

---

# Protocol Documentation

## Table of Contents

types.proto .....	1
Interval .....	1
IntervalEdit .....	1
SequenceReference .....	2
SequenceReference.Namespace .....	2
vmc.proto .....	2
Allele .....	2
Diplotype .....	2
Genotype .....	3
Haplotype .....	3
Scalar Value Types .....	3

## types.proto

### 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](http://gmod.org/wiki/Introduction_to_Chado#Interbase_Coordinates) for more information.

**Table 1. Interval Fields**

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

### IntervalEdit

IntervalEdit represents a located sequence change.

Consider renaming fields to match message name. One possibility is  $\text{location} \Rightarrow \text{interval}$  and  $\text{replacement} \Rightarrow \text{edit}$ , thus matching the message name IntervalEdit.

**Table 2. IntervalEdit Fields**

Field	Type	Label	Description
location	Interval	required	location of sequence change
replacement	string	required	replacement sequence; empty for deletion

## SequenceReference

SequenceReference represents a named reference to a sequence in a database. For the purposes of VMC, it is essential that the mapping from SequenceReference to sequence is many-to-one and immutable.

**Table 3. SequenceReference Fields**

Field	Type	Label	Description
namespace	SequenceReference.Namespace	required	name of recognized sequence reference
accession	string	required	replacement sequence; empty for deletion

## SequenceReference.Namespace

**Table 4. SequenceReference.Namespace Values**

Name	Number	Description
NCBI	0	versioned NCBI identifier, such as NM_000059.3 or NC_000001.10
ENSEMBL	1	versioned Ensembl identifier, such as ENST00000380152.7
LRG	2	Locus Reference Genome; <a href="http://www.lrg-sequence.org/">http://www.lrg-sequence.org/</a>
MD5	100	MD5 of sequence
SHA1	101	SHA1 of sequence
SHA256	102	SHA256 of sequence
SHA512	103	SHA512 of sequence
SEGID	110	seguid of sequence
TD24	111	24-byte Truncated Digest of sequence

## vmc.proto

### Allele

Allele represents a single contiguous change on a specific reference sequence

**Table 5. Allele Fields**

Field	Type	Label	Description
seqref	SequenceReference	required	sequence reference (namespace and accession)
interval	Interval	required	location of sequence change
replacement	string	required	replacement sequence
id	string	optional	Allele identifier

## Diplotype

Diplotype represents a collection of haplotypes.

**Table 6. DiploType Fields**

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

## Genotype

Genotype represents multiple changes at a single location

**Table 7. Genotype Fields**

Field	Type	Label	Description
allele_ids	string	repeated	list of haplotypes by id
id	string	optional	Genotype identifier

## Haplotype

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

**Table 8. Haplotype Fields**

Field	Type	Label	Description
allele_ids	string	repeated	list of haplotypes by id
id	string	optional	Haplotype identifier

## Scalar Value Types

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

<b>.proto Type</b>	<b>Notes</b>	<b>C++ Type</b>	<b>Java Type</b>	<b>Python Type</b>
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 <sup>28</sup> .	uint32	int	int
fixed64	Always eight bytes. More efficient than uint64 if values are often greater than 2 <sup>56</sup> .	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