Trace Archive RFC v.2.4
7/22/2008

Preface

This is a revised version of the original Trace Archive RFC. The purpose of this document is to specify a means of exchange of traces and their ancillary data. In the four years that have elapsed since the Trace Archive was originally developed, the scope and uses of the data have evolved. Revisions have been made this document will clarify, and in some instances change, the contents that are to be submitted in specific fields. To accommodate new data sources, additional fields need to be added and specified. In addition, required data for specific trace types will be clarified. This proposal covers:

- Goals
- Ancillary Information
- Submission Information

The biggest changes involve:

The addition of new fields to accommodate environmental sample data

- These additional fields allow the storage of experimental data related to the field. This is especially important as these sets of data do not fit into our normal taxonomic classification of traces.

Modification of the description of the STRATEGY and TRACE_TYPE_CODE fields

- Currently, STRATEGY and TRACE_TYPE_CODE fields are largely redundant. The proposal involves changing the description of the STRATEGY field to reflect the experimental strategy (rather than the sequencing strategy) used to produce a trace. This should not only better reflect the data, but also make retrieval of large sets of data easier.

Changing the SUBSPECIES_ID field to STRAIN field

- Most of the data in SUBSPECIES_ID field was actually related to STRAIN or cultivar information, so it makes some degree of sense to change the name of this field. Subspecies information can be related in the SPECIES_CODE field.

Addition of new requirements

- Currently, it is quite difficult to use certain data sets because there is not enough ancillary data associated with the trace. Initially, the trace archive had few restrictions because not every field is applicable to every TRACE_TYPE_CODE and STRATEGY combination. There is no change to the fields that are required for all submissions, but there now are fields that are required for specific combinations of STRATEGY and TRACE_TYPE_CODE. You may check requirements in the Validation Table

Accommodation of new sequencing technologies and data types

- As sequencing technology emerges, standard dideoxy sequencing is being supplemented with other sequencing technologies such as chip based resequencing technologies as well as synthesis based technologies like the 454 Life Sciences technology. We will also be accepting other data types, such as AFLP mapping data.

In order to accept these data we are adding new fields, and in some cases additional files, in order to fully capture the raw information

Goals

The original goal of the Trace Archive was largely related to data storage. However, over the last three years, the scope of the Trace Archive has grown substantially. The current goals of the Trace Archive are:

Archival data storage of sequence traces:
- Database storage

Easy retrieval of individual traces and groups of traces:
- Retrieval based on specific ancillary information
- Retrieval based on sequence comparison

The NCBI and Ensembl are collaborating to store all of the traces. There is an ongoing effort to keep the two sites synchronized with respect to trace content.

Ancillary Information

For the Trace Archive to be a useful resource, the archive must contain information describing the traces. Different sequencing centers have different naming schemes that are not mutually exclusive and it is likely that the schemes can change over time. While the name of the trace conveys some information, it is not sufficient for fully describing the data. Sequencing centers generally use the trace name as a unique key within their databases. The Trace Archive will use a combination of the center name and trace name as a unique key. In addition, every trace will be assigned a unique trace identifier. The trace identifier will be an integer value and can also function as a unique key. When the actual trace for a particular record is updated, the current TI will be replaced by a new TI. A change in TI will not occur for an update of ancillary information only. The ancillary data information should be contained in a separate file. This file can be a tab delimited text file or it can use XML format. The format of the TRACEINFO file is described in the Submission Information.

The list of ancillary data fields are described below.

Field List

The ancillary information fields defined for the Trace Archive are listed below.

RED color designates fields that are required

GREEN color designates fields that may be required, depending upon the trace type and strategy employed.

A field may be mandatory, optional or not allowed for a given combination of strategy and trace type as indicated below.

Modifications/additions from the original RFC are in red if these changes affect either a field requirement, or the definition of a field. Slight changes to affect document clarity are not noted.

Name: **ANONYMIZED_ID**
Type: **varchar(100)**
Example: **2222anonym**
Used in projects to maintain the anonymity of donors. In many cases, there my be a controlled access database that can map many anonymized_ids in the trace archive to a single individual id for which phenotypic information may be available.

Name: **REFERENCE_SET_MIN**
Type: **int**
Example: **29829**
This field points to the starting coordinate of the accession.version described in the REFERENCE_ACCESSION field for a entire re-sequencing region. All coordinates should be in 1 base coordinates (i.e. sequences start at base 1, not base 0). This field is required for The Cancer Genome Atlas (TCGA) project. The REFERENCE_ACC_[MIN|MAX] and REFERENCE_SET_[MIN|MAX] should refer to the same REFERENCE_ACC.

Name: **REFERENCE_SET_MAX**
Type: **int**
Example: **29829**
This field points to the starting coordinate of the accession.version described in the REFERENCE_ACCESSION field for a entire re-sequencing region. All coordinates should be in 1 base coordinates (i.e. sequences start at base 1, not base 0). This field is required for The Cancer Genome Atlas (TCGA) project. The REFERENCE_ACC_[MIN|MAX] and REFERENCE_SET_[MIN|MAX] should refer to the same REFERENCE_ACC.

Name: **ACCESSION**
Type: **varchar(30)**
Example: **AC22227**
The ACCESSION is assigned upon deposition to a public repository (GenBank/EMBL/DDBJ). This field will not be applicable to all trace types (primarily WGS). However, if this field contains a valid accession identifier correlation between the primary sequence data (in Trace) and the secondary sequence data (in the public repository) is facilitated.

Name: **EXTENDED_DATA**
Type: **varchar()**
Example:
The '=' sign and the field separator character '|'should be excluded from names and their values. No other validity checks will be performed on the data. Although the number of <field> tags is not limited we are going to limit the total size of the block to 2 KB.

Name: **NCBI_PROJECT_ID**
Type: **int**
Example: **7**
NCBI_PROJECT_ID field would allow to link traces to Genome Project database and easily retrieve sets of traces from each Project. Genome sequencing centers may register their project prior the submission of genomic sequence data. Submitters need not submit sequencing data at the time they register their project.

Name: **PROJECT_NAME**
Type: **varchar(50)**
Example: **New Project**
In this way sequencing centers that are working on the same large project can group all of the traces for this project using a common term. This field has a controlled vocabulary. Sequencing centers wishing to submit data must contact the Trace Archive administrators (trace@ncbi.nlm.nih.gov) to determine a name that all members of the project agree on.

Name: **GENE_NAME**
Type: **varchar(100)**
Example: **transporter 1**
Free text. Mainly this field would be for TRACE_TYPE_CODE='Re-sequencing' or 'ENCODE'. When a group is analyzing a particular gene, they may want to refer to that gene by it's name or some

other common identifier. Gene names not in Entrez Gene can be used (for instance, Ensembl genes).

Name: **AMPLIFICATION_FORWARD**
Type: **varchar(100)**
Example: **GGATTCTGACTAACGAGC**
The AMPLIFICATION_FORWARD field is to allow submitters to define the primers used to amplify templates for sequencing. This field is required when TRACE_TYPE_CODE=PCR or RT-PCR.

Name: **AMPLIFICATION_REVERSE**
Type: **varchar(100)**
Example: **GGATTCTGACTAACGAGC**
The AMPLIFICATION_REVERSE field is to allow submitters to define the primers used to amplify templates for sequencing. This field is required when TRACE_TYPE_CODE=PCR or RT-PCR.

Name: **AMPLIFICATION_SIZE**
Type: **int**
Example: **500**
The AMPLIFICATION_SIZE field allows submitters to define the expected amplification size for a pair of primers (defined in the AMPLIFICATION_FORWARD and AMPLIFICATION_REVERSE fields). This number should be given in base pairs. If TRACE_TYPE_CODE=PCR, the amplification size is based on amplification of genomic DNA. If the TRACE_TYPE_CODE=RT-PCR, then the amplification size is based on amplification of transcript.

Name: **BASE_FILE**
Type: **varchar(200)**
Example: **./mytraces/123clone.fasta**
Tracefiles which do not include the basecalls must provide this information in a separate file. The file designations are recorded in the BASE_FILE and QUAL_FILE fields of the TRACEINFO file. The actual bases are stored in the file designated in the BASE_FILE field. If base calls and quality scores are provided in separate files the information in these files will overwrite any information in the trace (usually *.scf) file. If the base calls and quality scores that would be provided in the BASE_FILE and QUAL_FILE are the same as the information in the trace file DO NOT PROVIDE THE FILE. Providing redundant information complicates the loading process. However, it is important to note that if some formats do not include the quality scores, then these values must be provided as ancillary information. If the center provides the BASE_FILE and QUAL_FILE, then the peak index information should also be provided in a file called PEAK_FILE.

Name: **CENTER_NAME**
Type: **varchar(50)**
Example: **WUGSC**
Sequencing centers wishing to submit data must contact the Trace Archive administrators (trace@ncbi.nlm.nih.gov) to determine a center abbreviation. This abbreviation is used in the CENTER_NAME field. This field has a controlled vocabulary. For the complete list of submitting centers see:
http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=stat&f=xml_list_centers&m=obtain&s=center

Name: **CENTER_PROJECT**
Type: **varchar(100)**
Example: **HBBB**
The CENTER_PROJECT reflects a sequencing center's internal designation for a specific sequencing project. This field can be useful for grouping related traces.

Name: **CHEMISTRY**

Type: **varchar(50)**
Example: **BIGDYEV3.0**

Name: **CHEMISTRY_TYPE**
Type: **char(50)**
Example: **P**
The CHEMISTRY_TYPE uses a controlled list. Accepted values are:

```
Primer
Terminator
p=primer
t=terminator
```

Name: **CHROMOSOME**
Type: **varchar(8)**
Example: **11**
The CHROMOSOME indicates to which chromosome a trace has been assigned. Gene names or cytogenetic positions are not appropriate substitutes for chromosome information.

Name: **CLIP_QUALITY_LEFT**
Type: **int**
Example: **56**
The CLIP_QUALITY_LEFT field indicates the base at the beginning of the sequence at which the read should be clipped due to poor quality sequence. The given value would be the first base of the high quality region of the trace.

Name: **CLIP_QUALITY_RIGHT**
Type: **int**
Example: **256**
The CLIP_QUALITY_RIGHT field indicates the base at the end of the sequence at which the read should be clipped due to poor quality sequence. The given value would be the last base of the high quality region of the trace.

Name: **CLIP_VECTOR_LEFT**
Type: **int**
Example: **75**
The CLIP_VECTOR_LEFT field indicates the base at the beginning of the sequence at which the read should be clipped due to vector sequence. The given value would be the first base of non-vector sequence. This field is required for almost all combinations of STRATEGY and TRACE_TYPE_CODE. This information can be omitted if the INSERT_FLANK_LEFT field is populated or TRACE_TYPE_CODE is PCR or RT-PCR.

Name: **CLIP_VECTOR_RIGHT**
Type: **int**
Example: **275**
The CLIP_VECTOR_RIGHT field indicates the base at the end of the sequence at which the read should be clipped due to vector sequence. The given value would be the last non-vector sequence. This field is required for almost all combinations of STRATEGY and TRACE_TYPE_CODE. This information can be omitted if the INSERT_FLANK_RIGHT field is populated or TRACE_TYPE_CODE is PCR or RT-PCR.
NOTE: Many centers combine vector and quality analysis, and thus have only one set of clip values. In this case, the set of values should be placed in the CLIP_VECTOR_LEFT/CLIP_VECTOR_RIGHT fields.

NOTE: There have been some requests to make all of the clip value fields required. For various reasons, including the note above, this position has not been adopted. The decision was that

either the CLIP_VECTOR_LEFT/CLIP_VECTOR_RIGHT fields should be required or the vector information (SVECTOR_ACCESSION field) should be supplied. However, since most centers use sequencing vectors that are not in GenBank/EMBL/DDBJ it seems more likely that the trim values will be given or INSERT_FLANK_RIGHT and INSERT_FLANK_LEFT fields are populated.

Name: **CLONE_ID**
Type: **varchar(30)**
Example: **RP23-1123F10**
The CLONE_ID field is used to store the identifier related to an individual clone, for example a BAC clone, PAC clone or cDNA clone. If the clone is registered with the clone registry (http://www.ncbi.nlm.nih.gov/genome/clone/), standard clone registry nomenclature (see http://www.ncbi.nih.gov/genome/clone/ nomenclature.shtml for more details) should be used. It is now proposed that this field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=cDNA; TRACE_TYPE_CODE=Any
STRATEGY=EST; TRACE_TYPE_CODE=Any
STRATEGY=CLONEEND; TRACE_TYPE_CODE=CLONEEND
STRATEGY=CLONE; TRACE_TYPE_CODE=Any
STRATEGY=ENCODE; TRACE_TYPE_CODE=SHOTGUN; PrimerWalk; CLONEEND
STRATEGY=FINISHING; TRACE_TYPE_CODE=Any

Name: **CLONE_ID_LIST**
Type: **varchar(30)**
Example: **RP23-200A2;RP23-500P1**
The CLONE_ID_LIST field is used only if STRATEGY =PoolClone. In this case, the list of clones is provided as a semicolon delimited list. If the clones are registered with the Clone Registry, standard clone registry nomenclature should be used (see CLONE_ID field).
Note: The list of clones is not limited, but the size of the individual clone within the list is limited to 30 bytes.
It is now proposed that this field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=PoolClone; TRACE_TYPE_CODE=Any

Name: **COLLECTION_DATE**
Type: **datetime**
Example: **Mar 2 2006 12:00AM**
The COLLECTION_DATE field is used to define the date and time on which an environmental sample was collected. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Env Sample- Geo; TRACE_TYPE_CODE=Any
STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

Name: **CVECTOR_ACCESSION**
Type: **varchar(50)**
Example: **AY451994**
The CVECTOR_ACCESSION field holds the accession number for the cloning vector used. This cloning vector relates to the clone named in the CLONE_ID field.

Name: **CVECTOR_CODE**
Type: **varchar(50)**
Example: **PBACE3.6**
The CVECTOR_CODE field holds the user defined identifier for the cloning vector. Submitters are encouraged to submit all vector sequence information to public repositories. However, it is

understood that many sequencing centers sequence clones from libraries they did not prepare.

Name: **DEPTH**
Type: **float**
Example: **10M**
The DEPTH field is applicable to water samples and earth samples. If the value of this field is NULL, it is anticipated the sample was taken from the surface of the environment. While this field is only applicable to environmental samples, it is not required.

Name: **ELEVATION**
Type: **float**
Example: **500**
If the value of this field is NULL it is assumed the data were obtained at sea level. The field ELEVATION is only applicable to some environmental sample data, but is not a required field.

Name: **ENVIRONMENT_TYPE**
Type: **varchar(250)**
Example: **sea water**
The ENVIRONMENT_TYPE field is used to describe the specific environment from which an environmental sample was taken. While the LATITUDE and LONGITUDE fields describe the location many types of environmental types could exist at this location (for example, soil, sludge, tree roots, etc). This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Env Sample - Geo; TRACE_TYPE_CODE=Any

Name: **FEATURE_ID_FILE**
Type: **varchar(200)**
Example: **./mytraces/chip2.cdf**
The FEATURE_ID_FILE provides the location and sequence of the features for a given chip when TRACE_TYPE_CODE="CHIP".

Name: **FEATURE_ID_FILE_NAME**
Type: **varchar(200)**
Example:
This field is required when TRACE_TYPE_CODE="CHIP".

Name: **FEATURE_SIGNAL_FILE**
Type: **varchar(200)**
Example: **./mytraces/chip2.signal**
The FEATURE_SIGNAL_FILE provides the signal and variance of signal for the features on a given chip when TRACE_TYPE_CODE="CHIP".

Name: **FEATURE_SIGNAL_FILE_NAME**
Type: **varchar(200)**
Example:
This field is required when TRACE_TYPE_CODE="CHIP".

Name: **HI_FILTER_SIZE**
Type: **varchar(50)**
Example: **50 micron**
The HI_FILTER_SIZE field is applicable only to environmental sample data but is not a required field.

Name: **HOST_CONDITION**
Type: **varchar(100)**
Example: **HIV-positive**
The HOST_CONDITION field is only applicable to environmental sample data and is used to describe the condition (healthy, sick, etc) of the host from which a sample was taken.

Name: **HOST_ID**
Type: **varchar(100)**
Example: **yerkes pedigree #C0479 'Clint'**
The HOST_IDENTIFIER field is only applicable to environmental sample data and is used to capture the unique name for the specific host from which a sample was obtained. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

Name: **HOST_LOCATION**
Type: **varchar(100)**
Example: **rumen**
The HOST_LOCATION field is only applicable to environmental sample data and is used to describe the specific part of the host from which the sample was obtained, for example: dental plaque, hindgut, root surfaces. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

Name: **HOST_SPECIES**
Type: **varchar(100)**
Example: **Pan troglodytes**
The HOST_SPECIES field is only applicable to environmental sample data. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Env Sample- Host; TRACE_TYPE_CODE=Any

Name: **INSERT_FLANK_LEFT**
Type: **varchar(100)**
Example:
AAGGTGCGATGCAGTGGCAGTAGCAGTGTCGACGTGACGATTCGTCCGGA
The INSERT_FLANK_LEFT field should provide from 50 up to 100 bases of sequence (including linkers) to the left of the cloning junction. This information will allow users to perform their own vector trimming of reads. This field is required for almost all combinations of STRATEGY and TRACE_TYPE_CODE. This field can be omitted if CLIP_VECTOR_LEFT is populated. However, INSERT_FLANK_LEFT is the preferred choice. If there was no cloning step involved in the sequencing, please populate the field with 'NONE'.

Name: **INSERT_FLANK_RIGHT**
Type: **varchar(100)**
Example:
AAGGCGCGATGCAGTGAGCGAGGCTGACGTCGGCTAGCGTCGCGTCGGGT
The INSERT_FLANK_RIGHT field should provide from 50 up to 100 bases of sequence (including linkers) to the right of the cloning junction. This information will allow users to perform their own vector trimming of reads. This field is required for almost all combinations of STRATEGY and TRACE_TYPE_CODE. This field can be omitted if CLIP_VECTOR_RIGHT is populated. However, INSERT_FLANK_RIGHT is the preferred choice. If there was no cloning step involved in the sequencing, please populate the field with 'NONE'. It is anticipated that if INSERT_FLANK_LEFT is populated that INSERT_FLANK_RIGHT will also be populated. It is not anticipated that a mixture of clip values and junction sequence will be specified. (i.e. CLIP_VECTOR_LEFT and INSERT_FLANK_RIGHT populated for the same record.

Name: **INSERT_SIZE**
Type: **int**
Example: **2000**
The INSERT_SIZE field indicates the expected insert size of the clone that is sequenced. It is understood that this is an estimate based upon the average insert sizes found in a given library. However, this information is critical for certain experiments, such as whole genome assembly. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Any; TRACE_TYPE_CODE=WGS
STRATEGY=Any; TRACE_TYPE_CODE=WCS
STRATEGY=cDNA; TRACE_TYPE_CODE=CLONEEND
STRATEGY=CLONEEND; TRACE_TYPE_CODE=CLONEEND

Name: **INDIVIDUAL_ID**
Type: **varchar(100)**
Example: **NA12345**
The INDIVIDUAL_ID field provides a center specific unique id that can associate a specific trace to an individual. This will be used primarily for population based studies.

Name: **INSERT_STDEV**
Type: **int**
Example: **200**
The INSERT_STDEV field reflects the approximate standard deviation of the insert size. It is understood that this information is an approximation and may change as better data is obtained. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Any; TRACE_TYPE_CODE=WGS
STRATEGY=Any; TRACE_TYPE_CODE=WCS
STRATEGY=cDNA; TRACE_TYPE_CODE=CLONEEND
STRATEGY=CLONEEND; TRACE_TYPE_CODE=CLONEEND

Name: **ATTEMPT**
Type: **tinyint(1-255)**
Example: **2**

Name: **LATITUDE**
Type: **float**
Example: **54.736**
The LATITUDE field is required to describe the collection of some environmental sample data. The latitude range is [-90,90] with the equator as 0 latitude and positive values of latitude are north of the equator. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE: STRATEGY=Env Sample- Geo; TRACE_TYPE_CODE=Any

Name: **LO_FILTER_SIZE**
Type: **varchar(50)**
Example: **25 micron**
The LO_FILTER_SIZE field is only applicable to environmental sample data but is not a required field.

Name: **LONGITUDE**
Type: **float**
Example: **-86.403**

The LONGITUDE field is required to describe the collection of some environmental sample data. The longitude is ranging from 0° at the Prime Meridian to +180° eastward and #180° westward. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Env Sample- Geo; TRACE_TYPE_CODE=Any

***Name:* LIBRARY_ID**
***Type:* varchar(100)**
***Example:* RP23**
The LIBRARY_ID field documents the source library of the archival clone resource. Many genomic libraries have been registered with the Clone Registry (http://www.ncbi.nlm.nih.gov/genome/clone/) and the standard nomenclature (http://www.ncbi.nlm.nih.gov/genome/clone/clbrowse.cgi) should be used for these libraries. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=cDNA; TRACE_TYPE_CODE=Any
STRATEGY=EST; TRACE_TYPE_CODE=Any
STRATEGY=CLONEEND; TRACE_TYPE_CODE=CLONEEND
STRATEGY=CLONE; TRACE_TYPE_CODE=Any
STRATEGY=ENCODE; TRACE_TYPE_CODE=SHOTGUN; PrimerWalk; CLONEEND

***Name:* ORGANISM_NAME**
***Type:* varchar(100)**
***Example:* Acanthocybium solandri**
The ORGANISM_NAME field is used to classify the read by species for BARCODE data, using proper taxonomic name in accordance with Taxonomy Browser. SPECIES_CODE="BARCODE SPECIES" for all traces from this project. This field would be required for the STRATEGY=BARCODE.

***Name:* PEAK_FILE**
***Type:* varchar(200)**
***Example:* ./mytraces/123clone.peak**
Consult the BASE_FILE field description for more information.

***Name:* PH**
***Type:* float**
***Example:* 7.2**
The PH field is only applicable to environmental sample data but is not a required field.

***Name:* PICK_GROUP_ID**
***Type:* int**
***Example:* 939065**

***Name:* PLACE_NAME**
***Type:* varchar(250)**
***Example:* Octopus Springs**
The PLACE_NAME field is applicable to environmental sample data, but is not required.

***Name:* PLATE_ID**
***Type:* varchar(32)**
***Example:* 203**
The PLATE_ID and WELL_ID fields are intended to identify the storage location of the sequencing template (not the library well coordinate of an archival clone named in the CLONE_ID field). This

may enable flipped or contaminated trays to be easily identified. If a particular experiment did not require the use of a plate, please populate this field with '0'.

Name: **POPULATION_ID**
Type: **varchar(100)**
Example: **CEPH**
The POPULATION_ID field is used to capture center specific designations of groups of individuals. This will likely only be useful in population studies (usually STRATEGY=SNP).

Name: **PREP_GROUP_ID**
Type: **varchar(30)**
Example: **A2**

Name: **PRIMER**
Type: **varchar(200)**
Example: **GAATACCTACGATCGCC**
The value of the PRIMER field is the actual base sequence of the sequencing primer used. If a center uses a primer extensively, the primer sequence can be entered into the list of primer codes and the PRIMER_CODE field can be used.

Name: **PRIMER_CODE**
Type: **varchar(30)**
Example: **Sp6**

Name: **PRIMER_LIST**
Type: **varchar(100)**
Example: **AAGGTCTGCGCGTGTC;AGCTGCGTACGTAATCG;**
This field is required if Strategy ="AFLP" and TRACE_TYPE_CODE ="PCR".

Name: **PROGRAM_ID**
Type: **varchar(100)**
Example: **phred-19990722h**
The PROGRAM_ID field is used to indicate the base calling program. This field is free text. Program name, version numbers or dates are very useful. More example values:

- **phred-19980904e**
- **abi-3.1**
- **ATQA**
- **TraceTuner**
- **Licor**
- **Megabase**
- **Beckman**

Name: **QUAL_FILE**
Type: **varchar(200)**
Example: **./mytraces/123clone.fasta.qs**
See note associated with the BASE_FILE field.

Name: **REFERENCE_ACCESSION**
Type: **varchar(50)**
Example: **NT_029829.1**
This field is required for the following combination of STRATEGY and TRACE_TYPE_CODE: STRATEGY=Re-sequencing; Comparative TRACE_TYPE_CODE=Any

Name: **REFERENCE_ACC_MIN**
Type: **int**
Example: **29829**
This field points to the starting coordinate of the accession.version described in the REFERENCE_ACCESSION field. All coordinates should be in 1 base coordinates (i.e. sequences start at base 1, not base 0). This field is required for the following combination of STRATEGY and TRACE_TYPE_CODE: STRATEGY=Re-sequencing; TRACE_TYPE_CODE=SHOTGUN; PCR; RT-PCR

Name: **REFERENCE_ACC_MAX**
Type: **int**
Example: **30929**
This field points to the finishing coordinate of the accession.version described in the REFERENCE_ACCESSION field. All coordinates should be in 1 base coordinates (i.e. sequences start at base 1, not base 0). This field is required for the following combination of STRATEGY and TRACE_TYPE_CODE: STRATEGY=Re-sequencing; TRACE_TYPE_CODE=SHOTGUN; PCR; RT-PCR

Name: **REFERENCE_OFFSET**
Type: **int**
Example: **1520899**
This field points to the starting coordinate of the accession.version described in the REFERENCE_ACCESSION field. All coordinates should be in 1 base coordinates (i.e. sequences start at base 1, not base 0). This field is required for the following combination of STRATEGY and TRACE_TYPE_CODE: STRATEGY=Re-sequencing; TRACE_TYPE_CODE=CHIP

Name: **RUN_DATE**
Type: **datetime**
Example: **2000-10-28**

Name: **RUN_GROUP_ID**
Type: **varchar(30)**
Example: **group2**

Name: **RUN_LANE**
Type: **int**
Example: **1**
The RUN_LANE documents the specific lane or capillary on which a trace was obtained.

Name: **RUN_MACHINE_ID**
Type: **varchar(30)**
Example: **machine2**

Name: **RUN_MACHINE_TYPE**
Type: **varchar(30)**
Example: **ABI 310**

Name: **SALINITY**
Type: **float**
Example: **20#**
The SALINITY field is only applicable to environmental sample data but is not a required field.

Name: **SEQ_LIB_ID**
Type: **varchar(255)**
Example: **22194**
The SEQ_LIB_ID field is the center identifier for the M13/PUC based clone that is actually sequenced. This will allow grouping of traces by the actual ligation event and is applicable to most projects. This value will be unique within a given center. This field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY=Any; TRACE_TYPE_CODE=SHOTGUN
STRATEGY=Any; TRACE_TYPE_CODE=WGS/WCS

Name: **SOURCE_TYPE**
Type: **varchar(50)**
Example: **GENOMIC DNA**
The SOURCE_TYPE field consists of a code. Possible values are:

- **G = Genomic DNA (includes PCR products from genomic DNA)**
- **N = Non Genomic DNA (EST, cDNA, RT-PCR, screened libraries)**
- **VIRAL RNA = Viral RNA**
- **SYNTHETIC = Synthetic DNA**

Accepted values are G, N, GENOMIC, NON GENOMIC, VIRAL RNA, SYNTHETIC

Name: **SPECIES_CODE**
Type: **varchar(100)**
Example: **Homo sapiens**
The SPECIES_CODE field is used to classify the read by species, using proper taxonomic names where possible. This field currently is maintained as a controlled vocabulary. For a list of species currently contained within the Trace Archive, see:
http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=stat&f=xml_list_species&m=obtain&s=species
To submit a new species, please contact us (trace@ncbi.nlm.nih.gov) prior to submission. For cases in which it is unclear of the taxonomic origin of a specific trace the taxonomic classification 'ENVIRONMENTAL SEQUENCE' can be used in a case of environmental samples or 'ARTIFICIAL SEQUENCE' in a case of artificial material. A second proposal for this field involves incorporating subspecies information into the species code identifier and making the field SUBSPECIES_ID obsolete.

Name: **STRATEGY**
Type: **varchar(50)**
Example: **MODEL VERIFY**
In the original RFC, the STRATEGY field was proposed to contain the sequencing STRATEGY used in obtaining the trace. This definition made this field largely redundant with the TRACE_TYPE_CODE field. The proposal in this new version of the RFC is to make this field reflective of the experimental STRATEGY used when obtaining the trace. In some cases, this may still be redundant with the TRACE_TYPE_CODE field. Some records in the Trace Archive already contain some values in this field that are reflective of this idea. For example, the STRATEGY 'MODEL VERIFY' was proposed for a group of traces that were obtained in the process of verifying proposed gene models. In addition to conveying some information as to the original purpose of the trace, it will likely be useful in retrieving groups of traces in batch sets. It is proposed that this would be a controlled vocabulary, but that submitters would contribute to this list as needed to define various experiments and projects.

Original values:

- **CCS: Concatenated cDNA sequencing**
- **CLONE: Clone based sequencing**
- **ENCODE: Reads generated for the Encode project**
- **MODEL VERIFY: traces obtained to verify proposed gene models**
- **POOLCLONE: Pools of clones (BACs mostly)**
- **TRANSPOSON: Transposon based sequencing**
- **WCS: Whole Chromosome shotgun sequencing**

- **WGS: Whole Genome shotgun sequencing**

Current values (this list would continually be expanding):
- **AFLP: Amplified Fragment Length Polymorphism**
- **BARCODE: DNA sequence analysis of a uniform target gene to enable species identification**
- **CCS: Concatenated cDNA sequencing**
- **cDNA: Sequences generated in the process of sequencing cDNA clones**
- **CF-S: Cot-filtered single/low-copy genomic DNA**
- **CF-M: Cot-filtered moderately repetitive genomic DNA**
- **CF-H: Cot-filtered highly repetitive genomic DNA**
- **CF-T: Cot-filtered theoretical single-copy DNA**
- **CLONE: Genomic clone based (hierarchical) sequencing**
- **CLONEEND: Sequences generated from the end of a clone (BAC/PAC/Fosmid or cDNA)**
- **Comparative: Sequences obtained using primers design from related species**
- **CTS: Concatenated Tag Sequencing**
- **Env Sample-GEO: Geographically generated environmental sample**
- **Env Sample-Host: Environmental samples collected from a specific host**
- **EST: single pass sequencing of cDNA templates**
- **FINISHING: a read specifically made for finishing, could be either BAC finishing or Whole Genome Assembly (WGA) finishing**
- **MODEL VERIFY: Sequences obtained to verify proposed gene models**
- **PoolClone: Pools of clones (BACs mostly)**
- **SNP: Reads used for SNP identification**
- **TARGETED LOCUS: Sequences obtained from templates generated by primers designed to amplify a specific genetic locus**
- **Re-sequencing: Re-sequencing of targeted genomic regions**
- **RT-PCR: Sequences obtained using templates generated by Reverse Transcriptase Polymerase Chain Reaction**
- **WGA: Whole Genome Assembly**

Name: **SUBMISSION_TYPE**
Type: **varchar(50)**
Example: **NEW**
The SUBMISSION_TYPE field allowed values:
- **NEW – use to submit new data**
- **UPDATE – use to renew traces and their ancillary information. Previous data will be saved with their TI's; new traces with the same trace_name's will receive new TI's and they will become active**
- **UPDATEINFO – use to update or add ancillary information for already existing traces without re-submitting the entire package of data**
- **WITHDRAW – use to withdraw traces**

Name: **STRAIN**
Type: **varchar(50)**
Example: **C57BL/6J**
Strain is required for Strategy = "SNP"

Name: **SVECTOR_ACCESSION**
Type: **varchar(50)**
Example: **X52325**

Name: **SVECTOR_CODE**
Type: **varchar(50)**
Example: **pBluescript SK(+)**

Name: **TEMPERATURE**
Type: **float**

Example: **30**
The TEMPERATURE field is only applicable to environmental sample data but it is not a required field.

Name: **TEMPLATE_ID**
Type: **varchar(50)**
Example: **HBBBA2211**
The TEMPLATE_ID field is used to uniquely identify the actual template that is sequenced. This field, in conjunction with the TRACE_END field, can be used to identify traces that should be marked as 'mate_pairs' because they come from opposite ends of the same clone.

Name: **TRACE_END**
Type: **varchar(50)**
Example: **F**
The TRACE_END field can have the following values:

- **F: FORWARD**
- **R: REVERSE**
- **N: UNKNOWN**

Name: **TRACE_FILE**
Type: **varchar(200)?**
Example: **./traces/TRACE001.scf**

Name: **TRACE_FORMAT**
Type: **varchar(20)**
Example: **scf**
The TRACE_FORMAT field can have the following values:

- **SFF (new) - Standard Flowgram Format.**
- **SCF - A standard file format for data from DNA sequencing instruments.**
- **ZTR - The ZTR format is used for storing analogue chromatogram data from DNA sequencing instruments.**
- **ABI - A ABI-tracefile is a binary file including the tracedata and the sequence.**

Name: **TRACE_NAME**
Type: **varchar(250)**
Example: **HBBBA1U2211**
The TRACE_NAME field must be unique within a center, but is not required to be unique between centers. The combination of TRACE_NAME and CENTER_NAME act as a unique key within the Trace Archive.

Name: **TRACE_TYPE_CODE**
Type: **varchar(50)**
Example: **wgs**
The field TRACE_TYPE_CODE reflects the sequencing STRATEGY used to obtain the trace.

Original values:

- **CLONEEND: BAC/PAC/fosmid end sequence**
- **EST: Expressed sequence tag sequencing- single pass sequencing of a cDNA template**
- **FINISHING: a read generated for finishing a BAC project**
- **GSS: Genome Survey Sequences**
- **PCR: Sequences obtained using templates generated by Polymerase Chain Reaction**
- **RT-PCR: Sequences obtained using templates generated by Reverse Transcriptase Polymerase Chain Reaction**
- **SHOTGUN: generally refers to BAC based shotgun sequencing**

- **WCS: Whole Chromosome Shotgun**
- **WGS: Whole Genome Shotgun**

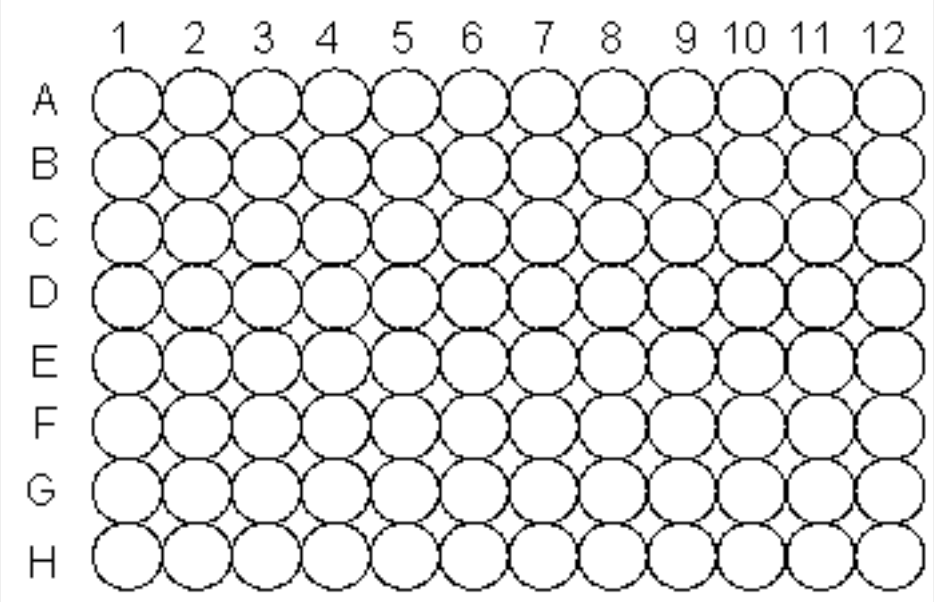
Current values:
- **CHIP: Sequences obtained using microarrays (also called DNA chips or gene chips)**
- **CLONEEND: Sequences generated from the end of a large insert (BAC/PAC/Fosmid) or cDNA clone**
- **EST: Single Pass Expressed Sequence Tag**
- **HTP SELEX: High throughput SELEX**
- **OTHER: Other than PCR, PrimerWalk, SHOTGUN or TRANSPOSON for FINISHING STRATEGY**
- **PCR: Sequences obtained using templates generated by genomic Polymerase Chain Reaction**
- **PrimerWalk: Sequences generated through a primer walking step**
- **RT-PCR: Sequences obtained using templates generated by Reverse Transcriptase Polymerase Chain Reaction**
- **SHOTGUN: Shotgun sequencing of clones (genomic or cDNA)**
- **TRANSPOSON: Sequences obtained using templates generated by transposons**
- **WCS: Whole Chromosome Shotgun**
- **WGS: Whole Genome Shotgun**

Obsolete values:
- **454: Sequences obtained using the 454 technology**

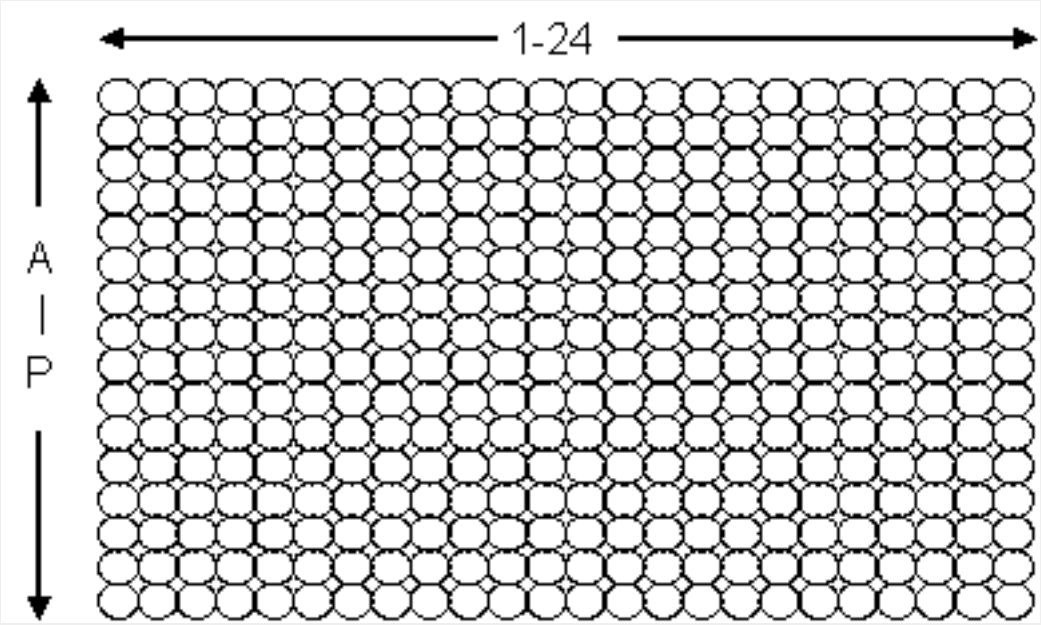
Name: **TRANSPOSON_CODE**
Type: **varchar(50)**
Example: **Mu transposon**
This TRANSPOSON_CODE field would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY =Any; TRACE_TYPE_CODE =TRANSPOSON

Name: **TRANSPOSON_ACC**
Type: **varchar(50)**
Example: **X00913**
The TRANSPOSON_ACC would be required for the following combination of STRATEGY and TRACE_TYPE_CODE:
STRATEGY =Any; TRACE_TYPE_CODE =TRANSPOSON

Name: **WELL_ID**
Type: **varchar(50)**
Example: **A1**
The field WELL_ID in combination with the field PLATE_ID, is used to define the storage location of the sequencing reaction (see note with the field WELL_ID). Typically, sequencing reactions are performed in standard microtiter dishes having either 96 or 384 wells. (see standard configurations below).



Standard 96 well microtiter configuration



Standard 384 well microtiter configuration

Internal Fields

Fields automatically assigned for each record as it is uploaded. Part of these fields is available to the user either following the submitter provided ancillary information on the webpage, or in the <ncbi_trace_archive> block in the xml_info retrieval section.

Name: **TI**
Type: **int**
Example: **304753779**
It is given for a record at the loading stage, and any record, or number of records can be obtain by their identifiers. These numbers can be used to query the database, either in a list "1,2,3,7,100003", through specifying a range "1-10" or in a combination of both "1,2, 5-10".

Name: **BASES_20**
Type: **smallint**

Example: **50**
These fields can be used in a query to limit retrieve for the quality traces only.
Example of query:

center_name = 'ABC' and BASES_20 > 100

Warning: There are some depositions that don't have quality scores. This is likely due to the center submitting ABI files and not providing quality calls separately.

Name: **BASES_40**
Type: **smallint**
Example: **50**
These fields can be used in a query to limit retrieve for the quality traces only.
Example of query:

center_name = 'BCM' and BASES_40 > 70

Warning: There are some depositions that don't have quality scores. This is likely due to the center submitting ABI files and not providing quality calls separately.

Name: **BASES_60**
Type: **smallint**
Example: **50**
These fields can be used in a query to limit retrieve for the quality traces only.
Example of query:

center_name = 'WIBR' and BASES_60 > 50

Warning: There are some depositions that don't have quality scores. This is likely due to the center submitting ABI files and not providing quality calls separately.

Name: **LOAD_DATE**
Type: **smalldatetime**
Example: **Jan 8 2001 11:59AM**
This field helps to specify retrieve data by date.
Example of query:

species_code='MUS MUSCULUS' and load_date >= '12/11/2005'

Name: **STATE**
Type: **varchar**
Example: **active**
Possible values:
- **active**
- **updated**
- **withdrawn**

Name: **REPLACED_BY**
Type: **int**
Example: **304753779**
This field points to the more recent data set. If trace was updated then the REPLACED_BY field stores the TI for the new trace. If only ancillary information has been updated, then replaced_by=0 and is not shown.

Name: **UPDATE_DATE**
Type: **smalldatetime**
Example: **Jul 19 2001 3:48PM**
This field is used to store the date of the last update.

Name: **MATE_PAIR**
Type: **int**
Example: **203682255**
MATE PAIR is the pair of reads obtained from two ends of the same template (FORWARD and REVERSE).

Name: **TAXID**
Type: **int**
Example: **10090**
This field links Trace Archive with NCBI Taxonomy Browser.

Name: **BASECALL_LENGTH**
Type: **int**
Example: **396**

Obsolete fields

Name: **ASSEMBLY_ID**
Type: **varchar(50)**
Example: **NCBI Build 33**
Please use REFERENCE_ACCESSION and REFERENCE_OFFSET

Name: **CHROMOSOME_REGION**
Type: **varchar(50)**
Example: **2:105000-106000**
Please use REFERENCE_ACCESSION and REFERENCE_OFFSET.

Name: **SUBSPECIES_ID**
Type: **varchar(50)**
Example: **Verus**
Please use STRAIN and SPECIES_CODE. Subspecies information should be incorporated into the SPECIES_CODE field.

Name: **TRACE_DIRECTION**
Type: **varchar(50)**
Example: **FORWARD**
Please use TRACE_END instead.

Submission Information

The submitting data should be placed on a provided by NCBI secure FTP site (ftp-trace.ncbi.nih.gov). Contact trace@ncbi.nlm.nih.gov to obtain a secure FTP account. Please have a contact information as well as a full center's name and the center's acronym provided with the request.

All submissions made to NCBI via ftp are automatically picked up by Ensembl. (http://trace.ensembl.org)

Submissions made to Ensembl are placed on NCBI FTP site to pick up and load.

Each submission is a single file in UNIX USTAR format compressed with "gzip" utility. It is suggested to have the size of the submission file between 1 and 4 GB. It also is

suggested to use unique names for the submissions and include the center's name and the date into its name.

All submissions when extracted should have a top directory. The top directory may be named similar to the submission's file. All ancillary files should be placed under that directory. In case when the submission should contain trace files at least one more directory should be introduced to the top directory and all trace files should be placed under that directory.

Below is what should be placed under the top directory.

- TRACEINFO.xml or TRACEINFO.txt: either one must be present. This is the main file describing the submission. It contains ancillary data and references to trace files if necessary. It can be either in XML or tab delimited format.
- MD5: md5 hashes, suggested to be present.
- README: free text describing this volume and preparation.

Below are examples of the submission directory hierarchy and examples of TRACEINFO ancillary files.

The trace files should not appear in the top level directory, but rather should be in a subdirectory. It is suggested to use the name of the traces or the name of the project for subdirectories. There may be subdirectories within and this is encouraged to group traces.

NEW and UPDATE submissions

NEW and UPDATE submissions should have the structure shown below:

```
TOP_DIRECTORY/
TOP_DIRECTORY/TRACEINFO.txt
TOP_DIRECTORY/MD5
TOP_DIRECTORY/README
TOP_DIRECTORY/traces
TOP_DIRECTORY/traces/HBBA/
TOP_DIRECTORY/traces/HBBA/HBBAA1U0001.scf
TOP_DIRECTORY/traces/HBBA/HBBAA1U0002.scf
TOP_DIRECTORY/traces/HBBA/HBBAA1U0003.scf
...
```

Examples are available for download:
ftp://ftp.ncbi.nlm.nih.gov/pub/TraceDB/misc/examples

The ancillary TRACEINFO file describes the submitted data as well as points to the location of the chromatograms. XML format is preferable, since it is easier for a human to read if necessary. The ancillary data requirements are in the Validation Table (Excel format) for specific combinations of STRATEGY and TRACE_TYPE_CODE. Both types of ancillary files can contain common fields section at the beginning of it. This section defines common for the submission values if any.

TRACEINFO.xml example
If the trace info is provided as an XML file the info fields will serve as the tags. To preserve the grouping, the trace_volume tag is used.

```
<?xml version="1.0" encoding="UTF-8"?>
<trace_volume>
   <common_fields>
      <center_name>CENTER NAME ACRONYM IS HERE</center_name>
      <submission_type>NEW</submission_type>
      <strategy>WGS</strategy>
      <trace_type_code>WGS</trace_type_code>
      <center_project>Gorilla WGS</center_project>
```

```
      <source_type>G</source_type>
      <species_code>Gorilla gorilla</species_code>
      <insert_size>1500</insert_size>
   </common_fields>
   <trace>
      <template_id>HBBAA1U0001</template_id>
      <trace_name>HBBAA1U0001</trace_name>
      <trace_file>traces/HBBA/HBBAA1U0001.scf</trace_file>
      <trace_format>scf</trace_format>
      <trace_end>R</trace_end>
      <clip_vector_left>56</clip_vector_left>
      <clip_vector_right>737</clip_vector_right>
      <run_machine_id>legrenzi</run_machine_id>
      <chemistry>BIGDYEV2</chemistry>
      <program_id>phred version=0.020425.c</program_id>
      <run_machine_type>ABI 3700</run_machine_type>
   </trace>
   <trace>
      <template_id>HBBAA1U0002</template_id>
      <trace_name>HBBAA1U0002</trace_name>
        ...more info...
    </trace>
</trace_volume>
```

TRACEINFO.txt example
Tabular file has the following format, and either has no extension at all, or its name is extended with '.txt' or '.tbl'. Data represents the actual values of the fields described in the header. It is also tab delimited

```
center_name     = CENTER NAME ACRONYM IS HERE
submission_type = NEW
strategy        = WGA
trace_type_code = WGS
center_project  = Gorilla WGS
source_type     = G
species_code    = Gorilla gorilla
insert_size     = 1500
template_id    trace_name    trace_file    trace_format    trace_end
clip_vector_left    clip_vector_right    run_machine_id    chemistry
program_id    run_machine_type
HBBAA1U0001    HBBAA1U0001    traces/HBBA/HBBAA1U0001.scf    scf    R   56
737    legrenzi    BIGDYEV2    phred version=0.020425.c   ABI 3700   s2SCF
scf    F    44    793    agricola    BIGDYEV2    phred version=0.020425.c
ABI 3700
HBBAA1U0002    HBBAA1U0002  ...
    ...
```

MD5 example

```
728018368a7820c50cbaad633bc608a1  TRACEINFO
0cbaad633bc608a1728018368a7820c5  traces/TRACE0001.scf
```

UPDATEINFO submission

An UPDATEINFO submission should have the structure shown below:

```
TOP_DIRECTORY/
TOP_DIRECTORY/TRACEINFO.txt
TOP_DIRECTORY/MD5
TOP_DIRECTORY/README
```

TRACEINFO file in this case has to have SUBMISSION_TYPE=UPDATEINFO, the unique keys to the traces(CENTER_NAME and TRACE_NAME) and fields with their

values that you wish to update. These data will be uploaded into our database without changing the ti's and the rest information. This file can contain common fields section at the beginning of it. This section defines common for the submission values if any.

TRACEINFO.xml example

If the trace info is provided as an XML file the info fields will serve as the tags. To preserve the grouping, the trace_volume tag is used.

```
<?xml version="1.0"?>
<trace_volume>
    <common_fields>
        <center_name>CENTER NAME ACRONYM IS HERE</center_name>
        <submission_type>UPDATEINFO</submission_type>
        <trace_type_code>WGS</trace_type_code>
        <insert_size>40000</insert_size>
    </common_fields>
    <trace>
        <trace_name>HBBA0001</trace_name>
        <template_id>template_id_HBBA0001</template_id>
        ...more info...
    </trace>
    <trace>
        <trace_name>HBBA0002</trace_name>
        <template_id>template_id_HBBA0002</template_id>
        ...more info...
    </trace>
</trace_volume>
```

TRACEINFO.txt example

Tabular file has the following format, and either has no extension at all, or its name is extended with '.txt' or '.tbl'. Data represents the actual values of the fields described in the header. It is also tab delimited

```
SUBMISSION_TYPE=UPDATEINFO
CENTER_NAME=CENTER NAME ACRONYM IS HERE
trace_name   clip_vector_left    clip_vector_right     more fields (if
necessary)...
my_trace1     33   89    ...
my_trace2     19   80    ...
my_trace2     1    68    ...
more trace_name's...
```

MD5 example

```
728018368a7820c50cbaad633bc608a1  TRACEINFO
0cbaad633bc608a1728018368a7820c5  traces/TRACE0001.scf
```

WITHDRAW submission

To delete traces use SUBMISSION_TYPE =WITHDRAW

A WITHDRAW submission is a TRACEINFO file inside a tar file, just as any regular Trace Archive submission. It should have the structure shown below:

```
TOP_DIRECTORY/
TOP_DIRECTORY/TRACEINFO.txt
```

TRACEINFO.txt example

WITHDRAW type of submission is very similar to UPDATEINFO, except you do not have to supply extra fields but center_name, trace_name and submission_type=WITHDRAW

```
submission_type = WITHDRAW
center_name     = CENTER NAME ACRONYM IS HERE
trace_name
my_trace1
my_trace2
my_trace2
   ...
```

Tracking Submissions

When a submission is loaded a log file is generated. This log file contains the ti and read name for passed reads and a list of the reads that were rejected.

If more than 5% of the reads from a particular submission fail, the entire submission will be rejected.

A tracking system has been implemented that will allow the tracking of individual submissions. Each FTP submission is given a unique tracking identifier (SID). Submissions can be tracked by name, SID, date or status. The submitting center will be notified via ftp when a submission has been processed.

After each submission has been processed log files documenting the load are placed on the FTP site.

There is an ability to track the submissions with query_tracedb Perl script. The output is in XML format.

Examples:

```
$ query_tracedb "track name='NISC_mkp_2006-09-22.tar.gz'"
$ query_tracedb "track sid=174661"
$ query_tracedb "track name in ('NISC_mkp_2006-09-22.tar.gz',
'NISC_jyp_2006-09-22.tar.gz')"
$ query_tracedb "track sid in (174661, 174657)"
```

If submission does not completely comply to the RFC it will be either rejected or a warning will be sent.

Some ancillary fields are mutually exclusive or not required for a particular type of submission. Please do not include redundant fields into the submission; it can be rejected because of this. For example, if no chromosome information is available for a read, the CHROMOSOME field should not be included.

If a read fails the reason of it failure will be documented in the log file. For example it can fail for the following reasons:

- Information in the ancillary information file, but no trace file
- Zero length trace file
- Number of bases does not match the number of quality values
- There is a trace file but no ancillary information
- If the SUBMISSION_TYPE field has the value 'NEW' but the values in the CENTER_NAME and TRACE_NAME fields are already in the database, the read will be rejected.
- If the same read name is found more than one time in the tar file all reads with that name are failed.