

A The DAS XML Specification

The following is a summary of the current DAS specification, It is based on version 1.01 (October 2001) of the specification. The specification was originally written by Lincoln D. Stein, Sean R. Eddy, and Robin D. Dowell. The current version of the specification is available at <http://www.biodas.org/documents/spec.html>.

A.1 Reference Sequence IDs

Reference sequence IDs indicate a segment of the genome. They can correspond to low-level primary sequences such as sequenced clones, or to higher-level assemblies such as contigs.

A reference ID can contain any set of printable characters (including the space character), but not the colon character (":"), which is reserved for separating reference IDs from sequence ranges (see below). The newline, tab and carriage return characters are also reserved for future use.

A data source that uses the colon character for its internal IDs must map this character to another one on the way out and on the way in. For example:

Client's request	Server's internal id	Response to client
gi-123456	gi:123456	gi-123456
gi-123456:1,1000	gi:123456 start=1 stop=1000	gi-123456:1,1000

A.2 The Queries

This section lists the queries recognized by sequence and annotation servers. Each of these queries begins with some site-specific prefix, denoted here as *PREFIX*. The other meta-variable used in these examples is *DSN*, which is a symbolic data source name.

Data Sources The **dsn** query returns a list of data sources available from the server. A single annotation provider (unique *PREFIX*) may present a number of annotation databases by using different data source names.

Scope: Reference and annotation servers.

Command: *dsn*

Format: *PREFIX/das/dsn*

Arguments: *none*

Return Document:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DASDSN SYSTEM "http://biodas.org/dtd/dasdsn.dtd">
<DASDSN>
  <DSN>
    <SOURCE id="id1" version="version"> source name 1 </SOURCE>
```

```

    <MAPMASTER> URL </MAPMASTER>
    <DESCRIPTION> descriptive text 1 </DESCRIPTION>
</DSN>
<DSN>
    <SOURCE id="id2" version="version"> source name 2 </SOURCE>
    <MAPMASTER> URL </MAPMASTER>
    <DESCRIPTION href="url"> descriptive text 2 </DESCRIPTION>
</DSN>
. . .
</DASDSN>

```

<!DOCTYPE> (required; one only) The doctype indicates which formal DTD specification to use. For the dsn query, the doctype DTD is “dasdsn.dtd”.

<DASDSN> (required; one only) The appropriate doctype and root tag is DASDSN.

<DSN> (required; one or more) There are one or more <DSN> tags, one for each data source. Each <DSN> contains one <SOURCE> tag, one <MAPMASTER> tag, and optionally one <DESCRIPTION> tag.

<SOURCE> (required; one per DSN tag) This tag indicates the symbolic name for a data source. The symbolic name to use for further requests can be found in the **id** (required) attribute. A source **version** attribute is optional, but strongly recommended. The tag body contains a human-readable label which may or may not be different from the ID.

<MAPMASTER> (required; one per DSN tag) This tag contains the URL (http://site.specific.prefix/das/data_src) that is being annotated by this data source. For an annotation server, this is the reference server which is being annotated. For a reference server this would echo its own URL.

<DESCRIPTION> (optional) This tag contains additional descriptive information about the data source. If an **href** (optional) attribute is present, the attribute contains a link to further human-readable information about the data source, such as its home page.

Entry Points The **entry_points** query returns the list of sequence entry points available and their sizes in base pairs.

Scope: Reference servers.

Command: *entry_points*

Format: PREFIX/das/DSN/entry_points

The **entry_points** command is not intended for transmitting sequence assembly, but rather to provide the top level entry points for browsers. See fetching the sequence assembly for the mechanism for retrieving assembly information.

Return Document:

```
<?xml version="1.0" standalone="no"?>
```

```

<!DOCTYPE DASEP SYSTEM "http://biodas.org/dtd/dasep.dtd">
<DASEP>
  <ENTRY_POINTS href="url" version="X.XX">
    <SEGMENT id="id1" size="length1"> descriptive text </SEGMENT>
    <SEGMENT id="id2" size="length2"> descriptive text </SEGMENT>
    <SEGMENT id="id3" size="length3"> descriptive text </SEGMENT>
    . . .
  </ENTRY_POINTS>
</DASEP>

```

<!DOCTYPE> (required; one only) The doctype indicates which formal DTD specification to use. For the entry_points query, the doctype DTD is “dasep.dtd”.

<DASEP> (required, one only) The appropriate doctype and root tag is DASEP.

<ENTRY_POINTS> (required, only one) There is a single <ENTRY_POINTS> tag. It has a **version** number (required) in the form “N.NN”. Whenever the sequence map changes, the version number should change as well. The **href** (required) attribute echoes the URL query that was used to fetch the current document.

<SEGMENT> (optional; zero or more) Each segment contains the attributes **id** and **size** (both required). The id is a unique identifier, which can be used as the reference ID in further requests to DAS. The size is the full length of the segment object. The body of the <SEGMENT> sections contains human-readable text (optional) for the purposes of display and selection.

DNA The **dna** query returns the DNA corresponding to the indicated segment.

Scope: Reference servers.

Command: *dna*

Format: PREFIX/das/DSN/dna?segment=RANGE [;segment=RANGE...]

Arguments:

segment (required; one or more) A reference sequence and optional range in the format: **ref:start,stop** or simply **ref**. Multiple segments are allowed. Here is an example of a request to fetch three non-overlapping segments:

```
dna?segment=chr1:1,1000;segment=chr1:5000,5200;segment=AC12345
```

Return Document:

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE DASDNA SYSTEM "http://biodas.org/dtd/dasdna.dtd">
<DASDNA>
  <SEQUENCE id="id" start="start" stop="stop" version="X.XX">
    <DNA length="NNNN">
      atttccttggcgtaaataagagtctcaatgagactctcagaagaaaattgataaatattat

```

```

    taatgatataataataatcttggtgatccgttctatctccagacgattttcctagtctcc
    ...
    gaaacaatgaacactattataacattttcagaaaatgtagtatttaagcgaaggtagtgc
    acatcaaggccgtcaaacggaaaaattttgcaagaatca
    </DNA>
  </SEQUENCE>
</DASDNA>

```

<!DOCTYPE> (required; one only) The doctype indicates which formal DTD specification to use. For the dna query, the doctype DTD is “dasdna.dtd”.

<DASDNA> (required; one only) The appropriate doctype and root tag is DASDNA.

<SEQUENCE> (required; one or more) There is one <SEQUENCES> tag per requested segment. It has the attributes **id**, which indicates the reference ID for this sequence, **start** and **stop**, which indicate the position of this segment within the reference sequence, and **version**, which provides the sequence map version number. All four attributes are required.

<DNA> (required; one per SEQUENCE) This tag surrounds the DNA data. It has the attribute **length** (required), which indicates the length of the DNA. The DNA is found in the body of the tag and is required. DNA will be lower-case and adhere to the IUPAC code conventions.

Summary Information The **types** query returns a summary of the annotation available for a segment of sequence.

Scope: Reference and annotation servers

Command: *types*

Format: PREFIX/das/DSN/types?segment=RANGE
 [;segment=RANGE...] [;type=TYPEPATTERN]

Arguments:

segment (required; one or more) A reference sequence and optional range in the format: **ref:start,stop** or simply **ref**. Multiple segments are allowed.

type (optional) One or more type IDs to be used for filtering annotations on the type field. If multiple type names are provided, the resulting list of features will be the logical OR of the list.

If one or more segment arguments are provided, the list of types returned is restricted to the indicated segments. If no segment argument is provided, then all feature types known to the source are returned.

Return Document:

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE DASTYPES SYSTEM "http://biодas.org/dtd/dastypes.dtd">
<DASTYPES>
  <GFF version="1.0" href="url">

```

```

<SEGMENT id="id" start="start" stop="stop" version="X.XX"
          label="label">
  <TYPE id="id1" method="method" category="category">
    Type Count 1 </TYPE>
  <TYPE id="id2" method="method" category="category">
    Type Count 2 </TYPE>
  . . .
</SEGMENT>
</GFF>
</DASTYPES>

```

<!DOCTYPE> (required; one only) The doctype indicates which formal DTD specification to use. For the types query, the doctype DTD is “dastypes.dtd”.

<DASTYPES> (required; one only) The appropriate doctype and root tag is DASTYPES.

<GFF> (required; one only) There is a single <GFF> tag. Its **version** (required) attribute indicates the current version of the XML form of the General Feature Format. The current version is 1.0. The **href** (required) attribute echoes the URL query that was used to fetch the current document.

<SEGMENT> (required; one or more) The <SEGMENT> tag, provides information on the reference segment . The **id**, **start** and **stop** attributes indicate the coordinate system of the segment. The **version** attribute indicates the current version of the sequence map. The id, start, stop, and version attributes are required. The optional **label** attribute supplies a human readable label for display purposes.

<TYPE> (required; one or more per SEGMENT) Each segment has one or more <TYPE> tags, which summarize the types of annotation available. The attributes are **id** (required), which is a unique id for the annotation type and can be used to retrieve further information from the annotation server (see Linking to a Feature), **method** (optional) which indicates the method (subtype) for a feature type and the **category** (optional) attribute, which provides functional grouping to related types. The tag contents (optional) is a count of the number of features of this type across the segment.

Annotations The **features** query returns the annotations across a segment of sequence.

Scope: Reference and annotation servers

Command: *features*

Format: PREFIX/das/DSN/features?segment=RANGE

 [;segment=RANGE...][;type=TYPEPATTERN]

 [;category=CATEGORYPATTERN]

Arguments:

segment (required; one or more) A reference sequence and optional range in the format: **ref:start,stop** or simply **ref**. Multiple segments are allowed.

type (zero or more) Zero or more type IDs to be used for filtering annotations on the type field. If multiple type names are provided, the resulting list of features will be the logical OR of the list.

category (zero or more) Zero or more category IDs to be used for filtering annotations by category. If multiple categories are provided, they are treated as the logical OR.

categorize (optional) Either "yes" or "no" (default). If "yes", then each annotation must include its functional category.

Annotation servers are only required to return annotations which are completely contained within the indicated segment. Servers may also return annotations which overlap the segment, but are not completely contained within them. Annotations must be returned using the coordinate system in which they were requested. For example, if a contig ID was used to specify the segment, then the annotation endpoints must use contig coordinates.

If multiple segment arguments are provided and they happen to overlap, then the annotation server may return the same annotation multiple times, possibly using different coordinate systems. It is the responsibility of the client to merge annotations based on the assembly.

Return Document:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DASGFF SYSTEM "http://biodas.org/dtd/dasgff.dtd">
<DASGFF>
  <GFF version="1.0" href="url">
    <SEGMENT id="id" start="start" stop="stop" version="X.XX"
      label="label">
      <FEATURE id="id" label="label">
        <TYPE id="id" category="category" reference="yes|no"
          subparts="yes|no" superparts="yes|no"> label </TYPE>
        <METHOD id="id"> method label </TYPE>
        <START> start </START>
        <END> end </END>
        <SCORE> [X.XX|-] </SCORE>
        <ORIENTATION> [0|-|+] </ORIENTATION>
        <PHASE> [0|1|2|-] </PHASE>
        <NOTE> note text </NOTE>
        <LINK href="url"> link text </LINK>
        <TARGET id="id" start="x" stop="y"> target name </TARGET>
        <GROUP id="hash" label="label" type="type">
          <NOTE> note text </NOTE>
          <LINK href="url"> link text </LINK>
        </GROUP>
      </FEATURE>
      ...
    </SEGMENT>
  </GFF>
</DASGFF>
```

The positions of all returned annotations are given relative to the indicated reference

sequence.

<!DOCTYPE> (required; one only) The doctype indicates which formal DTD specification to use. For the feature query, the doctype DTD is “dasgff.dtd”.

<DASGFF> (required; one only) The appropriate doctype and root tag is DASGFF.

<GFF> (required; one only) There is a single <GFF> tag. Its **version** (required) attribute indicates the current version of the XML form of the General Feature Format. The current version is (arbitrarily) 1.0. The **href** (required) attribute echoes the URL query that was used to fetch the current document.

<SEGMENT> (required; one or more) The <SEGMENT> tag, provides information on the reference segment coordinate system. The **id**, **start** and **stop** attributes indicate the position of the segment. The **version** attribute indicates the current version of the sequence map. The **id**, **start**, **stop**, and **version** attributes are required. The optional **label** attribute provides a human readable label for display purposes.

<FEATURE> (optional; zero or more per SEGMENT) Each <FEATURE> tag provides information on one annotation. The **id** attribute (required) is a unique identifier for the feature. It can be used as a reference point for further navigation. The **label** attribute (optional) is a suggested label to display for the feature. If not present, the **id** attribute can be used instead.

<TYPE> (required; one per FEATURE) Each feature has just one <TYPE> field, which indicates the type of the annotation. The attributes are **id** (optional), which is a unique id for the annotation type and can be used to retrieve further information from the annotation server (see Linking to a Feature), and the **category** (optional) attribute, which provides functional grouping to related types. The reference server’s annotations can consist of additional overlapping landmarks (parents, children, and neighbors), which should be marked “yes” in the third attribute **reference** (optional, defaults to “no”) to indicate that the feature is a structural landmark within the map (this feature can be annotated). The tag contents (optional) is a human readable label for display purposes. If a **reference** annotation has either or both of the optional attributes **subparts = “yes”** and **superparts = “yes”** then in addition to being useable as a reference sequence, the annotation contains subparts that themselves can act as reference features. This can be used to reconstruct reference server’s assembly (see the Entry Points document).

<METHOD> (required; one per FEATURE) Each feature has one <METHOD> field, which gives the method used to identify the feature. The **id** (optional) tag can be used to retrieve further information from the server. The tag contents (optional) is a human readable label.

<START>, <END> (required; one apiece per FEATURE) These tags indicate the start and end of the feature in the coordinate system of the reference sequence given in the <SEGMENT> tag. The relationship between the feature start and stop positions and the segment start and stop is that the two spans are guaranteed to overlap.

<SCORE> (required; one per FEATURE) This is a floating point number indicating the “score” of the method used to find the current feature. The number can only be understood in the context of information retrieved from the server by linking to the method. If this field is inapplicable, the contents of the tag can be replaced with a “-” symbol.

- <**ORIENTATION**> (**required; one per FEATURE**) This tag indicates the orientation of the feature relative to the direction of transcription. It may be **0** for features that are unrelated to transcription, **+**, for features that are on the sense strand, and **-**, for features on the antisense strand.
- <**PHASE**> (**required; one per FEATURE**) This tag indicates the position of the feature relative to open reading frame, if any. It may be one of the integers **0**, **1** or **2**, corresponding to each of the three reading frames, or **-** if the feature is unrelated to a reading frame.
- <**NOTE**> (**optional; zero or more per FEATURE**) A human-readable note in plain text format.
- <**LINK**> (**optional; zero or more per FEATURE**) A link to a web page somewhere that provides more information about this feature. The **href** (required) attribute provides the URL target for the link. The link text is an optional human readable label for display purposes.
- <**TARGET**> (**optional; zero or more per FEATURE**) The target sequence in a sequence similarity match. The **id** attribute provides the reference ID for the target sequence, and the **start** and **stop** attributes indicate the segment that matched across the target sequence. All three attributes are required. More information on the target can be retrieved by linking back to the annotation server. See Linking to a Feature. (Earlier versions of this specification required the TARGET tag to be in the GROUP section.)
- <**GROUP**> (**optional; if present, one per FEATURE**) The <GROUP> section is an oddity, as it is derived from an overloaded field in the GFF flat file format. It provides a unique “group” ID that indicates when certain features are related to each other. The canonical example is the CDS, exons and introns of a transcribed gene, which logically belong together. The group **id** (required) tag provides an identifier that should be used by the client to group features together visually. Unlike other IDs in this protocol, the group ID cannot be used as a database handle to retrieve further information about the group. Such information can, however, be provided within <GROUP> section, which may contain up to three optional tags. The **label** attribute (optional) provides a human-readable string that can be used in graphical representations to label the glyph. The **type** attribute (optional) provides a type ID for the group as a whole, for example “transcript”. This ID can be used as a key into the stylesheet to select the glyph and graphical characteristics for the group as a whole.
- <**NOTE**> (**optional; if present, one per GROUP**) A human-readable note in plain text format.
- <**LINK**> (**optional; if present, one per GROUP**) A link to a web page somewhere that provides more information about this group. The **href** (required) attribute provides the URL target for the link. The link text is an optional human readable label for display purposes.

Linking to a Feature The **link** query can be issued in order to retrieve further human-readable information about an annotation. It is best to pass this URL directly to a browser, as the type of the returned data is not specified (it will typically be an HTML file, but any MIME format is allowed).

Scope: Annotation Servers

Command: *link*

Format: PREFIX/das/DSN/link?field=TAG;id=ID

Arguments:

field (required) The field to fetch further information on. Options are:

- **feature** – the feature itself
- **type** – the feature type
- **method** – the feature method
- **category** – the feature category
- **target** – the target, applicable to sequence similarities only

id (required) The ID of the indicated annotation field.

Returns: A web page.

Stylesheet The **stylesheet** query can be issued to an annotation server in order to retrieve the server's recommendations on formatting annotations retrieved from it. These recommendations are not normative. A viewer is free to use any display format it chooses.

Scope: Annotation Servers

Command: *stylesheet*

Format: PREFIX/das/DSN/stylesheet

Arguments: None.

Return Document:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DASSTYLE SYSTEM "http://biодas.org/dtd/dasstyle.dtd">
<DASSTYLE>
  <STYLESHEET version="X.XX">
    <CATEGORY id="default">
      <TYPE id="default">
        <GLYPH zoom="high"> <ID>
          <ATTR>value</ATTR>
          <ATTR>value</ATTR>
          ...
        </ID> </GLYPH>
      </TYPE>
    </CATEGORY>
    <CATEGORY id="category1">
      <TYPE id="default">
        <GLYPH zoom="medium"> <ID>
          <ATTR>value</ATTR>
        </ID> </GLYPH>
      </TYPE>
      <TYPE id="type1">
        <GLYPH zoom="low"> <ID>
          <ATTR>value</ATTR>
```

```

    ...
    </ID> </GLYPH>
  </TYPE>
  ...
</CATEGORY>
<CATEOGRY id="group">
  <TYPE id="group_id1">
    <GLYPH> <ID>
      <ATTR>value</ATTR>
    </ID> </GLYPH>
  </TYPE>
</CATEGORY>
<CATEGORY id="category2">
  ...
</CATEGORY>
  ...
</STYLESHEET>
</DASSTYLE>

```

<!DOCTYPE> (required; one only) The doctype indicates which formal DTD specification to use. For the stylesheet query, the doctype DTD is “dasstyle.dtd”.

<DASSTYLE> (required; one only) The appropriate doctype and root tag is DASSTYLE.

<STYLESHEET> (required; one only) There is a single <STYLESHEET> tag. Its **version** (required) attribute indicates the current version of the stylesheet, and can be used for caching purposes.

<CATEGORY> (required; one or more) There are one or more <CATEGORY> tags, each providing information on the display of a high-level feature category. The **id** (required) tag uniquely names the category. A special name is “default”, which tells the annotation viewer what format to use for categories that are not otherwise specified in the stylesheet. Another special id, “group”, identifies which style to use for a particular group of features.

<TYPE> (required; one or more per CATEGORY) There are one or more <TYPE> tags per <CATEGORY>, each providing display suggestions for one type of annotation. The **id** (required) uniquely identifies the type. A special id is “default”, which, if present, identifies a default style for the enclosing category.

<GLYPH> (required; one per TYPE) There is a single <GLYPH> tag per <TYPE>. It provides information on what glyph (graphical widget) to use to display the indicated annotation type. The optional **zoom** attribute implements a simple form of semantic zooming, and allows the client to select the glyph and its attributes based on the current zoom level. Possible values are “high”, “medium” or “low”. If multiple <GLYPH> tags are present, this attribute **must** be present in order to select among them. A “high” zoom means that there are fewer base pairs per pixel (high magnification), “low” zoom shows more base pairs, and “medium” is intermediate. It is left to the client to determine the boundaries for “high”, “medium” and “low” since it is a function of graphics rendering.

<ID> (required; one per GLYPH) The ID tag is a particular instance of one of the recognized glyph from the glyph types list. (for example <BOX> is an instance of <ID>)

<ATTR> (optional; one or more per ID) The recognized ATTR (attributes) are determined by which glyph ID is specified.

Glyphs and Groups Glyphs and their attributes are typically applied to individual features. However, they can be applied to entire groups as well (via the <GROUP> type attribute). In this case, the glyph will apply to the connecting regions between the components of the group.

For example, to indicate that the exons in a “transcript” group should be drawn with a yellow box, that the utrs should be drawn with a blue box, and that the connections between exons should be drawn with a hat-shaped line:

```
<CATEGORY id="Transcription">
  <TYPE id="exon">
    <GLYPH>
      <BOX>
        <BGCOLOR>yellow</BGCOLOR>
      </BOX>
    </GLYPH>
  </TYPE>

  <TYPE id="utr">
    <GLYPH>
      <BOX>
        <BGCOLOR>blue</BGCOLOR>
      </BOX>
    </GLYPH>
  </TYPE>
</CATEGORY>

<CATEGORY id="group">
<TYPE id="transcript">
  <GLYPH>
    <LINE>
      <STYLE>hat</STYLE>
      <FGCOLOR>black</FGCOLOR>
    </LINE>
  </GLYPH>
</TYPE>
...
```

A.3 Fetching Sequence Assemblies

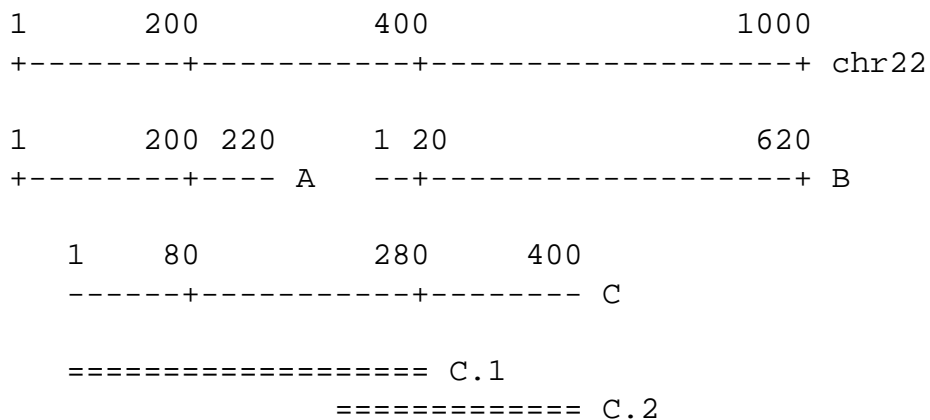
Reference servers, but not annotation servers, must represent and serve genome assemblies.

The components of an assembly are treated as a set of features with a type category attribute of "component" and a reference attribute of "yes". Intermediate components of the assembly will also have a subparts attribute of "yes" and/or *superparts* attribute of "yes". Components that are the parents of the reference sequence in the assembly have a category attribute of "supercomponent".

Moving Downwards in an Assembly

For those components that have subparts, the start and end of the feature give the feature's position in the requested segment's coordinate system, and the start and end of the <TARGET> element gives the feature's position in its native coordinates.

For example:



A request for this assembly will look like the following:

<http://www.wormbase.org/db/das/elegans/features?segment=chr22:1,1000;category=component>

The reference server will return the following (abbreviated) document:

```
<SEGMENT id="chr22" start="1" stop="1000">

<FEATURE id="chr22">
  <START>1</START>
  <STOP>1000</STOP>
  <TYPE id="Contig" category="component" reference="yes"
        superparts="no" subparts="yes">chr 22</TYPE>
  <TARGET id="chr22" start="1" stop="1000">Contig A</TARGET>
  ...
</FEATURE>
```

```

<FEATURE id="Contig:A">
  <START>1</START>
  <STOP>200</STOP>
  <TYPE id="Contig" category="component" reference="yes"
        superparts="yes" subparts="no"> a contig</TYPE>
  <TARGET id="A" start="1" stop="200">Contig A</TARGET>
  ...
</FEATURE>
<FEATURE id="Contig:B">
  <START>400</START>
  <STOP>1000</STOP>
  <TYPE id="Contig" category="component" reference="yes"
        superparts="yes" subparts="no"> a contig</TYPE>
  <TARGET id="B" start="20" stop="620">Contig A</TARGET>
  ...
</FEATURE>
<FEATURE id="Contig:C">
  <START>200</START>
  <STOP>400</STOP>
  <TYPE id="Contig" category="component" reference="yes"
        superparts="yes" subparts="no"> a contig</TYPE>
  <TARGET id="C" start="80" stop="280">Contig A</TARGET>
  ...
</FEATURE>
</SEGMENT>

```

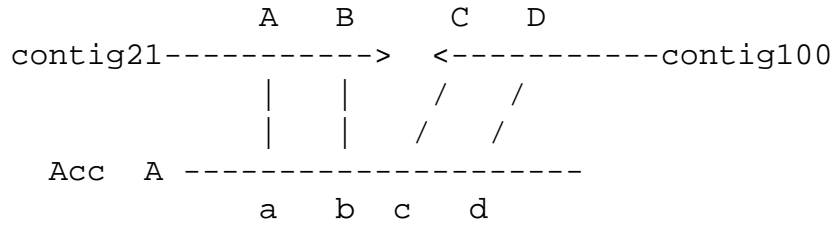
Notice that contig C is marked as having subparts. This is an indication to the client that it should emit a features request that includes segment C:80,280 in order to discover its components (C.1 and C.2).

Notice also that chr22 appears as a component of itself with the attribute **superparts="no"** and **subparts="yes"**. This is a side effect of providing information about the component parent.

Moving Upwards in an Assembly

It is desirable for a client to fetch the parent of a segment, so as to accomodate the situation in which the user enters the browser at a contig or sequenced clone, and wants to “zoom out.”

This situation is complicated by rough draft issues, in which a single rough draft sequence segment may have multiple parents, and some sections of the segment may not belong in the assembly at all. For example:



Here, the segment “Acc A” contains two fragments, one of which is located on contig21 and the other on contig100.

To retrieve this information, the client requests the category supercomponent. For segments that are in the middle of the assembly, one or more assembly parents will be returned in addition to subcomponents. The parent <START>, <STOP> and <ORIENTATION> tags are presented in the coordinate system of the requested segment, as always. The start and stop attributes of the <TARGET> tag, denote the corresponding segment in the coordinate system of the parent. As always, start is less than stop, for both the feature and the target.

```
<SEGMENT id="Acc A" start="1" stop="1000">
  <FEATURE id="contig21_goldenpath_map">
    <START>a</START>
    <STOP>b</STOP>
    <ORIENTATION>+</ORIENTATION>
    <TYPE id="Contig" category="supercomponent" reference="yes"
      superparts="yes" subparts="yes">a contig</TYPE>
    <TARGET id="contig21" start="A" stop="B"></TARGET>
    ...
  </FEATURE>
  <FEATURE id="contig100_goldenpath_map">
    <START>c</START>
    <STOP>d</STOP>
    <ORIENTATION>-</ORIENTATION>
    <TYPE id="Contig" category="supercomponent" reference="yes"
      superparts="yes" subparts="yes">a contig</TYPE>
    <TARGET id="contig100" start="D" stop="C"></TARGET>
    ...
  </FEATURE>
</SEGMENT>
```

To continue following the parents upward in the assembly, the client will issue further features requests for the target IDs, in this case “contig21” and “contig100”. In the general case, following parents will project the requested segment onto a discontinuous set of regions, potentially on different chromosomes. The client may wish to alert the user and refuse to proceed further when it encounters a segment with multiple parents.

A.4 Feature Types and Categories

Features are associated with a graphical representation, a glyph, by the type id and category. The “category” is designed to describe broad annotation types (i.e. transcribed). The “id” refers to a more specific instance within its category (i.e. exon). An annotation provider is free to use any categories and types that they feel are appropriate to their annotations. If an annotation is described by a type id and category which are not defined by the accompanying stylesheet, programmatic defaults are used. The following is a list of generic feature categories and specific feature types within them.

Feature Types and Categories

- **component** – indicate that the feature is a child component of the reference sequence in the current assembly. When combined with **reference = “yes”** attribute this indicates that the feature can be used as a reference point to retrieve subfeatures contained within it (including subcomponents).
- **supercomponent** – indicate that the feature is the parent of the reference sequence in the current assembly. When combined with **reference = “yes”** attribute this indicates that the feature can be used as a reference point to retrieve features that completely contain the selected range of the reference sequence.
- **translation** – features that relate to regions of the sequence that are translated into proteins. (ex: stop, ATG, CDS, 5’UTR, 3’UTR, misc_translated)
- **transcription** – features that relate to regions of the sequence that are transcribed into RNA. (ex: exon, intron, tRNA, mRNA, ncRNA, 5’Cap, PolyA, Splice5, Splice3, misc_transcribed)
- **variation** – features that relate to regions of the sequence that are polymorphic (ex: insertion, deletion, substitution, misc_variation)
- **structural** – features that relate to mapping, sequencing, and assembly, as well as various landmarks that carry no intrinsic biological information. (ex: clone, primer_left, primer_right, oligo, misc_structural)
- **similarity** – areas of sequence that are similar to other sequences. Similarity features should have a <METHOD> tag that indicates the algorithm used for the sequence comparison, and a <TARGET> tag in the <GROUP> field that indicates the target of the match. (ex: NN (nucleotide to nucleotide), NP (nucleotide to peptide), PN (peptide to nucleotide), PP (peptide to peptide), misc_homology)
- **repeat** – areas that contain repetitive DNA. This is used for both low complexity regions and for more biologically interesting features such as transposon insertion sites. (ex: microsatellite, inverted, tandem, transposable_element, LINE, misc_repeat)

- **experimental** – used to flag areas where there is interesting experimental data of one sort or another. (ex: knockout, expression_tag, microarrayed, RNAi_result, transgenic, mutant, misc_experimental)

An annotation provider is free to use any category and type name. It is recommended, but not required, that the <GROUP> section contain <LINK> and/or <NOTE> tags that provide further information on the all features.

A.5 Glyphs

This section defines a set of generic “glyphs” that can be used by sequence display programs to display the position of features on a sequence map. The annotation server may use these glyphs to send display suggestions to the viewer via the stylesheet document. Each glyph has a set of possible attributes associated with it. Attributes come in a few basic data types:

- INT (an integer)
- STRING (a text string)
- COLOR (specified as the #RRGGBB format commonly used for HTML or as one of the 16 IBM VGA colors)
- BOOL (either “yes” or “no”)
- FONT (any of the fonts recognized by Web browsers)
- FONT_STYLE (“bold”, “italic”, or “underline”).
- LINE_STYLE (“hat”, “solid”, “dashed”).

Some attributes are shared by all glyphs. Others are glyph-specific. The following are attributes shared in common:

- **BGCOLOR** – (*COLOR*) The background color of the glyph. For hollow glyphs, such as boxes, this is the color of the interior of the box. For solid glyphs, such as text, this is ignored.
- **BUMP** – (*BOOL*) Determines if the viewer should “bump” intersecting glyphs so that they do not overlap.
- **FGCOLOR** – (*COLOR*) The foreground color of the glyph. This is the line and outline color for graphical glyphs and the font color for text glyphs.
- **HEIGHT** – (*INT*) The height of the glyph. The height is orthogonal to the axis that defines the extent of the feature on the sequence map. For a text glyph, this is equivalent to the **FONTSIZE** attribute.
- **LABEL** – (*BOOL*) Whether the glyph should be labeled with its name, as dictated by the <FEATURE> label attribute in the DASGFF document.

ARROW – A double-headed arrow with an axis either orthogonal or parallel to the sequence map.

- **PARALLEL** – (*BOOL*) Arrows run either parallel or orthogonal to the sequence axis.

ANCHORED_ARROW – A arrow that has an arrowhead at one end and an “anchor” (typically a diamond or line) at the other. The arrow points in the direction indicated by the `<ORIENTATION>` tag.

- **PARALLEL** – (*BOOL*) Arrows run either parallel or orthogonal to the sequence axis.

BOX – A rectangular box

- **LINEWIDTH** – (*INT*) Width of the glyph outline.

CROSS – A cross “+”. Commonly used for point mutations and other point-like features. (*no glyph-specific attributes*)

DOT – A small circle. Commonly used for point mutations and other point-like features. (*no glyph-specific attributes*)

EX – “X” marks the spot. Commonly used for point mutations and other point-like features. (*no glyph-specific attributes*)

HIDDEN – A feature that is invisible, intended to support semantic zooming schemes in which a feature is hidden at particular zooms. (*no attributes*)

LINE – A line. Lines are equivalent to arrows with both the `NORTHEAST` and `SOUTHWEST` attributes set to “no”.

- **STYLE** – (*LINE_STYLE*) A style of “hat” draws an inverted V (commonly used for introns). A type of “solid” draws a horizontal solid line in the indicated color. A type of “dashed” draws a dashed horizontal line in the indicated color.

SPAN – A spanning region, the recommended representation is a horizontal line with vertical lines at each end. (*no glyph-specific attributes*)

TEXT – A bit of text.

- **FONT** – (*FONT*) The font.
- **FONTSIZE** – (*INT*) The font size.

- **STRING** – (*STRING*) The text to render.
- **STYLE** – (*FONT_STYLE*) The style in which to render this glyph. Multiple **FONT_STYLE** attributes may be present.

TOOMANY – Too many features than can be shown. Recommended for use in consolidating sequence homology hits. The recommended visual presentation is a set of overlapping boxes.

- **LINEWIDTH** – (*INT*) Width of the glyph outline.

TRIANGLE – A triangle. Commonly used for point mutations and other point-like features.

- **LINEWIDTH** – (*INT*) Width of the glyph outline.