of template ids of the variant monomer	yes	array of string		monomerList["Ala", "Cys"]	

Groups definition in KET

groups are a series of monomers, where each monomer can be connected to other monomer using their attachment points.
In KET, a group like a monomer instance or template can be defined outside the root block using the ***group<N>*** or straight as element of nodes array inside the root.
There are three types of the groups: "group", "variant" and "repeat_unit". They have common and specific fields.

Field	Description	Required	Type	Value constraints	Usage	Comment
type	object type	yes	string	the value can be "group"	"type": "group"	
groupClass	group class	yes	string	the value can be "generic", "variant", "repeatUnit"		
id	group unique id	yes	string	any non-empty unique string	"id": "grp1"	
label	group display label	no	string	any non-empty string	"label": "oxytocine"	
attachment Points	optional attachment point for variant connection	no	array of objects			
▪ **id**	attachment point Id	yes	string			R1...Rn or custom label
▪ **variantList**	group with "type"="variant"	yes	object			
nodes	list of monomers' instances	yes	array of objects		"nodes": [{ "$ref": "monomer1" }, { "$ref": "monomer2" }, { "$ref": "group2" }]	
Group specific (if groupClass = generic)						
connections	connections definitions	no	array of connections			
Variant specific (if groupClass = variant)						
operator	logic operator. define relationship between elements in the **nodes** array.	yes	string	the value can be "xor" or "and"		
ratio	proportion value	no	integer			
Repeating unit specific (if group = repeatUnit)						

repeatRange	repetition range. from = to means fixed repletion count.	yes	object	to >= from	"repeatRange":
					{
					"from": "5",
					"to": "10"
					}	

■ **from**		yes	integer			
■ **to**		yes	integer			

Connections definition in KET

In KET, connections can be defined outside the root block using the reference ***connection<N>*** or straight as element of nodes array inside the root, also the local connections can be defined inside the groups.

Field	Description	Required	Type	Value constraints	Usage
connectionType	object type	yes	string	"single" or "hydrogen"	"connectionType": "hydrogen"
label	optional connection label	no	string	any string	"label": "my comment"

endPoint1	connection endpoint (at least one field should present)	yes	object		"endPoint1": {
					"groupId": "group1",
					"monomerId":"monomer1",
					"attachmentPointId": "R3"
					}

■ **groupId**	group to be connected	no	string	same as for **group<N>**	"groupId": "group1"
■ **monomerId**	monomer inside the group	no	string	same as for **monomer<N>**	"monomerId":"monomer1"
■ **attachmentPointId**	monomer's or group's attachment point	no	string	R1..Rn for monomer's attachment point.	"attachmentPointId": "R3"

endPoint2	same field should be specified as for **endPoint1**	yes	object		"endPoint2": {
					"groupId": "group2",
					"monomerId":"monomer3",
					"attachmentPointId": "R2"
					}

• **groupId**	group to be connected	no	string	same as for **group<N>**	"groupId": "group1"
• **monomerId**	monomer inside the group	no	string	same as for **monomer<N>**	"monomerId":"monomer1"
• **attachmentPointId**	monomer's or group's attachment point	no	string	R1..Rn for monomer's attachment point.	"attachmentPointId": "R3"