
Protocol Documentation

Table of Contents

vmc.proto	1
Allele	1
Genotype	1
Haplotype	1
Identifier	2
Interval	2
Location	2
Position	2
Sequence	3
SequenceIdentifier	3
Completeness	3
Position.Type	3
Scalar Value Types	3

vmc.proto

Allele

Allele represents a single contiguous change on a specific reference sequence

Table 1. Allele Fields

Field	Type	Label	Description
id	string	optional	Allele identifier
location_id	string	optional	Location of feature
replacement	string	optional	replacement sequence

Genotype

Genotype represents multiple changes at a single location

Table 2. Genotype Fields

Field	Type	Label	Description
id	string	optional	Genotype identifier
haplotype_ids	string	repeated	list of haplotypes by id
completeness	Completeness	optional	Completeness of Genotype definition

Haplotype

Haplotype represents a collection of phased changes on a single reference.

Table 3. Haplotype Fields

Field	Type	Label	Description
id	string	optional	Haplotype identifier
sequence_id	string	optional	Alleles are defined on this sequence
completeness	Completeness	optional	Completeness of Haplotype definition
allele_ids	string	repeated	list of haplotypes by id

Identifier

Identifier is a namespaced accession to an object. May be represented as a URI as namespace:accession.

Table 4. Identifier Fields

Field	Type	Label	Description
namespace	string	optional	name of the object authority or algorithm
accession	string	optional	replacement sequence; empty for deletion

Interval

Represents the definite (i.e., not fuzzy) location of a sequence feature using an interval of interbase coordinates.

Interbase coordinates refer to the points *between* residues. For a sequence of length n , $0 \leq \text{start} \leq \text{end} \leq n$, where 0 refers to the point before the start of the sequence, n refers to the point at the end of the sequence. An interval in which $\text{start} == \text{end}$ is a zero width point between two nucleotides. See http://gmod.org/wiki/Introduction_to_Chado#Interbase_Coordinates for more information.

Table 5. Interval Fields

Field	Type	Label	Description
start	uint64	optional	start position
end	uint64	optional	end position

Location

Represents a specific position on a specific sequence.

Table 6. Location Fields

Field	Type	Label	Description
id	string	optional	Location identifier
sequence_id	string	optional	sequence id
position	Position	optional	position of sequence change on seqref

Position

Table 7. Position Fields

Field	Type	Label	Description
type	Position.Type	optional	
interval	Interval	optional	

Sequence

A biological sequence, represented using 1-letter nucleic or amino acids.

Table 8. Sequence Fields

Field	Type	Label	Description
id	string	optional	Internal Sequence id
sequence	string	optional	sequence itself

SequenceIdentifier

Table 9. SequenceIdentifier Fields

Field	Type	Label	Description
sequence_id	string	optional	Internal Sequence id
identifier	Identifier	optional	identifier

Completeness

Declares the completeness of haplotype or genotype definitions.

Table 10. Completeness Values

Name	Number	Description
UNKNOWN	0	Other alleles or haplotypes may exist
UNSPECIFIED	1	Other alleles or haplotypes exist but are not specified
COMPLETE	2	All alleles or haplotypes are included in this definition

Position.Type

Table 11. Position.Type Values

Name	Number	Description
INTERVAL	0	

Scalar Value Types

.proto Type	Notes	C++ Type	Java Type	Python Type
double		double	double	float

.proto Type	Notes	C++ Type	Java Type	Python Type
float		float	float	float
int32	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint32 instead.	int32	int	int
int64	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint64 instead.	int64	long	int/long
uint32	Uses variable-length encoding.	uint32	int	int/long
uint64	Uses variable-length encoding.	uint64	long	int/long
sint32	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int32s.	int32	int	int
sint64	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int64s.	int64	long	int/long
fixed32	Always four bytes. More efficient than uint32 if values are often greater than 2 ²⁸ .	uint32	int	int
fixed64	Always eight bytes. More efficient than uint64 if values are often greater than 2 ⁵⁶ .	uint64	long	int/long
sfixed32	Always four bytes.	int32	int	int
sfixed64	Always eight bytes.	int64	long	int/long
bool		bool	boolean	boolean
string	A string must always contain UTF-8 encoded or 7-bit ASCII text.	string	String	str/ unicode
bytes	May contain any arbitrary sequence of bytes.	string	ByteString	str