

```
-- $Id: haplotype.asn,v 1.00 2001/06/08 14:18:59 sherry Exp $
-- $Log: haplotype.asn,v $
-- HAPLOTYPE DEFINITIONS FOR NCBI HAPLOTYPE SUBMISSION SET (NHS)
NHS DEFINITIONS ::=
BEGIN
```

```
EXPORTS NHS-HapList, NHS-PopList, NHS-IndList, NHS-TrioList;
```

```
NHS-HapList ::= SEQUENCE {
    version VisibleString,                -- version number of docsum.asn/docsum.dtd specification
    submitted NSE-Date OPTIONAL ,         -- Date set-list generated
    hapset-list SET OF NSE-hapset        -- List of haplotypes, grouped into submitter-defined sets with
                                        -- common properties such as methodology and/or publication
}
```

```
NHS-hapset ::= SEQUENCE {
    handle VisibleString,                 -- submitter
    set-id VisibleString                  -- submitter name for set of haplotypes
    method VisibleString,                 -- local method id
    method-ex SEQUENCE OF VisibleString OPTIONAL, -- description of deviation from/addition to given method
    citation SEQUENCE OF VisibleString OPTIONAL, -- submitter citations for description of the haplotype set
    hap-set SET OF NHS-hap                -- set of haplotypes
}
```

```
NHS-hap ::= SEQUENCE {
    hap-id VisibleString,                 --submitter haplotype id
    inference-class INTEGER {             -- type of data used to infer haplotype
        pedigree-analysis (1),           -- familial transmission of genotypes in pedigrees
        molecular-cloning (2),           -- genotypes of cloned single molecule
        statistical-inference (3),        -- statistical inference (algorithm cited in method above)
        synthetic-construct (4),         -- haplotype defined by arbitrary means with no evidence
        mixed (5) },                     -- multiple classes used to make inference
    sample-size INTEGER,                 -- number of chromosomes in sample used to define haplotype
    error INTEGER OPTIONAL,              -- measure of error in haplotype inference
    support INTEGER OPTIONAL,            -- measure of support for statistically inferred haplotypes
    snp-set SET OF NHS-hapsnp            -- unordered set of dbSNP submissions with respective allelic states
}
```

```
NHS-hapsnp ::= SEQUENCE {
    rs-id INTEGER,                        -- dbSNP refSNP ID (rs#)
    rs-allele VisibleString,              -- allelic state of rs#
    map-order INTEGER OPTIONAL            -- position of SNP in ordered list of all SNPs defined for the haplotype
                                        -- this information is necessary for future reanalysis of data when map order
                                        -- is potentially ambiguous
    rs-orient INTEGER {                   -- orientation of alleles to dbSNP orientation of the refSNP (data integrity check)
        forward (1),                      -- alleles in rs_allele are reported in same orientation as dbSNP refSNP allele list
        reversed (2) } OPTIONAL,         -- alleles in rs_allele will conform to refSNP allele list when reverse complemented
}
```

```
NHS-PopList ::= SEQUENCE {
    version VisibleString,                -- version number of docsum.asn/docsum.dtd specification
    submitted NSE-Date OPTIONAL ,         -- Date set-list generated
    handle VisibleString,                 --submitter
    method VisibleString,                 -- local method id
    method-ex SEQUENCE OF VisibleString OPTIONAL, -- description of deviation from/addition to given method
    citation SEQUENCE OF VisibleString OPTIONAL, -- submitter citations for description of the haplotype frequency data
    pop-set SET of NHS-pop                --set of population-specific haplotype counts
}
```

```

NHS-pop ::= SEQUENCE {
    pop-id VisibleString,
    sample-size INTEGER,
    count-set SET OF NHS-hapcount,
    pop-descr SEQUENCE OF VisibleString OPTIONAL
}
-- submitter's sample ID
-- number of chromosomes in sample
-- set of haplotype counts in sample
-- description of the sample

NHS-hapcount ::= SEQUENCE {
    hap-id VisibleString,
    count INTEGER
}
-- submitter haplotype id
-- number of chromosomes observed for hap-id in sample

NHS-IndList ::= SEQUENCE {
    version VisibleString,
    submitted NSE-Date OPTIONAL ,
    handle VisibleString,
    method VisibleString,
    method-ex SEQUENCE OF VisibleString OPTIONAL,
    citation SEQUENCE OF VisibleString OPTIONAL,
    indpop-set SET of NHS-indpop
}
-- version number of docsum.asn/docsum.dtd specification
-- Date set-list generated
--submitter
-- local method id
-- description of deviation from/addition to given method
-- submitter citations for description of the haplotype set
-- set of population-specific individual haplotypes

NHS-indpop ::= SEQUENCE {
    pop-id VisibleString,
    pop-descr SEQUENCE OF VisibleString OPTIONAL,
    ind-set SET OF NHS-ind
}
-- submitter's sample ID
-- description of the sample
-- set of individual haplotypes, individuals can be in set multiple times -- once for each determined haplotype

NHS-ind ::= SEQUENCE {
    ind-id VisibleString,
    hap-id VisibleString,
    ped-id VisibleString OPTIONAL
}
-- submitter individual id
-- submitter haplotype id
-- submitter family (pedigree) id

NHS-TrioList ::= SEQUENCE {
    version VisibleString,
    submitted NSE-Date OPTIONAL ,
    handle VisibleString,
    citation SEQUENCE OF VisibleString OPTIONAL,
    trio-set SET of NHS-trio
}
-- version number of docsum.asn/docsum.dtd specification
-- Date set-list generated
--submitter
-- submitter citations for description of the haplotype set
--sets of parent / child trios to define pedigree

NHS-trio ::= SEQUENCE {
    pop-id VisibleString,
    ped-id VisibleString,
    ind-id VisibleString,
    father-id VisibleString OPTIONAL,
    mother-id VisibleString OPTIONAL
}
-- submitter population sample id
-- submitter pedigree id
-- submitter individual ID (ego)
-- submitter id of ego's father
-- submitter id of ego's mother

NSE-Date ::= CHOICE {
    str VisibleString ,
    std NSE-Date-std }
-- for those unparsed dates
-- use this if you can

NSE-Date-std ::= SEQUENCE {
    year INTEGER ,
}
-- NOTE: this is NOT a unix tm struct
-- full year (including 1900)

```

```
month INTEGER OPTIONAL ,  
day INTEGER OPTIONAL ,  
season VisibleString OPTIONAL }
```

```
-- month (1-12)  
-- day of month (1-31)  
-- for "spring", "may-june", etc
```

```
END
```