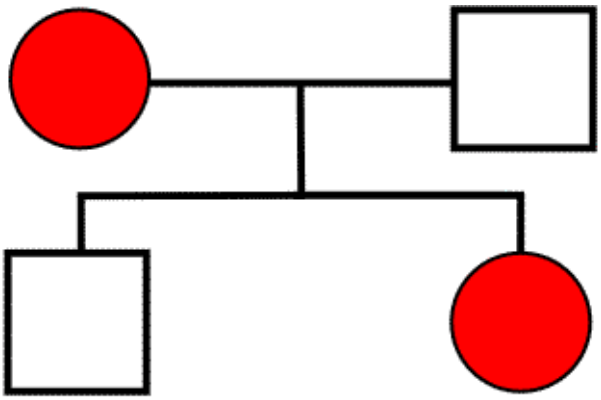


DataVista Pedigree

Programmer's Guide

DataVista Pedigree™
Version 1.0

Visualize Inc
Get The Picture

Title and Publication Revision

DataVista Pedigree Programmer's Guide, Version 1.0, February 2003, Rev. A

License, Copyright, and Trademark Information

Contact Information

Visualize, Inc.

3333 East Camelback Road, Suite 150

Phoenix, AZ 85018

Voice: (602) 861-0999

Fax: (602) 861-1650

E-mail: support@visualize.com

URL: www.visualize.com

Table of Contents

4 JavaScript Interface

5 Application Component Installation

6 User's Guide

7 XML Reference

1 Introduction

DataVista Pedigree Overview

DataVista Pedigree™ (DVP) is a Java™ applet and application component that supports the display of pedigrees and genetic data. Its XML-based configuration format and powerful API enable its effective deployment in a wide range of applications.

DVP supports the following features:

- Ability to be deployed as either a Java applet within a web application or application component within either Swing or AWT-based applications.
- XML-based configuration that enables the customization of many aspects of DVP.
- Functionality that supports user interaction including zoom, pan, and selection.
- An interface that displays pedigrees ranging from a few to several thousand subjects.
- Display of both genotypic and phenotypic data as properties, adornments, and haplotype visualizations.
- Data source support for XML, JDBC, CSV, or objects conforming to an open data source API.

Installing and configuring DVP involves the following tasks:

- Installing DVP to deploy within your application.
- Configuring DVP's licensing, interface, pedigree appearance, and data.

Using the DataVista Pedigree Programmer's Guide

The DataVista Pedigree Programmer's Guide provides information for configuring and using DVP. This document organizes information as follows:

- Chapter 1: Introduction–provides an overview of DVP.
- Chapter 2: Applet Installation–explains how to install DVP to be deployed within a web application.
- Chapter 3: Applet Configuration–describes how to configure DVP.

- Chapter 4: JavaScript Interface–explaines how to use JavaScript to configure the DVP interface.
- Chapter 5: Application Component Installation–explains how to install DVP to be deployed within a Swing or AWT-based application.
- Chapter 6: User's Guide–provides instructions for using DVP.
- Chapter 7: XML Reference–lists all XML elements for the DVP configuration file.

2 Applet Installation

Applet Installation Overview

You can deploy DVP as a Java applet embedded in a standard html-formatted web page and viewable on any modern browser that has Sun's Java Plug-In installed (such as Netscape® and Microsoft® Internet Explorer). This chapter includes information for installing DVP as a Java applet within a web application.

Installation includes the following tasks:

- Copying files from the DVP distribution package to a location on your web server.
- Configuring an .html file to allow your web application to deploy DVP.

Copying Distribution Files

After extracting the distribution files from the DVP distribution package, copy the following file from the jars directory to a location on your web server so that web pages deploying DVP may reference it:

- dvpedigree-applet-jdk12-1.0.x.jar

Note: In this document, "x" in file names represents the latest version. When locating and working with files, you will replace "x" with your version, such as "dvpedigree-applet-jdk12-1.0.5.jar."

After you have configured the applet, you will need to ensure that configuration files also reside in this location.

For more information, see: Configuring a Web Application to Deploy DataVista Pedigree and Applet Configuration.

Configuring a Web Application to Deploy DataVista Pedigree

When deploying DVP as an applet within a web application, you insert information about DVP within an <object> tag in your web application's .html file. In addition to code that allows DVP to run (such as the latest Java plug-in), the <object> tag should include references to the following files:

- DVP .jar file
- DVP configuration file

The following example shows the code to insert into an .html file:

```
<OBJECT
    classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
    WIDTH = 800 HEIGHT = 600 NAME = "dvped"
    codebase="http://java.sun.com/products/plugin/autodl/jinstall-
1_4-win.cab#Version=1,4,0,0">
    <PARAM NAME = CODE VALUE =
"com.visualizeinc.datavistapedigree.applet.DataVistaPedigree" >

<PARAM NAME = CODEBASE VALUE = "." >

<PARAM NAME = ARCHIVE VALUE = "dvpedigree-applet-jdk12-1.0.x.jar" >

<PARAM NAME = NAME VALUE = "dvped" >

<PARAM NAME = MAYSCRIPT VALUE = true >

    <PARAM NAME="type" VALUE="application/x-java-
applet;version=1.4">
    <PARAM NAME="scriptable" VALUE="true">
    <PARAM NAME = "ConfigFilename" VALUE ="ConfigFilename.dvped">

    <COMMENT>
 <EMBED
            type="application/x-java-applet;version=1.4"
            CODE =
"com.visualizeinc.datavistapedigree.applet.DataVistaPedigree"
            CODEBASE = "."
            ARCHIVE = "dvpedigree-applet-jdk12-1.0.x.jar"
            NAME = "dvped"
            WIDTH = 800
            HEIGHT = 600
            MAYSCRIPT = true
            ConfigFilename = "examplea.dvped"
```

Notice that this is where you reference the xml configuration file. For more information, see Applet Configuration.

```
     scriptable=true
     pluginspage="http://java.sun.com/products/plugin/
index.html#download">
         <NOEMBED>

    </NOEMBED>
 </EMBED>
    </COMMENT>
</OBJECT>
```

Note: You can learn more information about the Java Plug-in at the following web site: http://java.sun.com/products/plugin/. DVP supports Java 1.2 and later.

For more information about the DVP configuration file and pedigree data files, see Applet Configuration.

3 Applet Configuration

Applet Configuration Overview

DVP's XML-based configuration enables the customization of all aspects of its interaction and display.

The configuration XML involves four main blocks:

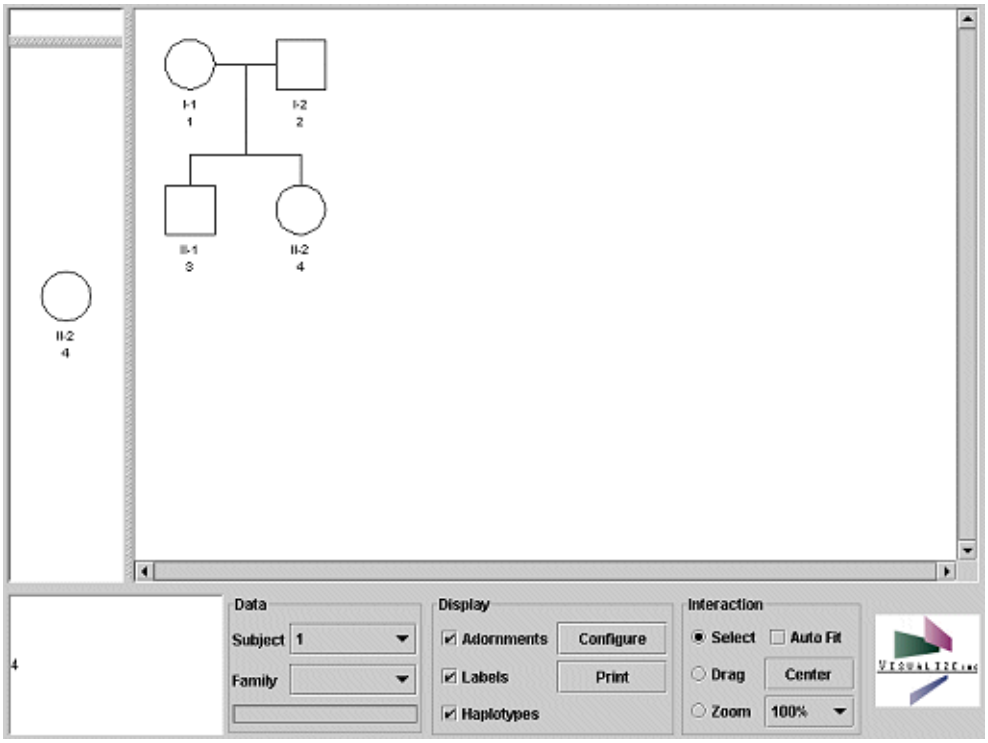
- <licensing>–details licensing information for DVP.
- <options>–defines certain aspects of DVP behavior.
- <viewer>–defines pedigree appearance.
- <dataset>–defines the pedigree data, including the family structure, phenotypic, and genotypic data.

You insert configuration XML within these top-level XML elements in your configuration file. You then save the file with a .dvped extension and reference it in your application's .html page (for more information, see Configuring a Web Application to Deploy DataVista Pedigree).

❖ **Example:** The following code and figure shows a basic pedigree. You can view the BasicExample applet and associated configuration file (BasicExample.html and BasicExample.dvped) in the distribution package Examples directory.

```
<datavistapedigree>
    <licensing>
        <keyword>Visualize</keyword>
        <license>ABCDEFGHIJKLMNOPQRST</license>
    </licensing>
    <options>
        <handlers>
            <handler type="hyperlink">
            </handler>
        </handlers>
    </options>
    <viewer>
        <pedigree-renderer>
            <subject>
                <annotations>
                    <annotation centered="true" type="symbol">
```

```
                        </annotation>
                        <annotation type="index">
                        </annotation>
                        <annotation type="label">
                        </annotation>
                        <annotation type="property">
                        </annotation>
                    </annotations>
                </subject>
                </pedigree-renderer>
            </viewer>
            <dataset>
                <pedigree>
                    <subject id="1">
                        <label>"1"</label>
                        <gender>female</gender>
                    </subject>
                    <subject id="2">
                        <label>"2"</label>
                        <gender>male</gender>
                    </subject>
                    <subject id="3">
                        <label>"3"</label>
                        <parents>1 2</parents>
                        <gender>male</gender>
                    </subject>
                    <subject id="4">
                        <label>"4"</label>
                        <parents>1 2</parents>
                        <gender>female</gender>
                    </subject>
                </pedigree>
            </dataset>
        </datavistapedigree>
```



The following information describes how to configure the elements within each main block.

Configuring DataVista Pedigree Licensing

You configure DVP's licensing information within the Licensing block. The <licensing> element enables DVP to function in your application.

The following code shows how you would configure licensing information:

```
<licensing>
    <keyword>YourCompany</keyword>
```

Type your keyword, such as "Visualize". An evaluation license, where the keyword is "Evaluation," may have been provided to you for evaluation purposes. A non-evaluation license is provided when arrangements of a licensing agreement are final.

```
    <license>LICENSECODE</license>
```

Type your license code, such as "`ABCDEFGHIJKLMNOPQRST`".

```
    <servers>
    server1
    server2
    server3
    </servers>
```

If applicable, type the servers from where DVP may be hosted.

```
    <expiration>"dd-MMM-yyyy"</expiration>
```

If applicable, type the expiration date of your license, such as "08-JAN-2003".

```
</licensing>
```

Make sure to input the licensing information carefully. Entries are case-sensitive. DVP will interpret typographical errors as though the license were invalid. An invalid license will prevent the program from running and will result in an error message being output to the appropriate console or log file.

If the license is determined to be valid for the information contained within the <licensing> block, but the actual server is not listed in the <servers> tag, then the license will be rejected, and the name of the server that should be included in the <servers> tag will be output. For example, DVP would output the following message:

```
DataVista Pedigree 1.0.*
2003-01-15 14:49:13
Copyright (c) 1996-2003 VI/Visualize, Inc.
See http://www.visualize.com for more details.
[SP00709] License is invalid
License based on server: [192.168.127.163, server]
```

Configuring Additional DataVista Pedigree Behavior

DVP already supports three main forms of interaction: selecting, panning, and zooming. Through interacting with DVP, users can select subjects, pan through a pedigree when it does not fit within the display area, and zoom into regions of interest.

You can configure additional applet behavior within an Options block. The <options> element allows you to configure the “handlers” associated with the applet. A handler is code that determines the behavior of the applet when a subject is right-clicked and the desired handler selected. A typical handler would be one that causes the browser to jump to a web page related to the selected subject.

The following code shows an <options> configuration:

```
<options>
    <handlers>
       <handler type="hyperlink">
       </handler>
    </handlers>
</options>
```

Configuring Pedigree Appearance

You configure pedigree appearance within the Viewer block. The <viewer> element defines the display of subject data, including annotations, symbols, adornments, labels, indexes, alleles, properties, and haplotypes.

Defining Annotations

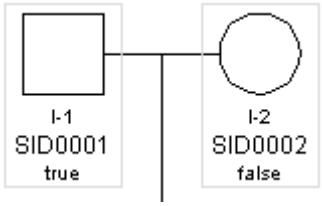
Annotations are defined within the <subject> tag, which is subordinate to <pedigree-renderer>. Each annotation is defined by an <annotation> tag, which is given with an associated type.

The following is typical XML for defining the subject appearance:

```
<viewer>
    <pedigree-renderer>
       <subject>
          <annotations>
             <annotation centered="true" type="symbol">
             . . .
```

```
                </annotation>
                <annotation type="index">
                </annotation>
                <annotation type="label">
                . . .
                </annotation>
                <annotation type="property">
                . . .
                </annotation>
            </annotations>
        </subject>
        </pedigree-renderer>
</viewer>
```

This XML defines a subject to have four annotations: symbol, index, label, and property. The subject will be centered on the symbol annotation. By default, a subject is centered on the first annotation.



Configuring Symbol Appearance

A symbol is displayed as either a square, circle, or diamond. Each symbol corresponds to a subject that is male, female or of unknown gender respectively. Each symbol may display one or more adornments that correspond to the values of properties for a given subject.

Configuring Adornments

Adornments are configured within the <adornments> tag, which is contained within the <annotation> tag that defines the symbol. Each adornment is defined by a tag whose name is the location of the adornment within the symbol.

Table 3-1 Adornment locations within symbols

Position	Appearance
full	
left	
right	
top	
bottom	

Table 3-1 Adornment locations within symbols

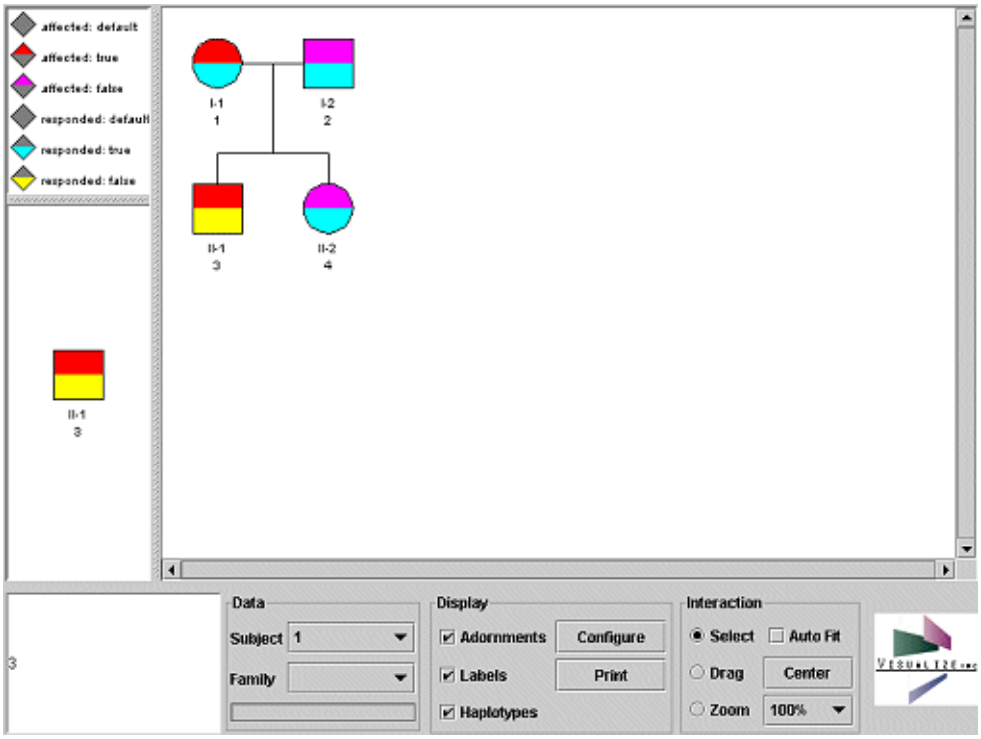
Position	Appearance
topleft	
topright	
bottomleft	
bottomright	

Within each adornment tag, such as <full>, is a <property-name> tag and a number of <color> tags. The <property-name> tag defines the property whose value will be employed in determining the adornment color. The <color> tag maps from the property value to a color. Each <color> tag contains a red, green, and blue triplet in the range of 0-255.

The following XML demonstrates how to display two adornments within a symbol. The first adornment is located in the top section of the symbol, and its color is a function of the “affected” property. When a subject has a value of “true” for the “affected” property, the color will be red, corresponding to a red, green, and blue triplet of 255 0 0. If the “affected” property is not defined, then the color will be grey, corresponding to a triplet of 127 127 127. The bottom adornment is defined similarly.

❖ **Example:** The following code and figure shows adornments added to a basic pedigree. You can view the AdornmentsExample applet and associated configuration file (AdornmentsExample.html and AdornmentsExample.dvped) in the distribution package Examples directory. For more information about the <properties> element added to the subject data, see Configuring Properties Data from an XML Datasource.

```
<annotation type="symbol">
    <adornments>
        <top>
            <property-name>affected</property-name>
            <color value="true">255 0 0</color>
            <color value=”false”>255 0 255</color>
            <color>127 127 127</color>
        </top>
        <bottom>
            <property-name>responded</property-name>
            <color value="true">0 255 255</color>
            <color value="false">255 255 0</color>
            <color>127 127 127</color>
        </bottom>
    </adornments>
</annotation>
```



Configuring Other Symbol Properties

In addition to adornments, you may also define other properties of the symbol annotation.

Table 3-2 Symbol annotation elements

Element	Values	Description
<insets>	integer (top) integer (left) integer (bottom) integer (right)	Four numbers defining the size of the padding around the symbol
<size>	Integer	The size of the symbol
<background-color>	red green blue triplet	The color displayed in the background of the symbol
<outline-color>	red green blue triplet	The color of the symbol outline
<legend-font>	<font> subelements	The font to use in the legend display
<legend-color>	red green blue triplet	The color of the legend text
<legend-insets>	integer (top) integer (left) integer (bottom) integer (right)	Four numbers defining the size of the padding around the legend entry

Table 3-3 Font elements

Element	Values	Description
<family>	Font family	The name of the font family
<size>	Integer	The point size of the font
<style>	italic | bold | plain	The font style

Configuring Label Appearance

Each subject has an associated label which may be displayed using the label annotation. The following XML defines a label annotation that will be displayed in a 12 point font:

```
<annotation type="label">
    <font><size>12</size></font>
</annotation>
```

Table 3-4 Label annotation elements

Element	Values	Description
<font>	<font> subelements	The font to be used in displaying the label text
<color>	red green blue triplet	The color of the text
<virtual-color>	red green blue triplet	The color of the text when the subject is virtual
<insets>	integer (top) integer (left) integer (bottom) integer (right)	The padding to surround this annotation
<background-color>	red green blue triplet	The background color
<outline-color>	red green blue triplet	The outline color
<draw-background>	Boolean	Whether the background should be colored
<draw-outline>	Boolean	Whether an outline should be drawn around the label

Configuring Index Appearance

Each subject has an associated generational index which may be displayed using the index annotation. The following XML defines an index annotation that will be displayed in a 12 point font:

```
<annotation type="index">
    <font><size>12</size></font>
</annotation>
```

Table 3-5 Index annotation elements

Element	Values	Description
<print-generation>	Boolean	Whether the generation index will be displayed
<print-index>	Boolean	Whether the individual index will be displayed. The individual index is the position of the subject from left to right.
<font>	<font> subelements	The font to be used in displaying the label text

Table 3-5 Index annotation elements

Element	Values	Description
<color>	red green blue triplet	The color of the text
<virtual-color>	red green blue triplet	The color of the text when the subject is virtual
<insets>	integer (top) integer (left) integer (bottom) integer (right)	The padding to surround this annotation
<background-color>	red green blue triplet	The background color
<outline-color>	red green blue triplet	The outline color
<draw-background>	Boolean	Whether the background should be colored
<draw-outline>	Boolean	Whether an outline should be drawn around the label

Configuring Allele Appearance

Each subject has an associated genotype which consists of markers and their associated alleles. The allele annotation displays the alleles for a specified marker. The following XML defines an allele annotation for the marker identified by D16SA. It will be displayed in a 12 point font.

```
<annotation type="allele">
    <marker-name>D16SA</marker-name>
    <font><size>12</size></font>
</annotation>
```

Table 3-6 Allele annotation elements

Element	Values	Description
<marker-name>	String	The name of the marker whose allele should be displayed
<font>	<font> subelements	The font to be used in displaying the label text
<color>	red green blue triplet	The color of the text
<virtual-color>	red green blue triplet	The color of the text when the subject is virtual
<insets>	integer (top) integer (left) integer (bottom) integer (right)	The padding to surround this annotation
<background-color>	red green blue triplet	The background color
<outline-color>	red green blue triplet	The outline color
<draw-background>	Boolean	Whether the background should be colored
<draw-outline>	Boolean	Whether an outline should be drawn around the label

Configuring Property Appearance

Each subject has associated properties which may correspond to phenotypic data or other subject-specific data. The property annotation displays the value of a specified property. The following XML defines an allele annotation for the property identified by “responded”. It will be displayed in a 12 point font.

```
<annotation type="property">
    <property-name>responded</property-name>
    <font><size>12</size></font>
</annotation>
```

Table 3-7 Property annotation elements

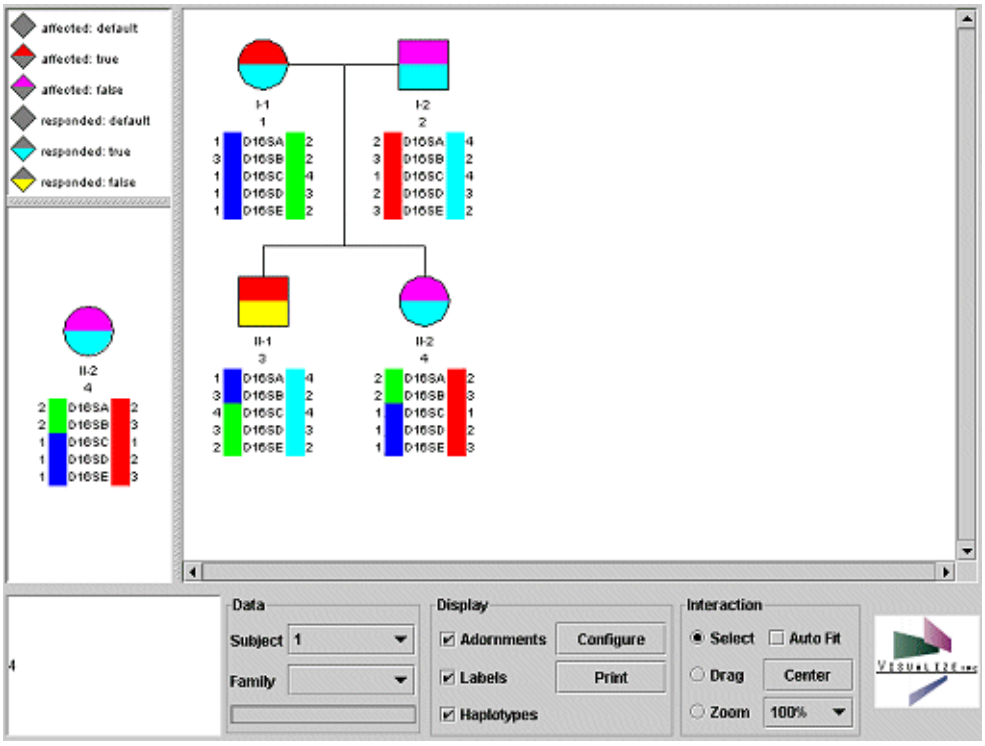
Element	Values	Description
<property-name>	String	The name of the property whose value should be displayed
<prefix>	String	Text that should be displayed before the property value
<font>	<font> subelements	The font to be used in displaying the label text
<color>	red green blue triplet	The color of the text
<virtual-color>	red green blue triplet	The color of the text when the subject is virtual
<insets>	integer (top) integer (left) integer (bottom) integer (right)	The padding to surround this annotation
<background-color>	red green blue triplet	The background color
<outline-color>	red green blue triplet	The outline color
<draw-background>	Boolean	Whether the background should be colored
<draw-outline>	Boolean	Whether an outline should be drawn around the label

Configuring Haplotype Appearance

Each subject has an associated genotype which consists of markers and their associated alleles. A collection of markers may be grouped into a haplotype, and each marker for a subject may be given a haplotype assignment. The haplotype annotation enables the display of the haplotype assignment for an individual. The following XML defines a haplotype annotation which will display the haplotype assignment for the haplotype whose id is D16S.

❖ **Example:** The following code and figure shows a haplotype appearance configuration. You can view the HaplotypeExample applet and associated configuration file (HaplotypeExample.html and HaplotypeExample.dvped) in the distribution package Examples directory. For more information about the <haplotype> element added to the subject data, see Configuring Haplotype Data from an XML Datasource.

```
<annotation type="haplotype">
    <id>D16A</id>
    <background-color>255 255 0</background-color>
    <draw-background>true</draw-background>
    <mutation-color>255 255 0</mutation-color>
 </annotation>
```



The following table lists all the haplotype elements:

Table 3-8 Haplotype annotation elements

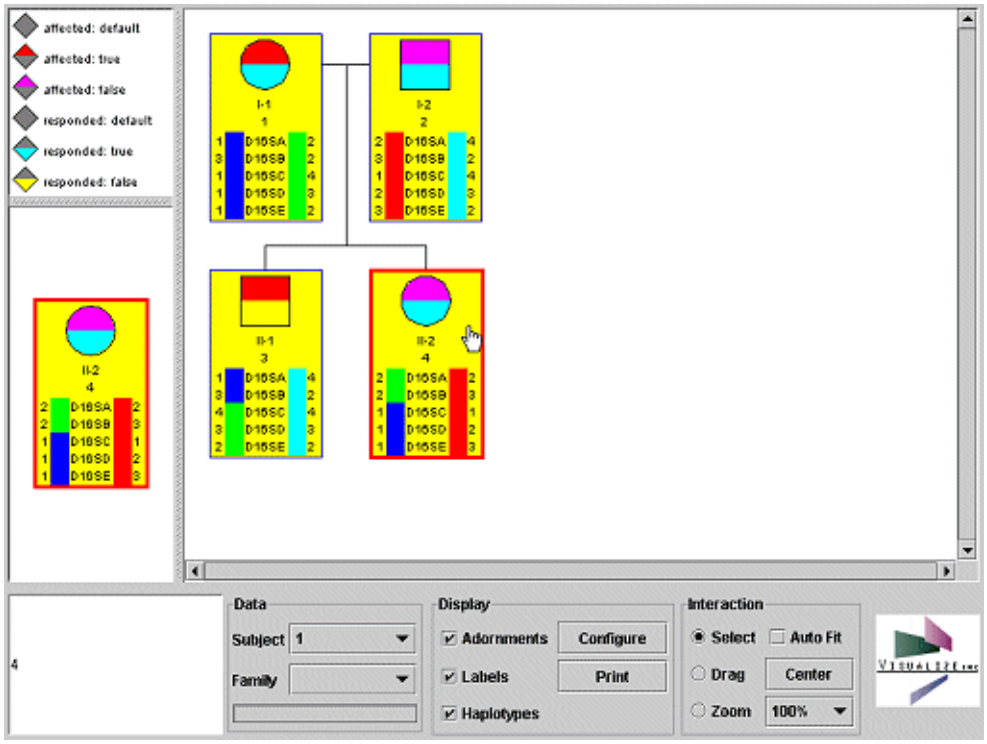
Element	Values	Description
<font>	<font> subelements	The font to be used in displaying the label text
<text-color>	red green blue triplet	The color of the text
<insets>	integer (top) integer (left) integer (bottom) integer (right)	The padding to surround this annotation
<background-color>	red green blue triplet	The background color
<outline-color>	red green blue triplet	The outline color
<draw-background>	Boolean	Whether the background should be colored
<draw-outline>	Boolean	Whether an outline should be drawn around the label
<mutation-color>	red green blue triplet	The color of the X displayed for mutation sites
<indeterminate-color>	red green blue triplet	The color of the vertical bar displayed for indeterminate values
<show-alleles>	Boolean	Whether to display the allele values
<id>	String	The id of the haplotype to display
<haplotype-color>	red green blue triplet	The color to assign to the given haplotype

Configuring Other Aspects of Subject Appearance

In addition to the aforementioned annotations, the subject display may be configured in other ways.

❖ **Example:** The following code and figure shows additional subject appearance configuration. You can view the SubjectAppearanceExample applet and associated configuration file (SubjectAppearanceExample.html and SubjectAppearanceExample.dvped) in the distribution package Examples directory.

```
<subject>
    <outline-color>0 0 255</outline-color>
    <background-color>255 255 0</background-color>
    <selected-color>255 0 0</selected-color>
</subject>
```



The following table enumerates the additional elements available in the configuration of subject appearance:

Table 3-9 Subject elements

Element	Values	Description
<insets>	integer (top) integer (left) integer (bottom) integer (right)	The padding to surround this subject
<background-color>	red green blue triplet	The color of the background area behind a subject.
<virtual-color>	red green blue triplet	The outline color that should be employed when the subject is virtual

Table 3-9 Subject elements

Element	Values	Description
<outline-color>	red green blue triplet	The color of the outline
<selected-color>	red green blue triplet	The color of the outline when the subject is selected
<highlighted-color>	red green blue triplet	The color of the outline when the subject is highlighted.

Configuring Pedigree Data

You configure pedigree data with the Dataset block. The <dataset> element defines the datasource, family structure, phenotypic, and genotypic data.

DVP can load data from the following sources:

- XML
- JDBC
- CSV
- Custom datasources

The data format you use depends on the type of datasource you choose. However, regardless of the datasource, pedigree data are defined as a collection of subjects, each of which contains defining data such as its parents, properties, and alleles. Each subject's position within the pedigree is specified by the identity of the parents.

Defining Data with an XML Datasource

When you use XML as the datasource, you define subject data within a <subject> element.

You can reference a separate XML file or embed the XML data within the configuration XML itself, as explained in the following sections.

Referencing an XML Data File

When you reference an XML data file, you insert a URL with the location of the file. In the following example, the XML data loads data from the URL http://pedserver.yourdomain.com/pedigree.xml:

```
<dataset>
    <pedigree source="servlet">
        <url>http://pedserver.yourdomain.com/pedigree.xml</url>
    </pedigree>
</dataset>
```

The resource located at the given URL contains XML identical in format to that embedded within a configuration XML.

Embedding XML Data

When you embed XML data within the configuration XML, you use the <dataset><pedigree> element. In the following example, the XML defines a pedigree with 21 subjects and a single haplotype identified by D16A:

```
<dataset>
    <pedigree>
        <subject id="1">
            <label>"SID0001"</label>
```

```
            <mates>2</mates>
            <gender>male</gender>
            <properties>
               <affected>true</affected>
            </properties>
            <genotype>
               <allele marker="D16SA">1 2</allele>
               <allele marker="D16SB">3 2</allele>
               <allele marker="D16SC">1 4</allele>
               <allele marker="D16SD">1 3</allele>
               <allele marker="D16SE">1 2</allele>
            </genotype>
         </subject>
         . . .
         <subject id="21">
            <label>"SID0021"</label>
            <parents>14 20</parents>
            <gender>male</gender>
            <living>false</living>
            <properties>
               <affected>true</affected>
            </properties>
            <genotype>
               <allele marker="D16SA">2 4</allele>
               <allele marker="D16SB">3 2</allele>
               <allele marker="D16SC">1 4</allele>
               <allele marker="D16SD">2 3</allele>
               <allele marker="D16SE">3 2</allele>
            </genotype>
         </subject>
         <haplotype id="D16A">
         D16SA D16SB D16SC D16SD D16SE
         </haplotype>
      </pedigree>
   </dataset>
```

Configuring Subject Data from an XML Datasource

You use the <subject> element to define the contents of each subject.

In the following example, XML elements define a single subject, identified by SID00003, and having parents with ids 1 and 2. The property “affected” has a value of “true”, and the genotype for markers D16SA, D16SB, D16SC, D16SD and D16SE is specified.

```
<subject id="3">
    <label>"SID0003"</label>
    <parents>1 2</parents>
    <gender>male</gender>
    <living>true</living>
    <properties>
       <affected>true</affected>
    </properties>
    <genotype>
       <allele marker="D16SA">1 2</allele>
       <allele marker="D16SB">3 2</allele>
       <allele marker="D16SC">1 4</allele>
       <allele marker="D16SD">1 3</allele>
       <allele marker="D16SE">1 2</allele>
    </genotype>
</subject>
```

Each subject requires an id, specified by the “id” attribute of the <subject> element. A subject is further defined by various XML elements that specify the label, parents, gender, and other subject attributes. The following table lists the sub-elements for <subject>.

Table 3-10 <subject> elements

Element	Values	Description
<label>	String	The text that should be employed in labeling the subject in the pedigree view.
<description>	String	Descriptive text pertaining to the subject.
<parents>	List of subject ids	The ids of the parents.
<gender>	male | female | unknown	The subject’s gender. When omitted, the gender defaults to unknown.
<properties>	Subtags	The properties assigned to this subject.
<living>	true | false	Whether the subject is living.
<proband>	true | false	Whether the subject is the proband.
<twin-group>	String	The name of the twin group to which this subject belongs.
<family-group>	String	The name of the family group for a subject.
<genotype>	<allele> subtags	The genotype of the subject, defined as a collection of alleles.
<haplotype-assignment>	<chromosome> subtags	The haplotype to which each allele should be assigned.

Configuring Properties Data from an XML Datasource

You define properties within the <properties> element, which is subordinate to the <subject> element. Properties are subject data that you want the pedigree to display.

Each property’s value is defined by a separate element.

For example, the following XML defines the affected property to have a value of “true”:

```
<properties>
    <affected>true</affected>
</properties>
```

Properties may have any value. Properties that contain spaces should be surrounded by quotation marks. For example:

```
<properties>
    <birthdate>”January 24, 1969”</birthdate>
</properties>
```

A subject may have any number of properties assigned to it.

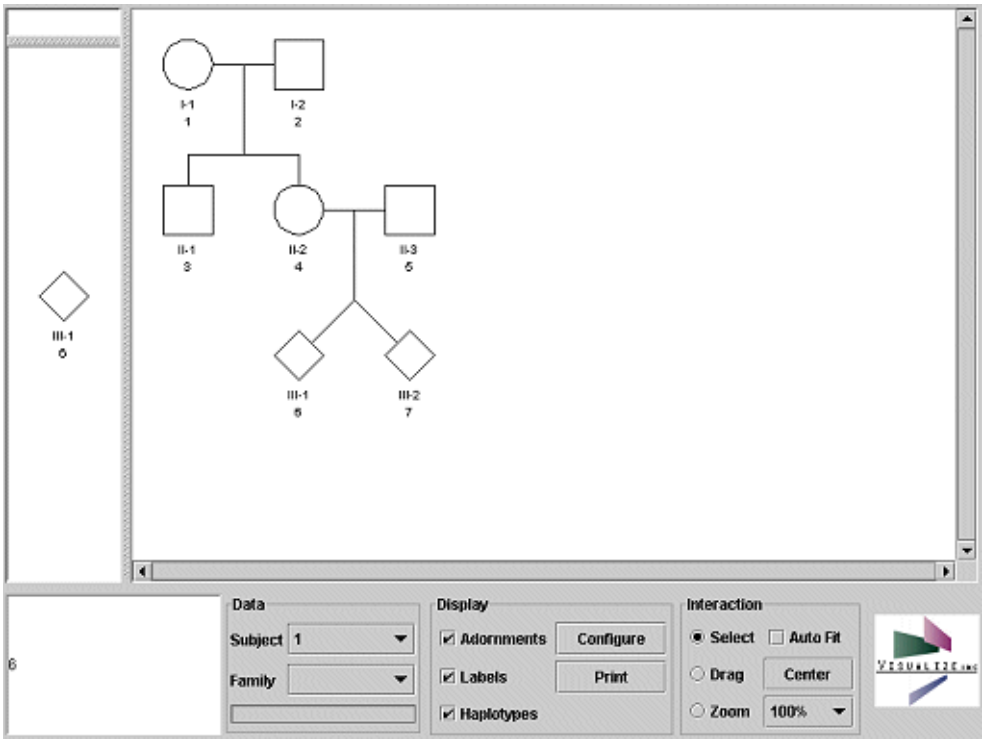
You can view an example of a <properties> element configuration in the AdornmentsExample configuration file (AdornmentsExample.dvped) in the distribution package Examples directory.

Configuring Twin Group Data from an XML Datasource

You can assign each subject to a “twin group” by specifying the id of the twin group using the <twin-group> element (which is subordinate to <subject>). A twin group is a group of siblings who should be displayed as twins within a family.

❖ **Example:** The following code and figure shows a twin group configuration. You can view the TwinGroupExample applet and associated configuration file (TwinGroupExample.html and TwinGroupExample.dvped) in the distribution package Examples directory.

```
<subject id="1">
    <label>"1"</label>
    <gender>female</gender>
</subject>
<subject id="2">
    <label>"2"</label>
    <gender>male</gender>
</subject>
<subject id="3">
    <label>"3"</label>
    <parents>1 2</parents>
    <gender>male</gender>
</subject>
<subject id="4">
    <label>"4"</label>
    <parents>1 2</parents>
    <gender>female</gender>
</subject>
<subject id="5">
    <label>"5"</label>
    <gender>male</gender>
</subject>
<subject id="6">
    <label>"6"</label>
    <parents>4 5</parents>
    <gender>female></gender>
    <twin-group>a</twin-group>
</subject>
<subject id="7">
    <label>"7"</label>
    <parents>4 5</parents>
    <gender>female></gender>
    <twin-group>a</twin-group>
</subject>
```



Configuring Genotype Data from an XML Datasource

You define the genotype of a subject within the <genotype> element, subordinate to <subject>. The <genotype> element consists of a list of allele values assigned to arbitrary markers. The <allele> element contains the allele assignment to a given marker.

For example, the following XML assigns alleles 1 and 2 to the marker “D16SA”:

```
<genotype>
    <allele marker="D16SA">1 2</allele>
</genotype>
```

There is no limit to the number of alleles assigned to a given subject.

You can view an example of a <genotype> element configuration in the HaplotypeExample configuration file (HaplotypeExample.dvped) in the distribution package Examples directory.

Configuring Haplotype Data from an XML Datasource

You define a haplotype with the <haplotype> element, which contains an enumeration of marker names. The <haplotype> element is subordinate to the <subject> element. Haplotypes are identified by an “id” attribute.

For example, the following XML defines three haplotypes, identified by D16A, D16B and D16C:

```
<haplotype id="D16A">
D16SA D16SB D16SC D16SD D16SE D16SF D16SG D16SH D16SI D16SJ D16SK
D16SL D16SM D16SN D16SO D16SP D16SQ D16SR D16SS D16ST D16SU
</haplotype>
<haplotype id="D16B">
D16SA D16SB D16SC D16SD D16SE D16SF D16SG D16SH D16SI D16SJ
</haplotype>
<haplotype id="D16C">
```

```
D16SK D16SL D16SM D16SN D16SO D16SP D16SQ D16SR D16SS D16ST D16SU
</haplotype>
```

You can view an example of a <haplotype> element configuration in the HaplotypeExample configuration file (HaplotypeExample.dvped) in the distribution package Examples directory.

Defining Data with a JDBC Datasource

JDBC is the standard Java means of accessing relational databases through the execution of SQL queries. Data retrieved via JDBC is done through the specification of an SQL query along with how each column in the returned table should be interpreted.

The following XML shows configuration code from a JDBC datasource:

```
<pedigree source="jdbc">
    <query>”select
    label,id,gender,mother,father,bestsub,alcohol,D16SA,D16SB,D16SC
    ,D16SD from datatable”</query>
    <server>jdbc:sqldb://databaseserver/geneticdata</server>
    <user>joe</user>
    <password>passwd</passwd>
    <label-field>1</label-field>
    <id-field>2</id-field>
    <gender-field>3</gender-field>
    <parent-fields>4 5</parent-fields>
    <properties>
       <property>
          <name>BestSub</name><field>6</field>
       </property>
       <property>
          <name>Alcohol</name><field>7</field>
       </property>
    </properties>
    <alleles>
       <allele>
          <name>D16SA</name>
          <fields>16 17</fields>
       </allele>
       <allele>
          <name>D16SB</name>
          <fields>18 19</fields>
       </allele>
       <allele>
          <name>D16SC</name>
          <fields>20 21</fields>
       </allele>
       <allele>
          <name>D16SD</name>
          <fields>22 23</fields>
       </allele>
    </alleles>
</pedigree>
```

The following table lists the XML elements to configure when you use a JDBC datasource:

Table 3-11 Elements for Using a JDBC Datasource

Element	Values	Description
<query>	String	The SQL query that will be executed on the server
<server>	URL	The URL of the database server
<driver>	String	The name of the JDBC driver class. The driver class must be in the classpath of the applet.
<user>	String	The username to be used in logging into the database server
<password>	String	The password to be used in logging into the database server
<label-field>	Integer	The index of the label field (starting at 1).
<id-field>	Integer	The index of the id field (starting at1).
<gender-field>	Integer	The index of the gender field (starting at 1).
<url-field>	Integer	The index of the url field (starting at 1). The URL field is employed when DVP links to related web resources in response to the user clicking on a subject.
<twin-group-field>	Integer	The index of the field that contains the twin group for a subject.
<family-group-field>	Integer	The index of the field that contains the family group for a subject.
<living-field>	Integer	The index of the living field (starting at 1).
<proband-field>	Integer	The index of the proband field (starting at 1).
<parent-fields>	Integers	The index of the parent fields (starting at 1).
<properties>	<property> subtags	The fields for each property.
<alleles>	<allele> subtags	The fields for each allele.
<debug>	true | false	Enables debug information to be output to the console.

Table 3-11 Elements for Using a JDBC Datasource

Element	Values	Description
<elide-indeterminate-haplotypes>	true | false	Whether the haplotype designation of alleles whose haplotype is indeterminate should be elided or interpolated from surrounding haplotype assignments.
<haplotype>	List of marker id's	A haplotype definition.

Defining Data with a CSV Datasource

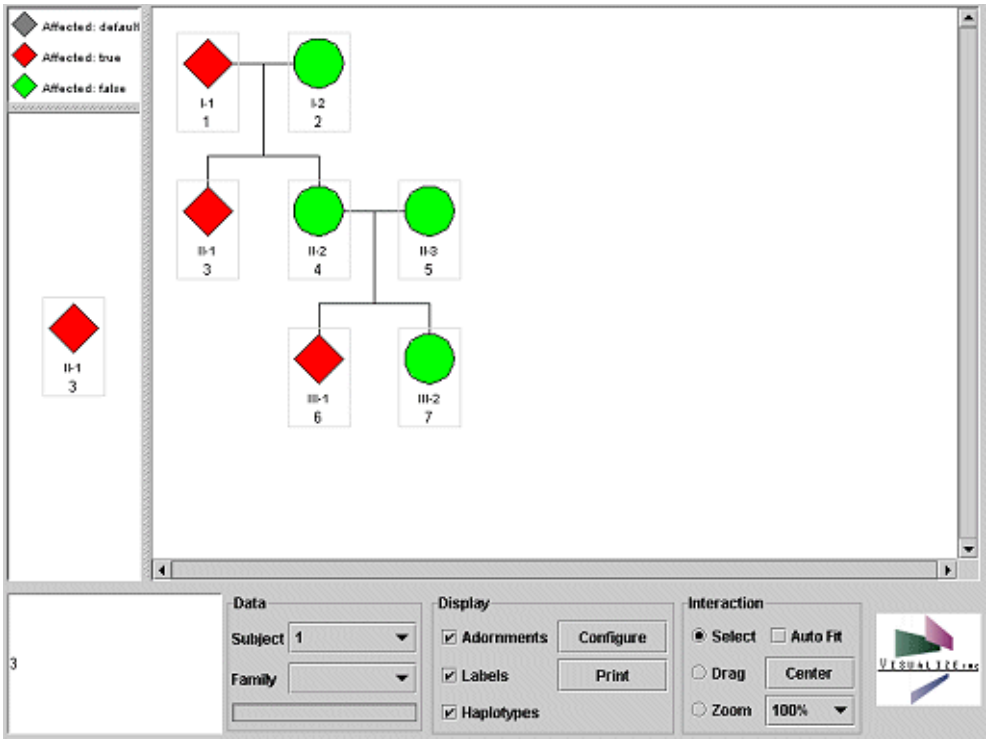
DVP can also load data from an ASCII delimited (CSV) source located at a specified URL on the network.

❖ **Example:** The following code and figure shows XML from a .dvped file that retrieves data from the text file: "CVSDatasourceExample.txt". You can view the CSVDatasourceExample applet (CSVDatasourceExample.html), associated configuration file (CSVDatasourceExample.dvped), and associated text file (CSVDatasourceExample.text) in the distribution package Examples directory.

```
<pedigree source="csv">
    <url>CSVDatasourceExample.txt</url>
    <label-field>1</label-field>
    <id-field>1</id-field>
    <gender-field>4</gender-field>
    <parent-fields>2 3</parent-fields>
    <properties>
       <property>
          <name>Affected</name><field>4</field>
       </property>
       <property>
          <name>Diagnosis</name><field>5</field>
       </property>
    </properties>
</pedigree>
```

The following code shows the data in the "CVSDatasourceExample.txt":

```
41  1   0   0   true   R-UP
41  2   0   0   false  BPI
41  3   1   2   true   BPII
41  4   1   2   false  BPI
41  5   0   0   false  BPII
41  6   4   5   true   R-UP
41  7   4   5   false  BPI
```



The following table lists the XML elements to configure when you use a CSV datasource:

Table 3-12 Elements for Using a CSV Datasource

Element	Values	Description
<url>	URL	The URL of the file to load.
<delimiters>	String	The field separators.
<label-field>	Integer	The index of the label field (starting at 0).
<id-field>	Integer	The index of the id field (starting at 0).
<gender-field>	Integer	The index of the gender field (starting at 0).
<url-field>	Integer	The index of the url field (starting at 0). The URL field is employed when DVP links to related web resources in response to the user clicking on a subject.
<twin-group-field>	Integer	The index of the field that contains the twin group for a subject.
<family-group-field>	Integer	The index of the field that contains the family group for a subject.
<living-field>	Integer	The index of the living field (starting at 0).

Table 3-12 Elements for Using a CSV Datasource

Element	Values	Description
<proband-field>	Integer	The index of the proband field (starting at 0).
<parent-fields>	Integers	The index of the parent fields (starting at 0).
<properties>	<property> subtags	The fields for each property.
<alleles>	<allele> subtags	The fields for each allele.
<debug>	true | false	Enables debug information to be output to the console.
<elide-indeterminate-haplotypes>	true | false	Whether the haplotype designation of alleles whose haplotype is indeterminate should be elided or interpolated from surrounding haplotype assignments.
<haplotype>	List of marker id's	A haplotype definition.

Configuring Properties Data from a CSV Datasource

You configure the fields that contain the values of a given property within the <properties> tag. Furthermore, each field is defined by an individual <property> tag.

For example, the following XML specifies that the value of the "Age" property may be found within the thirteenth field:

```
<properties>
    <property>
        <name>Age</name><field>13</field>
    </property>
</properties>
```

There is no limit to the number of properties that may be defined in this manner.

Configuring Alleles Data from a CSV Datasource

You define the fields that contain the alleles for a given marker within the <alleles> element. Furthermore, each field is defined by a subordinate <allele> tag.

For example, the following XML specifies that the alleles assigned to the D16SA marker should be found in fields sixteen and seventeen:

```
<alleles>
    <allele>
        <name>D16SA</name>
        <fields>16 17</fields>
    </allele>
</alleles>
```

There is no limit to the number of alleles that may be defined in this manner.

Configuring Haplotype Data from a CSV Datasource

You define a haplotype with the <haplotype> element, which contains an enumeration of marker names. Haplotypes are identified by an "id" attribute.

For example, the following XML defines three haplotypes, identified by D16A, D16B and D16C:

```
<haplotype id="D16A">
```

```
D16SA D16SB D16SC D16SD D16SE D16SF D16SG D16SH D16SI D16SJ D16SK
D16SL D16SM D16SN D16SO D16SP D16SQ D16SR D16SS D16ST D16SU
</haplotype>
<haplotype id="D16B">
D16SA D16SB D16SC D16SD D16SE D16SF D16SG D16SH D16SI D16SJ
</haplotype>
<haplotype id="D16C">
D16SK D16SL D16SM D16SN D16SO D16SP D16SQ D16SR D16SS D16ST D16SU
</haplotype>
```

Defining Data with a Custom Datasource

You can also import data through a custom datasource. Custom data sources are Java objects that implement the DataSourceIF interface defined by DVP.

To import data through a custom datasource, implement the following Java methods:

```
public interface DataSourceIF extends Cloneable {
    void configure(Enumeration in, String endtoken,
        ParameterQuery parameterQuery,
        ProgressTracker progressTracker,
        boolean querySource)
        throws ConfigurationException;

    void setDocumentBase(URL documentBase);

    Pedigree queryPedigree(ProgressTracker progressTracker) throws
    DataException;

    void addDataDeltaListener(DataDeltaListener ddl);

    void removeDataDeltaListener(DataDeltaListener ddl);

    Object clone();
}
```

The key method in this interface is queryPedigree(). It is responsible for returning the data defining the pedigree to be displayed by DVP. A pedigree is a collection of subjects, with each subject encapsulated within a Subject class.

☛ **Note:** For more information, see the Javadoc-based documentation.

Deploying DataVista Pedigree

After placing the HTML files and their associated DVP configuration files and data source plug-ins on your web server, you should be able to point your browser to the corresponding URL and interact with the applet. If it is not functioning properly, you should verify that the correct license is in the DVP configuration and that the APPLET tag is referencing the correct CODEBASE and ARCHIVE.

❖ **Example:** The following code and figure shows a configured applet. You can view the DVPAppletExample applet and associated configuration file (DVPAppletExample.html and DVPAppletExample.dvped) in the distribution package Examples directory. You may also want to use this example as a template for your configuration.

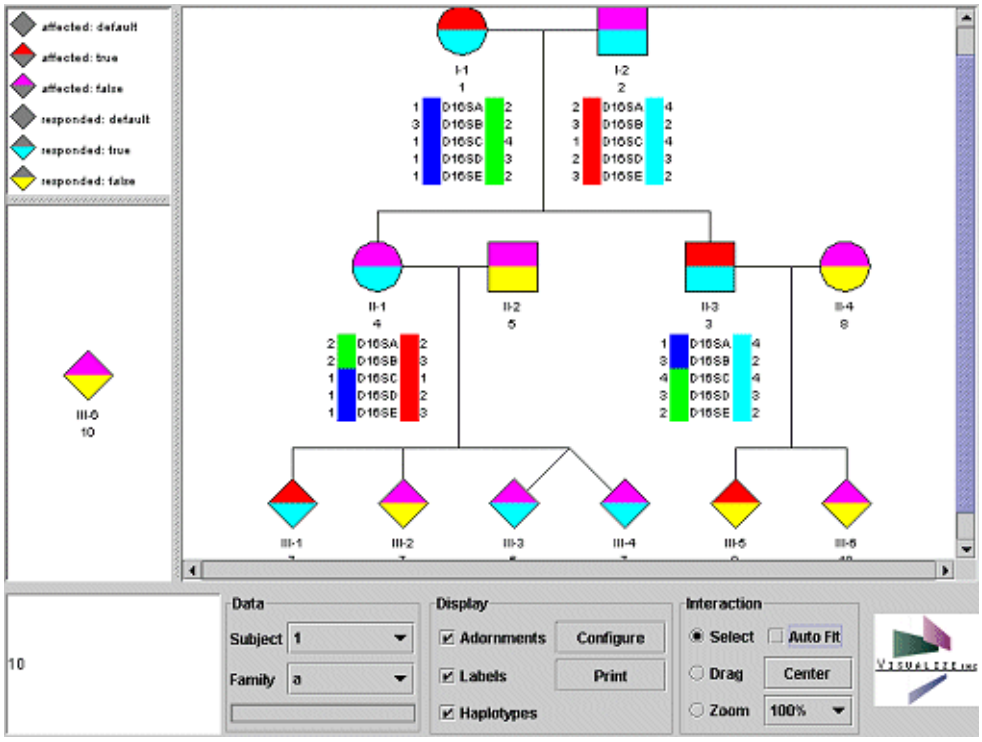
```
<datavistapedigree>
    <licensing>
```

```
    <keyword>Visualize</keyword>
    <license>AVLQNVSYETSBSVRTLWOW</license>
</licensing>
<options>
    <handlers>
        <handler type="hyperlink">
        </handler>
    </handlers>
</options>
<viewer>
    <pedigree-renderer>
        <subject>
            <annotations>
                <annotation centered="true" type="symbol">
                    <adornments>
                        <top>
                            <property-name>affected</property-name>
                            <color value="true">255 0 0</color>
                            <color value="false">255 0 255</color>
                            <color>127 127 127</color>
                        </top>
                        <bottom>
                            <property-name>responded</property-name>
                            <color value="true">0 255 255</color>
                            <color value="false">255 255 0</color>
                            <color>127 127 127</color>
                        </bottom>
                    </adornments>
                </annotation>
                <annotation type="index">
                </annotation>
                <annotation type="label">
                </annotation>
                <annotation type="property">
                </annotation>
                <annotation type="haplotype">
                    <id>D16A</id>
                    <background-color>255 255 0</background-color>
                    <draw-background>true</draw-background>
                    <mutation-color>255 255 0</mutation-color>
                </annotation>
            </annotations>
        </subject>
    </pedigree-renderer>
</viewer>
<dataset>
    <pedigree>
        <subject id="1">
            <label>"1"</label>
            <gender>female</gender>
            <properties>
                <affected>true</affected>
                <responded>true</responded>
            </properties>
            <genotype>
                <allele marker="D16SA">1 2</allele>
                <allele marker="D16SB">3 2</allele>
                <allele marker="D16SC">1 4</allele>
                <allele marker="D16SD">1 3</allele>
```

```
            <allele marker="D16SE">1 2</allele>
        </genotype>
    </subject>
    <subject id="2">
        <label>"2"</label>
        <gender>male</gender>
        <properties>
            <affected>false</affected>
            <responded>true</responded>
        </properties>
        <genotype>
            <allele marker="D16SA">2 4</allele>
            <allele marker="D16SB">3 2</allele>
            <allele marker="D16SC">1 4</allele>
            <allele marker="D16SD">2 3</allele>
            <allele marker="D16SE">3 2</allele>
        </genotype>
    </subject>
    <subject id="3">
        <label>"3"</label>
        <parents>1 2</parents>
        <gender>male</gender>
        <properties>
            <affected>true</affected>
            <responded>true</responded>
        </properties>
        <family-group>b</family-group>
        <genotype>
            <allele marker="D16SA">1 4</allele>
            <allele marker="D16SB">3 2</allele>
            <allele marker="D16SC">4 4</allele>
            <allele marker="D16SD">3 3</allele>
            <allele marker="D16SE">2 2</allele>
        </genotype>
    </subject>
    <subject id="4">
        <label>"4"</label>
        <parents>1 2</parents>
        <gender>female</gender>
        <properties>
            <affected>false</affected>
            <responded>true</responded>
        </properties>
        <family-group>a</family-group>
        <genotype>
            <allele marker="D16SA">2 2</allele>
            <allele marker="D16SB">2 3</allele>
            <allele marker="D16SC">1 1</allele>
            <allele marker="D16SD">1 2</allele>
            <allele marker="D16SE">1 3</allele>
        </genotype>
    </subject>
    <subject id="5">
        <label>"5"</label>
        <gender>male</gender>
        <properties>
            <affected>false</affected>
            <responded>false</responded>
        </properties>
```

```
    <family-group>a</family-group>
</subject>
<subject id="6">
    <label>"6"</label>
    <parents>4 5</parents>
    <gender>female></gender>
    <properties>
        <affected>false</affected>
        <responded>true</responded>
    </properties>
    <twin-group>a</twin-group>
    <family-group>a</family-group>
</subject>
<subject id="7">
    <label>"7"</label>
    <parents>4 5</parents>
    <gender>female></gender>
    <properties>
        <affected>false</affected>
        <responded>true</responded>
    </properties>
    <twin-group>a</twin-group>
    <family-group>a</family-group>
</subject>
<subject id="8">
    <label>"8"</label>
    <gender>female</gender>
    <properties>
        <affected>false</affected>
        <responded>false</responded>
    </properties>
    <family-group>b</family-group>
</subject>
<subject id="9">
    <label>"9"</label>
    <parents>3 8</parents>
    <gender>male></gender>
    <properties>
        <affected>true</affected>
        <responded>false</responded>
    </properties>
    <family-group>b</family-group>
</subject>
<subject id="10">
    <label>"10"</label>
    <parents>3 8</parents>
    <gender>female></gender>
    <properties>
        <affected>false</affected>
        <responded>false</responded>
    </properties>
    <family-group>b</family-group>
</subject>
<subject id="11">
    <label>"7"</label>
    <parents>4 5</parents>
    <gender>female></gender>
    <properties>
        <affected>true</affected>
```

```
                <responded>true</responded>
            </properties>
            <family-group>a</family-group>
        </subject>
        <subject id="12">
            <label>"7"</label>
            <parents>4 5</parents>
            <gender>male></gender>
            <properties>
                <affected>false</affected>
                <responded>false</responded>
            </properties>
            <family-group>a</family-group>
        </subject>
        <haplotype id="D16A">
        D16SA D16SB D16SC D16SD D16SE
        </haplotype>
    </pedigree>
  </dataset>
</datavistapedigree>
```



4 JavaScript Interface

JavaScript Interface Overview

The XML-based configuration of DVP offers extensive control. However, you can also embed JavaScript methods within the .html page used to deploy DVP to add more user interactivity. By leveraging the power afforded by JavaScript, you can provide users the ability to:

- Modify the data displayed within the applet.
- Attach and detach the applet from within the web page.
- Set the display characteristics, such as zoom level, adornments, and so forth.
- Export data.
- Export graphics.

Note: The following documentation assumes that you are familiar with JavaScript.

The “JavaScript methods” mentioned in this chapter are public Java methods found in class com.visualizeinc.datavistapedigree.applet.DataVistaPedigree.

Configuring Data Modification Using JavaScript

You can provide the ability to modify data within the applet using the method in the following table:

Table 4-1 Data Modification JavaScript Methods

Method	Description
configureDataSet(java.lang.String xmlParameters)	Configures the DataSet with XML-formatted configuration information.

Configuring Applet Detachment Using JavaScript

You can provide the ability to attach and detach the applet from within the web page using the methods in the following table:

Table 4-2 Applet Detachment JavaScript Methods

Method	Description
detach(int width, int height)	Detaches the applet from the web page and displays it within a separate frame with the specified dimensions.
attach()	Returns the applet from being displayed in a separate frame to the window in which the program was initially launched.

Configuring Display Characteristics Using JavaScript

You can provide the ability to set the display characteristics, such as zoom level, adornments, and so forth, using the methods in the following table:

Table 4-3 Display Characteristic Modification JavaScript Methods

Method	Description
configurePedigreeViewer(java.lang.String xmlParameters)	Configures the pedigree viewer with XML-formatted configuration information.
getSelectedSubjectIds()	Returns the ids of the currently selected subjects within a Vector.
getSelectedSubjectIdArray()	Returns the ids of the currently selected subjects within an array.
getSelectedSubjectIdsString()	Returns the ids of the currently selected subjects as a String.
setDisplayAdornments(boolean displayAdornments)	Sets whether adornments should be displayed within the pedigree symbols.
setDisplayHaplotypes(boolean displayHaplotypes)	Sets whether haplotypes should be displayed within the pedigree.
setDisplayLabels(boolean displayLabels)	Sets whether labels should be displayed within the pedigree.
setZoomLevel(int zoomScaleFactor, boolean centered)	Sets the current zoom level of the pedigree display.

Configuring Data Export Using JavaScript

You can provide the ability to export data using the methods in the following table:

Table 4-4 Data Export JavaScript Methods

Method	Description
exportData(java.lang.String servletURL)	Exports the currently displayed data.
exportData(java.lang.String servletURL, java.lang.String mime, java.lang.String filename, boolean attachment, java.lang.String target)	Exports the currently displayed data.

Configuring Graphics Export Using JavaScript

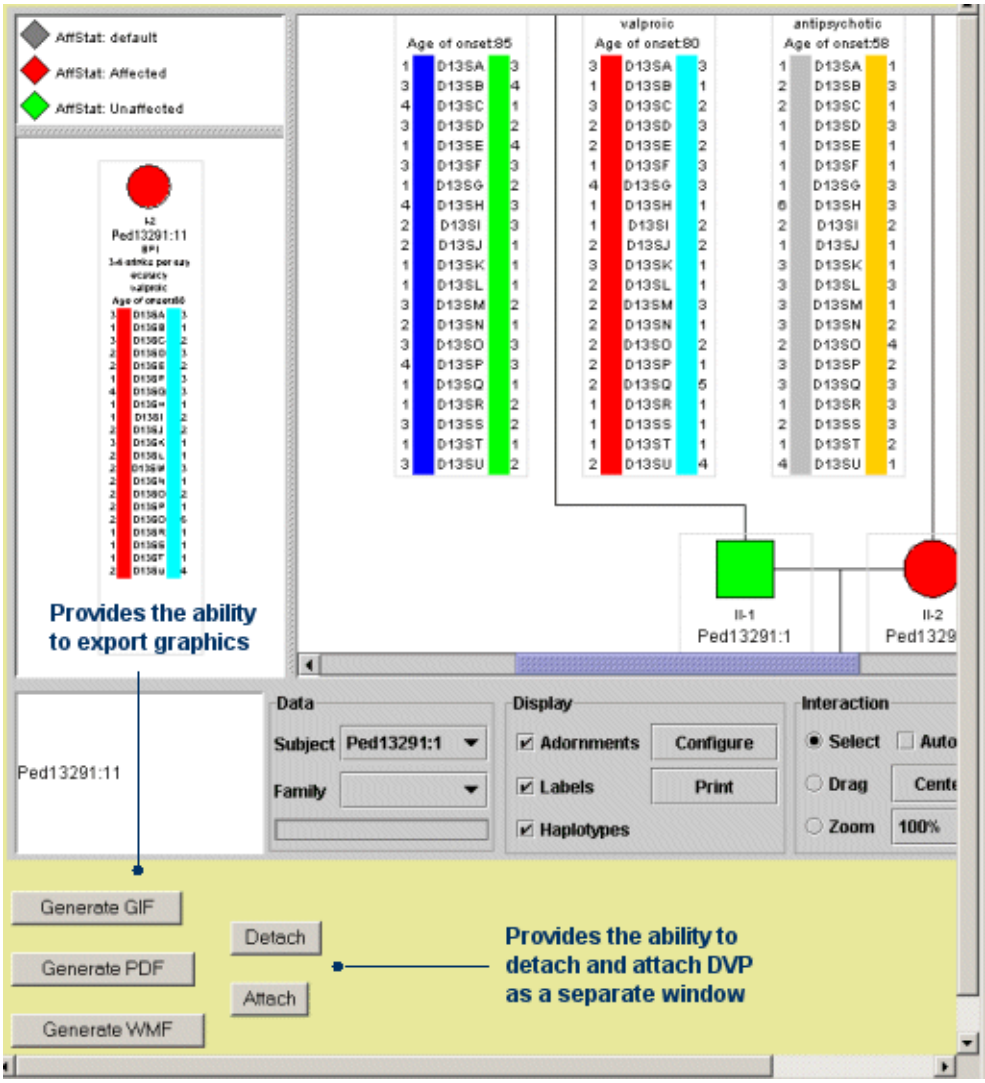
You can provide the ability to export graphics using the methods in the following table:

Table 4-5 Graphics Export JavaScript Methods

Method	Description
generateGraphics(java.lang.String servletURL, java.lang.String mimeType, int width, int height)	Generates a static image of the current pedigree.
generateGraphics(java.lang.String servletURL, java.lang.String mimeType, int width, int height, java.lang.String filename, boolean attachment, java.lang.String target)	Generates a static image of the current pedigree.

JavaScript Example

The following example shows how you would use JavaScript to provide the ability to detach and attach the applet and export graphics.




```
<html>
<head>
<title>DataVista Pedigree - Example A</title>
</head>
<body text="#000000" bgcolor="#e6e79c" background = "../
contentbackground.gif"link="#003399" vlink="#6C9EFF"
alink="#ff0000">
<font face="arial,helvetica">
<script language="JavaScript">

<!--
var popupImageWindow = null;

function generateGIF() {
  var name = "ExportedImage";

  var url = '/jvpub-gs-jdk12-2.2/servlet/GraphicsServlet';

  var width = 500;
  var height = 500;
```

```
  var imageID = document.dvped.generateGraphics(url,"image/
gif",width,height);

  url = url+"?graphics="+imageID;

  if(popupImageWindow == null || popupImageWindow.closed)
  {
    popupImageWindow = window.open(url, name,
'status=no,toolbar=no,menubar=no,scrollbars=no,width='+(width+20)+'
,height='+(height+30));
  }
  else
  {
    popupImageWindow.open(url,name);
  }

}

function generateWMF() {
  var url = '/jvpub-gs-jdk12-2.2/servlet/GraphicsServlet';

  document.dvped.generateGraphics(url,"image/x-
wmf",0,0,"dvped.wmf",true,"WMFFrame");
}

function generatePDF() {
  var url = '/jvpub-gs-jdk12-2.2/servlet/GraphicsServlet';

  var width = 625;
  var height = 550;

  document.dvped.generateGraphics(url,"application/
pdf",width,height,"dvped.pdf",false,"PDFFrame");
}

// -->

</script>

<OBJECT
    classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
    WIDTH = 800 HEIGHT = 600 NAME = "dvped"
    codebase="http://java.sun.com/products/plugin/autodl/jinstall-
1_4-win.cab#Version=1,4,0,0">
    <PARAM NAME = CODE VALUE =
"com.visualizeinc.datavistapedigree.applet.DataVistaPedigree" >

<PARAM NAME = CODEBASE VALUE = "." >

<PARAM NAME = ARCHIVE VALUE = "dvpedigree-applet-jdk12-1.0.6.jar" >

<PARAM NAME = NAME VALUE = "dvped" >

<PARAM NAME = MAYSCRIPT VALUE = true >


    <PARAM NAME="type" VALUE="application/x-java-
applet;version=1.4">
```

```
    <PARAM NAME="scriptable" VALUE="true">
    <PARAM NAME = "ConfigFilename" VALUE ="examplea.dvped">


    <COMMENT>
 <EMBED
            type="application/x-java-applet;version=1.4"
            CODE =
"com.visualizeinc.datavistapedigree.applet.DataVistaPedigree"
            CODEBASE = "."
            ARCHIVE = "dvpedigree-applet-jdk12-1.0.6.jar"
            NAME = "dvped"
            WIDTH = 800
            HEIGHT = 600
            MAYSCRIPT = true
            ConfigFilename = "examplea.dvped"
     scriptable=true
     pluginspage="http://java.sun.com/products/plugin/
index.html#download">
         <NOEMBED>

    </NOEMBED>
 </EMBED>
    </COMMENT>
</OBJECT>

<P>

<FORM>
<table>
<tr>
<td>
<INPUT TYPE="button" VALUE="Generate GIF"
onClick="generateGIF()"><P>
<INPUT TYPE="button" VALUE="Generate PDF"
onClick="generatePDF()"><P>
<INPUT TYPE="button" VALUE="Generate WMF"
onClick="generateWMF()"><P>
</td>
<td width = 10>
</td>
<td>
<INPUT TYPE="button" VALUE="Detach"
onClick="document.dvped.detach(800,600)"><P>
<INPUT TYPE="button" VALUE="Attach"
onClick="document.dvped.attach()"><P>
</td>
</tr>
</table>
</FORM>
</font>
</body>
```

5 Application Component Installation

Application Component Installation Overview

You can deploy DVP as either a Java applet within a web application or as an application component within a Swing-based application. This chapter includes information for installing DVP in the latter environment.

Component installation includes the following tasks:

- Adding the appropriate DVP jar to your CLASSPATH
- Populating a Pedigree object with the appropriate subjects and filial relationships
- Constructing a SwingPedigreeViewer object and setting it to display the desired Pedigree
- Configuring the pedigree display
- Setting up the interaction mode and any desired event listeners
- Adding the SwingPedigreeViewer to your application UI

The following documentation assumes that you are familiar with Java and have experience developing software in that language.

Adding the DVP Jar to Your CLASSPATH

DVP is distributed in two distinct jar files: dvpedigree-applet-jdk12-1.0.x.jar and dvpedigree-api-jdk12-1.0.x.jar. The second of these is the relevant one to employ when embedding DVP functionality within your application as it exposes the classes relevant to constructing and displaying a pedigree within a stand-alone application.

You should follow the steps appropriate to your operating system for appending the API jar to your classpath.

Populating a Pedigree

DVP encapsulates all data relating to the pedigree structure and subject attributes within a Pedigree object with its constituent Subject objects. A pedigree is constructed by first creating the individual Subjects that comprise it and then adding them into it, followed by a call to Pedigree's normalize() method. This final step instructs Pedigree to build the requisite filial relationships based upon the parentage of each individual subject.

Configuring a Subject Object

Each individual within a pedigree is encapsulated within a Subject object. Each subject has various attributes that define its phenotype and genotype, as well as its position within the pedigree. An individual's position within a pedigree is defined by its parents, with those at the root of the pedigree being individuals with no defined parents.

For example, data for a female individual identified by the label "SID001" would be instaniated and configured through the following Java code:

```
Subject mother = new Subject();
mother.setId("1");
mother.setLabel("SID001");
mother.setGender(Subject.giFemale);
mother.setProperty("affected","yes");
mother.setProperty("favorite-color","red");
```

Another male subject, may be created through similar Java code:

```
Subject father = new Subject();
father.setId("2");
father.setLabel("SID002");
father.setGender(Subject.giMale);
father.setProperty("affected","no");
father.setProperty("favorite-color","blue");
```

A third individual, a child of the previous two individuals, is created through the following code:

```
Subject child = new Subject();
child.setId("3");
child.setLabel("SID003");
child.setGender(Subject.giMale);
child.setProperty("affected","yes");
child.setProperty("favorite-color","purple");
Vector parentIDs = new Vector();
parentIDs.addElement("1");
parentIDs.addElement("2");
child.setParentIds(parentIDs);
```

Each of the previous subjects was given two properties: his or her affection status and favorite color. Each subject may be given an arbitrary number of properties that the DVP component may then be configured to display. The collection of properties of a Subject may be loosely deemed to be the individual's phenotype.

Genotypic information is configured through Subject's setAlleles() method. The following code configures an individual to have alleles 1 and 2 at the marker identified by D2S001.

```
child.setAlleles("D2S001","1","2");
```

Like phenotypic properties, a Subject may be configured to have an unlimited quantity of genotypic data.

Configuring a Pedigree Object

Once you instantiate the individual Subjects, you must assemble them into a Pedigree. You accomplish this by populating a Vector with the Subjects to be assembled and then adding it into a Pedigree object, followed by a call to normalize().

For example:

```
Vector subjects = new Vector();
subjects.addElement(mother);
subjects.addElement(father);
subjects.addElement(child);
Pedigree pedigree = new Pedigree();
pedgree.addSubjects(subjects);
pedigree.normalize();
```

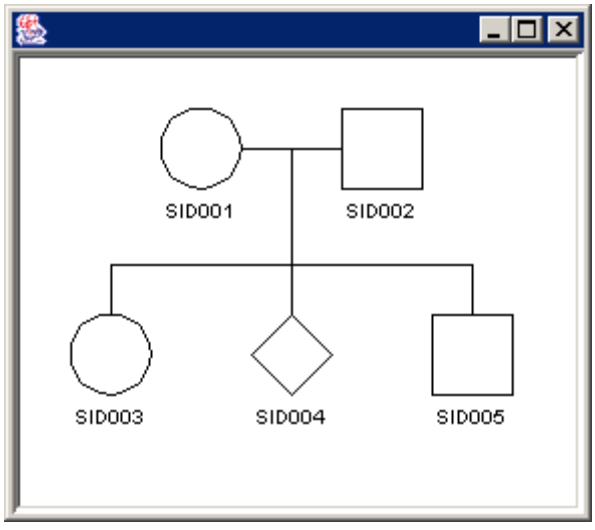
Constructing a SwingPedigreeViewer

SwingPedigreeViewer, a subclass of JComponent, is the Swing UI component that is responsible for displaying the pedigree. It also forwards events generated through user interaction with the pedigree to those objects regestered as listener for those events.

SwingPedigreeViewer is constructed and configured to display a pedigree with the following code:

```
SwingPedigreeViewer swingPedigreeViewer = new SwingPedigreeViewer();
swingPedigreeViewer.setPedigree(pedigree);
```

With the addition of two more children, this component will appear like the following figure:



Configuring the Pedigree Display

A SwingPedigreeViewer by default will show a very minimal amount of information for each subject displayed within the pedigree. This display may be formatted to reflect the desired display through manipulating the DefaultPedigreeViewerStandardFormatter object associated with the SwingPedigreeViewer object.

The DefaultPedigreeViewerStandardFormatter has a number of power means of configuring the pedigree display. The following code demonstrates a fraction of how it may be employed; you should reference the Javadoc-based documentation for a more full treatment of this topic.

First, acquire a reference to the DefaultPedigreeViewerStandardFormatter:

```
DefaultPedigreeViewerStandardFormatter formatter =
(DefaultPedigreeViewerStandardFormatter)((PedigreeRendererStandard)
pedigreeViewer.getPedigreeRenderer()).getFormatter();
```

Next, instantiate a SubjectPainter object. This is the object that will be employed in rendering each subject within the pedigree.

```
DefaultSubjectPainter subjectPainter = new DefaultSubjectPainter();
```

The background color of a subject is set to light gray.

```
subjectPainter.setBackgroundColor(Color.lightGray);
```

The display of a subject is handled by a seried of annotations. Annotations may be symbols, labels, or haplotypes.

The following example shows a Subject configured to be displayed as a Symbol with an adornment, its label, and its favorite color.

```
Vector subjectAnnotations = new Vector();

SubjectAnnotationSymbol subjectAnnotationSymbol = new
SubjectAnnotationSymbol();

Vector adornments = new Vector();
SymbolAdornment symbolAdornment = new
SymbolAdornment(SubjectAnnotationSymbol.atiFull);
symbolAdornment.setPropertyName("affected");
symbolAdornment.setDefaultColor(Color.white);
symbolAdornment.addValueColor("no",Color.green);
symbolAdornment.addValueColor("yes",Color.red);
adornments.addElement(symbolAdornment);

subjectAnnotationSymbol.setSymbolAdornments(adornments);
subjectAnnotations.addElement(subjectAnnotationSymbol);
subjectAnnotations.addElement(new SubjectAnnotationLabel());
subjectAnnotations.addElement(new
SubjectAnnotationProperty("favorite-color","Color: "));

subjectPainter.setSubjectAnnotations(subjectAnnotations);
```

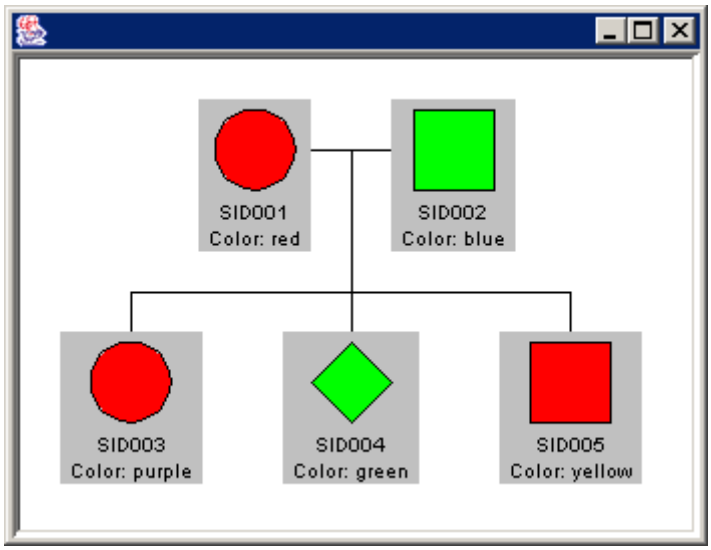
The following example sets the symbol to be the annotation that should be treated as the center point for positioning within the pedigree.

```
subjectPainter.setCenteredAnnotation((SubjectAnnotationIF)subjectAn
notations.firstElement());
```

Lastly, the following code configures the formatter to use this SubjectPainter in rendering each Subject.

```
formatter.setSubjectPainter(subjectPainter);
```

With this formatter, the previous pedigree is displayed like the following figure:



Configuring Interaction and Event Listeners

SwingPedigreeViewer may be placed in four interaction modes:

- Selection
- Translation
- Zoom
- None

Configuring Selection Mode

The user may select subjects with the mouse by either clicking on individual subjects, or dragging a box to select groups of subjects. The application may receive notification of selection activity by registering with SwingPedigreeViewer as a SubjectSelectListener. Use the following code to configure Selection mode:

```
pedigreeViewer.setInteractionMode(PedigreeViewer.InteractionModes.SELECT);
```

Configuring Translation Mode

The user may pan the pedigree display by clicking anywhere on the component and dragging the mouse. The application may receive notification of this interaction by registering as a TranslationUpdateListener. Use the following code to configure Translation mode:

```
pedigreeViewer.setInteractionMode(PedigreeViewer.InteractionModes.TRANSLATE);
```

Configuring Zoom Mode

The user may zoom into or out of the pedigree display by clicking and/or dragging with the right or left mouse button. The application may receive notification of this interaction by registering as a ZoomListener. Use the following code to configure Zoom mode:

```
pedigreeViewer.setInteractionMode(PedigreeViewer.InteractionModes.Z
OOM);
```

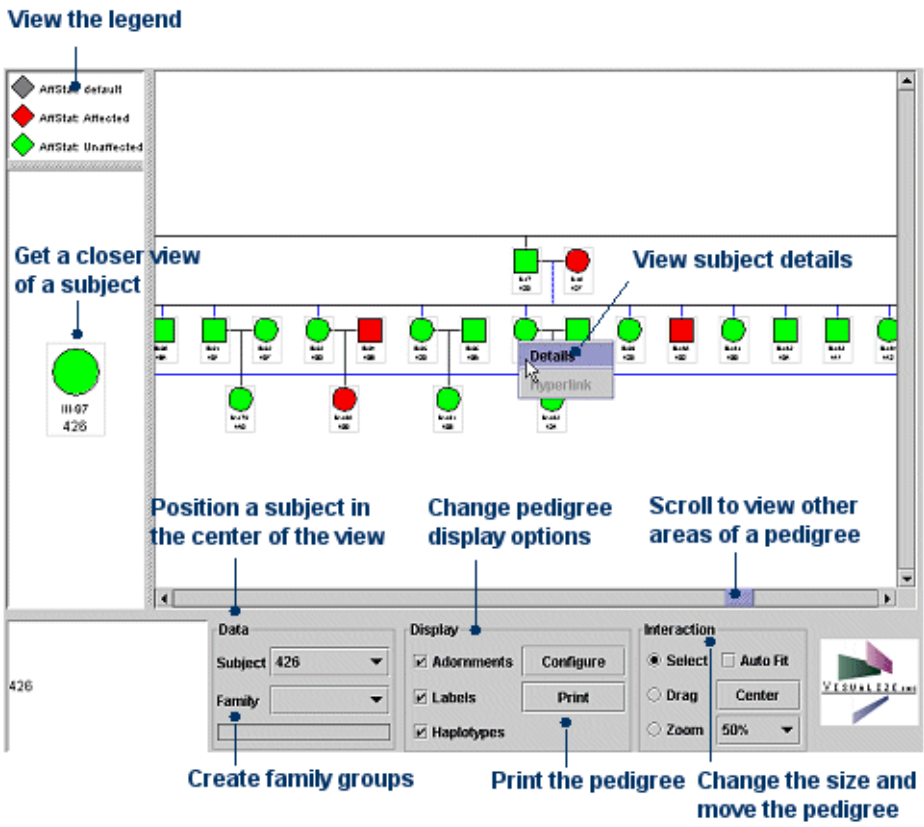
Viewing an Application Component Configuration Example

You can find the complete example from which the sample code in this section was extracted in the PedigreeApp.java and PedigreeFrame.java files in the examples folder.

6 User's Guide

Overview of Using DataVista Pedigree

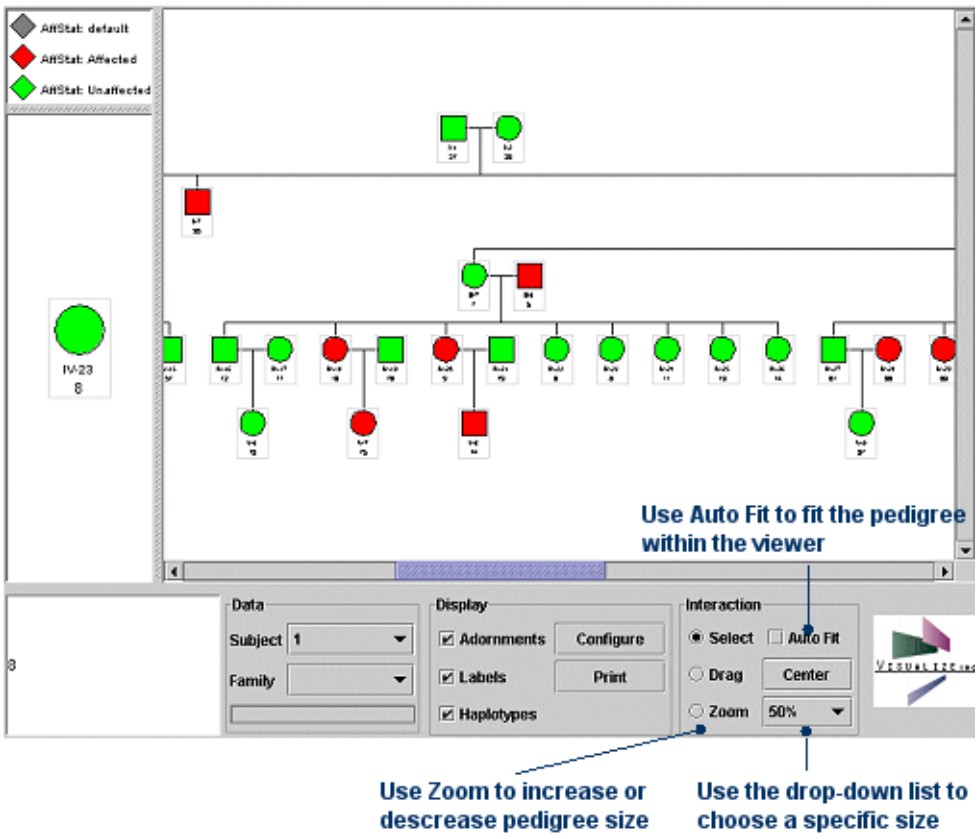
DataVista Pedigree (DVP) allows you to explore relationships among genetic data and quickly narrow in on specific information in pedigrees ranging from a few subjects to hundreds. You can change a pedigree size, move a pedigree, view subject details, change pedigree display options, and print.



Changing Pedigree Size

DVP makes working with any size of pedigree simple by allowing you to change the size of pedigrees.

You can increase and decrease the size of a pedigree by using the Auto Fit, Zoom, or Size drop-down list features.

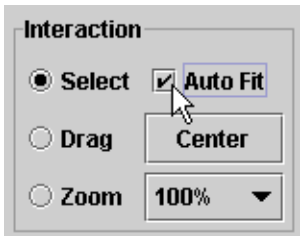


Using the Auto Fit Feature

The Auto Fit feature forces the entire pedigree to fit within the viewer.

To change a pedigree size with the Auto Fit feature

- Select the Auto Fit checkbox.

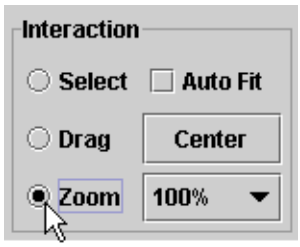


Using the Zoom Feature

The Zoom feature allows you to increase and decrease the pedigree size by using the mouse.

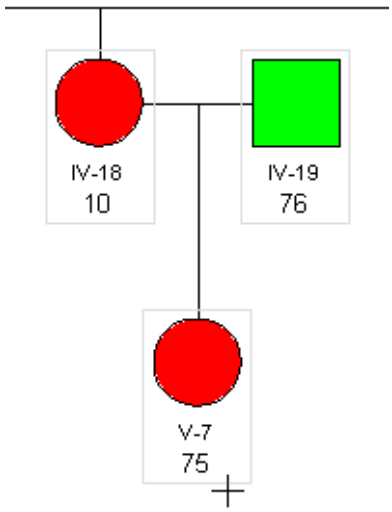
To change a pedigree size with the Zoom feature

1 Select the Zoom option.



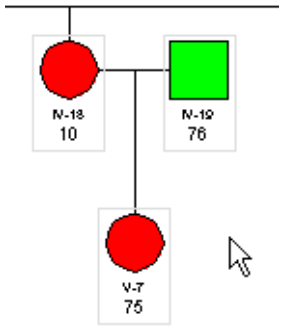
2 Do one of the following:

– To increase pedigree size, click on the pedigree with the left mouse button.



- or -

– To decrease pedigree size, click on the pedigree with the right mouse button.

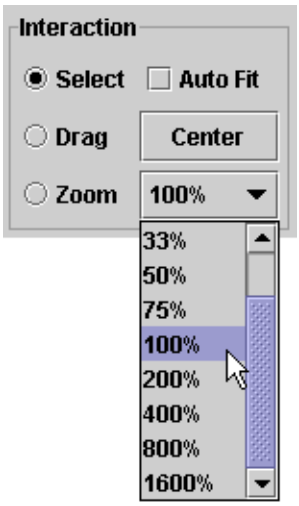


Using the Size Drop-down List

You can use a drop-down list to choose a specific size for the pedigree, such as 50%, 100%, and 400%.

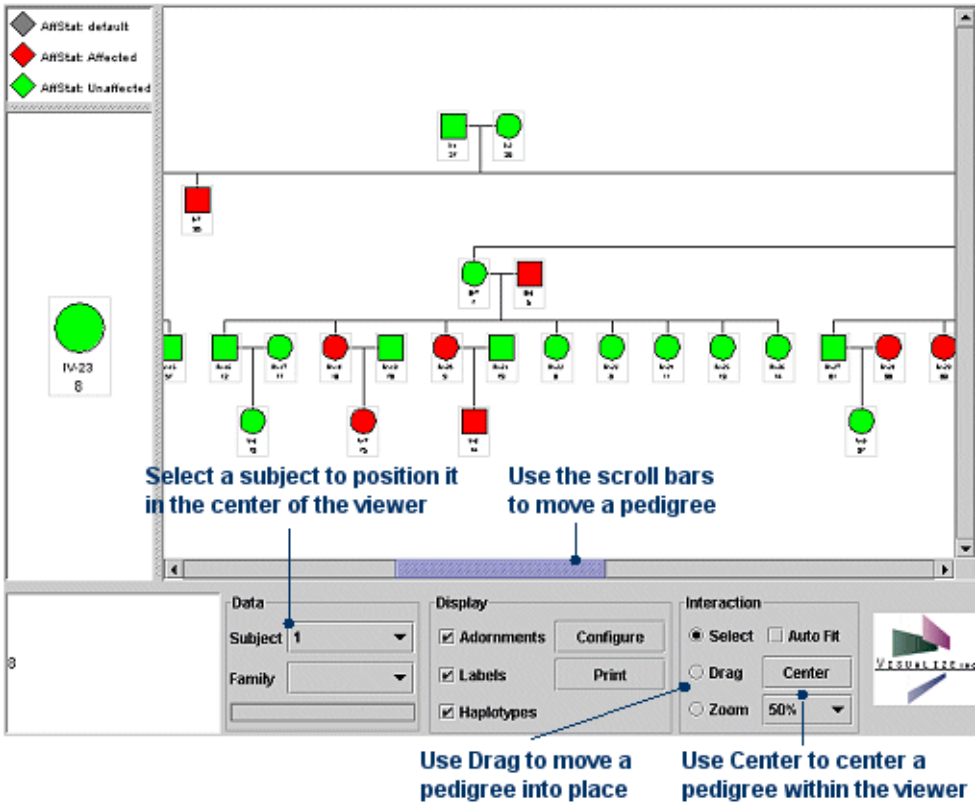
To change a pedigree size with the drop-down list

■ Select a size from the drop-down list.



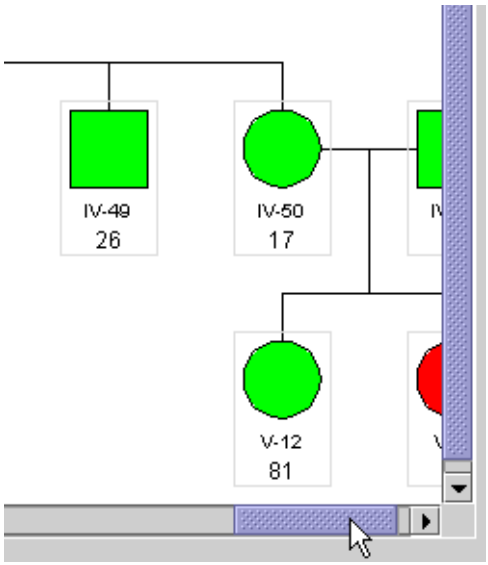
Moving Pedigrees

In addition to using the scroll bars to move a pedigree, you can use the Drag or Center feature. You can also position a particular subject in the center of the viewer by selecting a subject from the Data drop-down list.



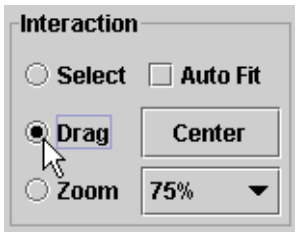
To move a pedigree

- Use the scroll bars to view different areas of the pedigree.



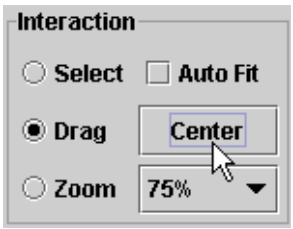
- or -

- Select the Drag option, click on the pedigree, and drag it into place.



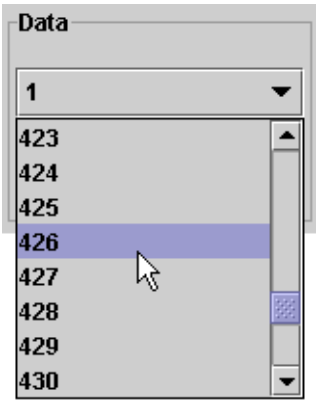
- or -

- Click the Center button to position the center of the pedigree in the center of the viewer.



- or -

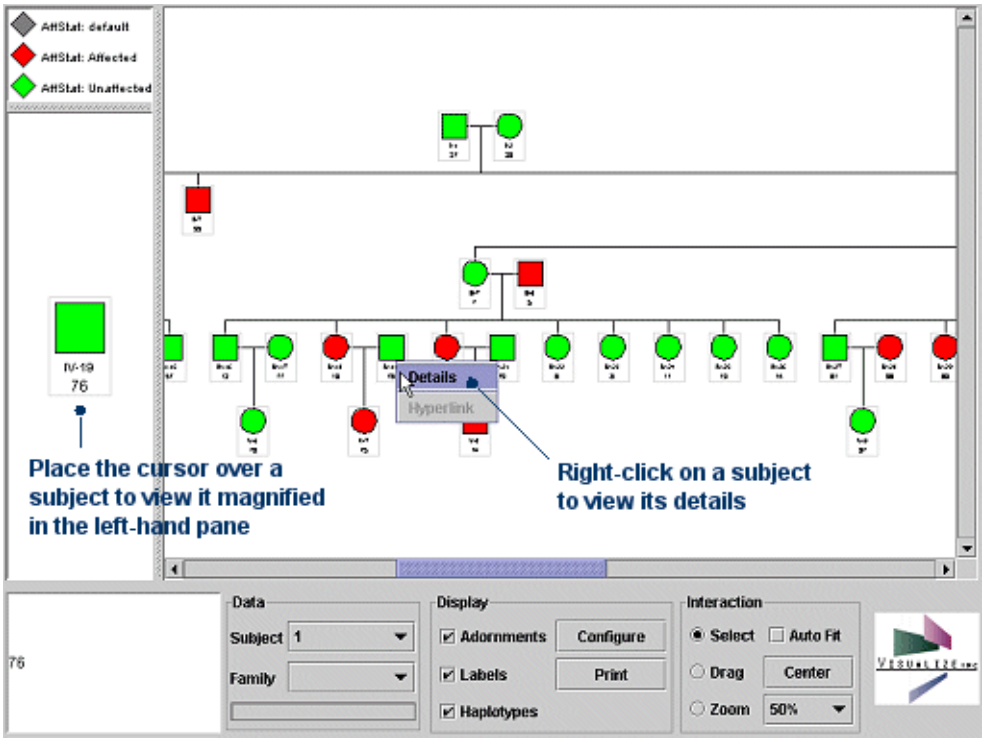
- Select a subject from the drop-down list in the Data area to position that subject in the center of the viewer.



Viewing Subject Details

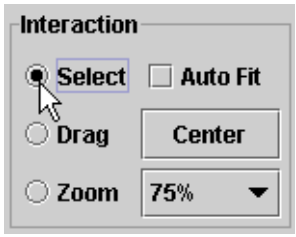
You can get a closer look at a subject when you place the cursor over a subject. The subject information appears magnified in the left-hand side of the viewer. This feature is especially helpful when you are viewing a decreased size of a large pedigree.

You can also view all subject details by using the Select feature.

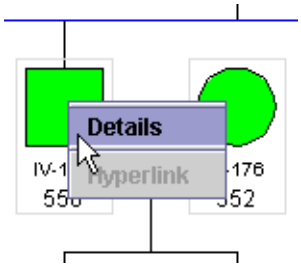


To view subject details:

1 In the Interaction area, select the Select option.

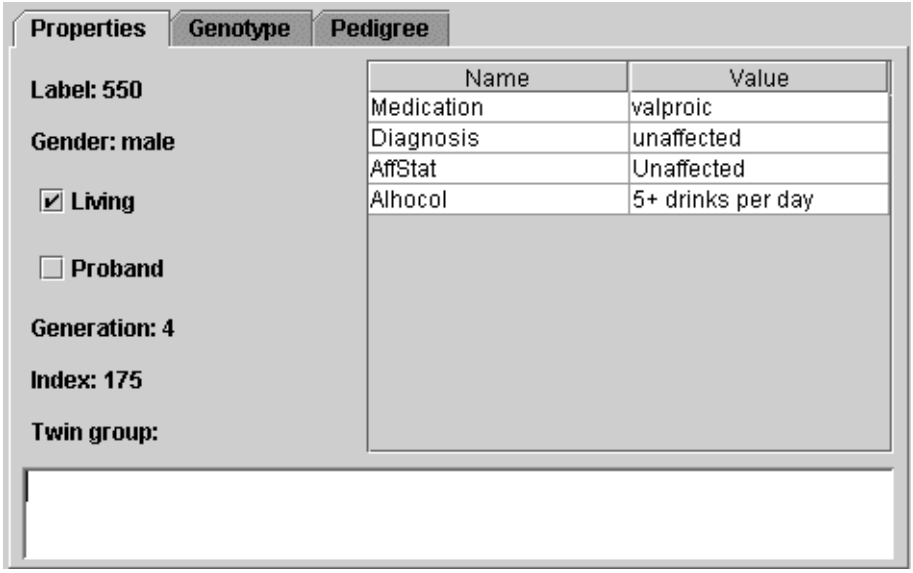


2 Right-click on a subject and choose Details from the short-cut menu.

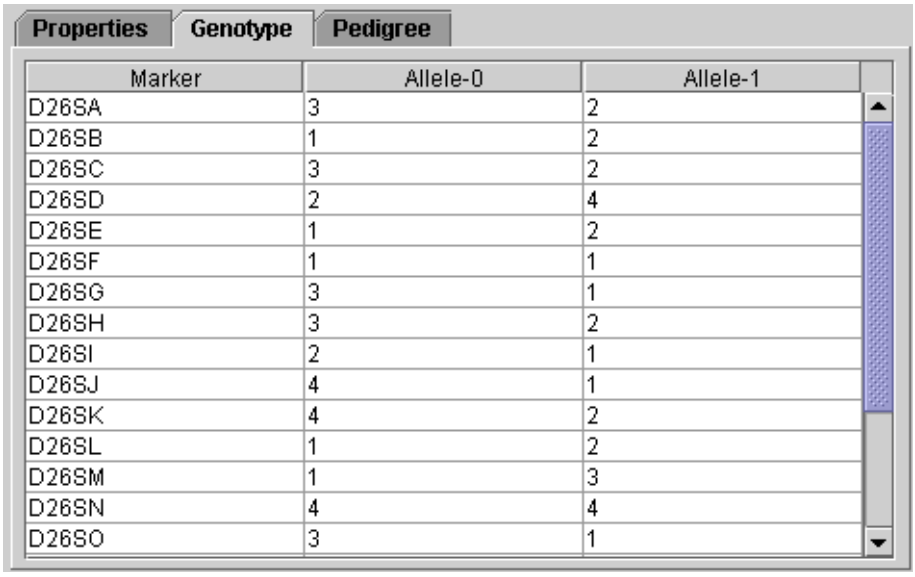


The window that appears contains three tabs of information about the subject:

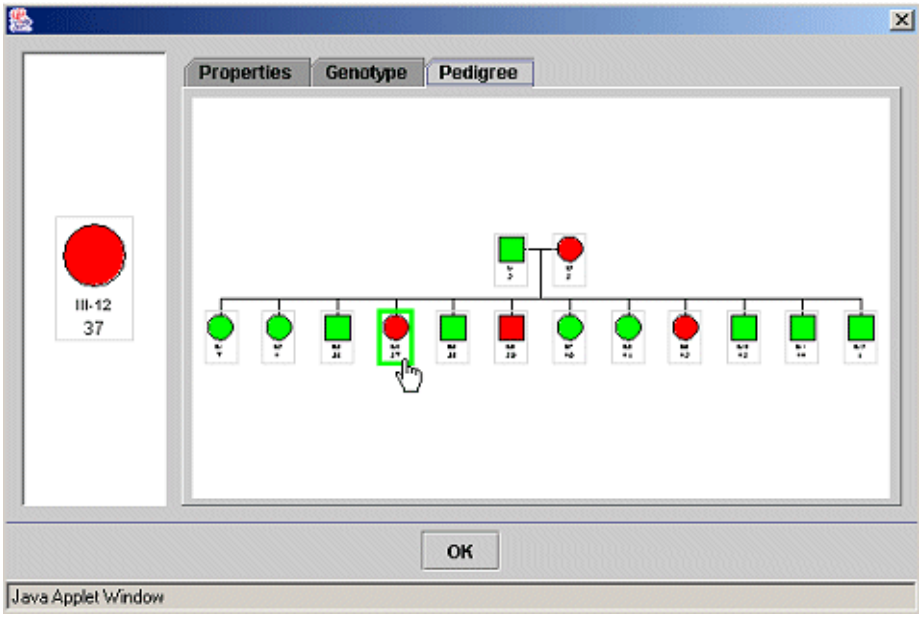
- Properties–displays detailed genetic and clinical data about the subject.



- Genotype–displays detailed data about the subject's genotype.



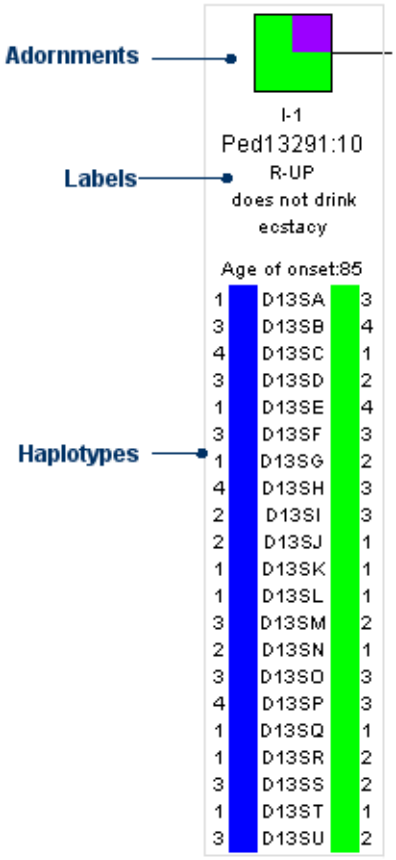
- Pedigree–shows the subject's "first-degree relatives." By clicking on a subject, you can view the subject's information magnified in the left-hand side of the window.



Changing Display Options

You can modify pedigree appearance by choosing whether or not to display the following annotations:

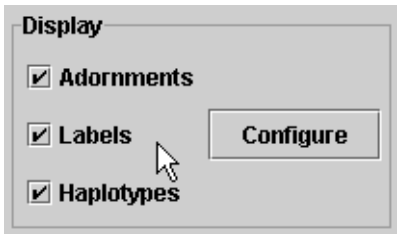
- Adornments–graphically display subject data within the subject symbol.
- Labels–textually display subject data underneath the subject.
- Haplotypes–graphically and textually display subject haplotype data underneath the subject.



In addition to choosing whether or not to display these annotations, you can further configure annotation appearance including fonts, colors, and additional data to display.

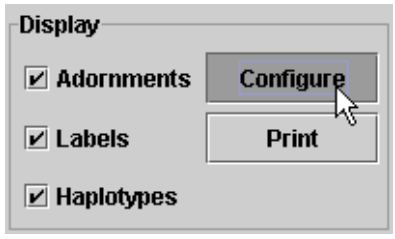
To choose whether or not to display the adornment, label, and haplotype annotations

- In the Display area, select or clear the Adornments, Labels, and Haplotypes options.

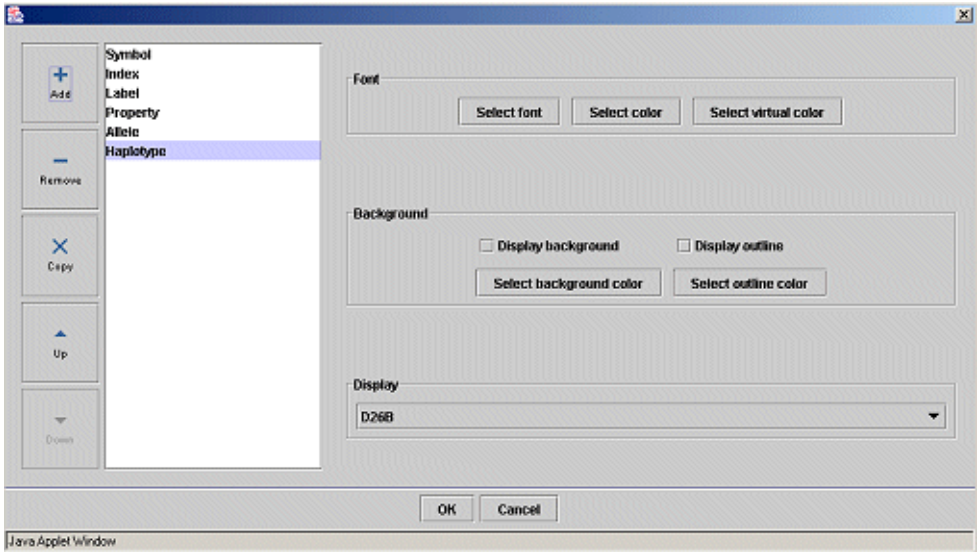


Configuring Annotations

You can further configure the appearance of pedigrees by clicking the Configure button in the Display area.



The configuration window allows you to configure all the pedigree annotations.



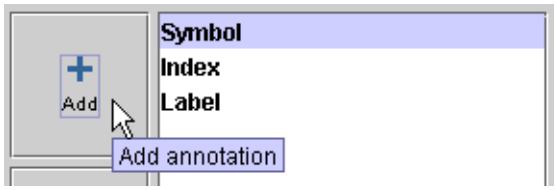
You can configure the following attributes of the pedigree annotations:

- Symbols–You can add and modify adornments within symbols. Symbols are the squares, circles, or diamonds that represent subjects. Adornments are the colored regions within symbols that may span a quarter, a half, or the entire region of the symbol. Adornments display data graphically within the symbol.
- Labels–You can modify the font and background of labels.
- Index–You can modify the font and background of the id number for the subject, as well as choose the type of information to display.
- Properties–You can modify the font and background of subject data, as well as choose the type of information to display.
- Alleles–You can modify the font and background of allele data, and also choose the type of information to display.
- Haplotypes–You can modify the font and background of haplotype data, as well as choose the type of information to display.

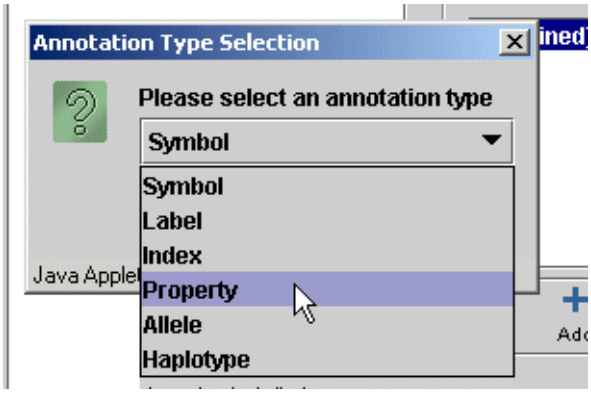
If an annotation doesn't appear in the list of annotations, you can add it.

To add an annotation

1 Click the Add button near the list of annotations.



2 Select an annotation type from the drop-down list, and then click the OK button.

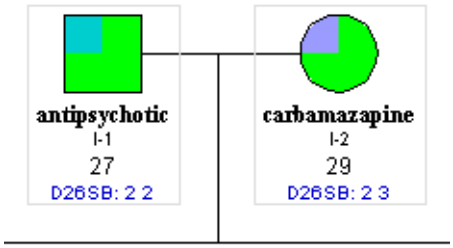


3 Modify the annotation attributes depending on the type of annotation.

You modify annotations attributes by selecting an annotation from the list and then modifying the fonts, backgrounds, data to display, or adornments (for Symbols).

Modifying Fonts

You can modify the style and color of fonts for labels, indexes, properties, alleles, and haplotypes.

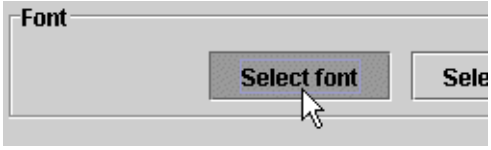


To modify fonts

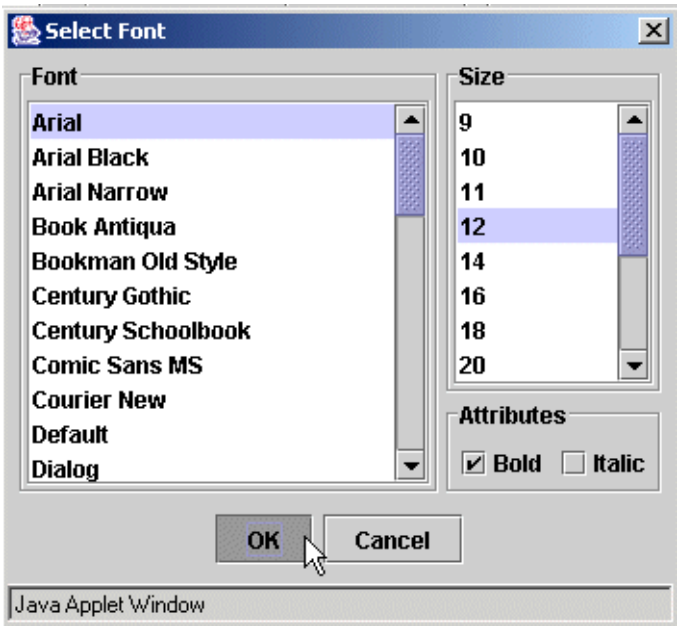
1 Select an annotation from the list of annotations.

2 Complete the following steps to select the font type and style:

a Click the Select Font button.



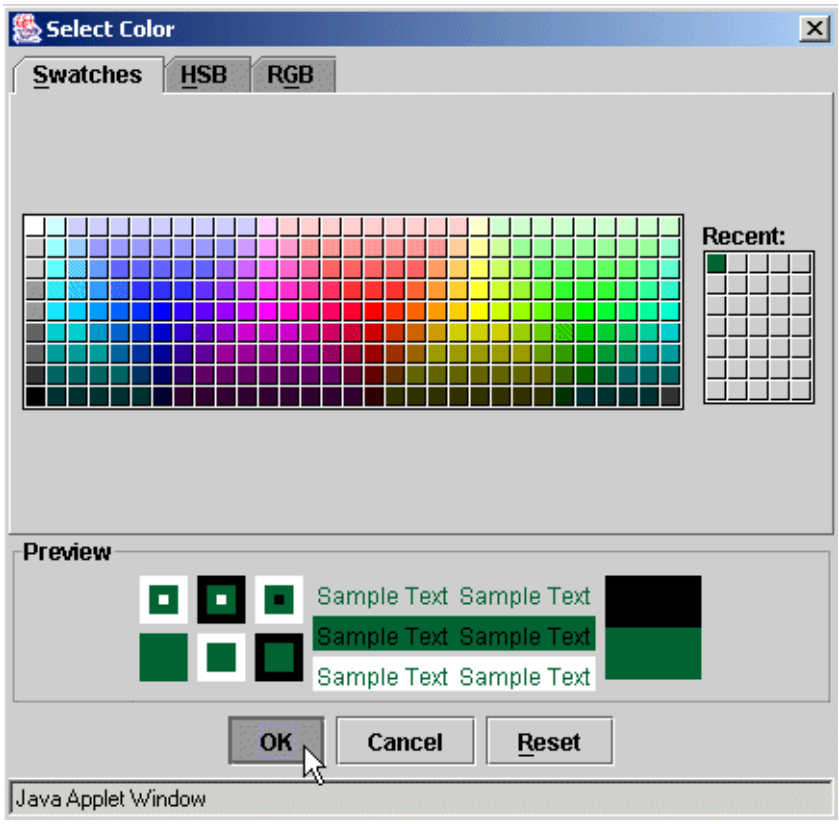
b Select the font type and style, and then click the OK button.



3 Complete the following steps to select the font or virtual color:

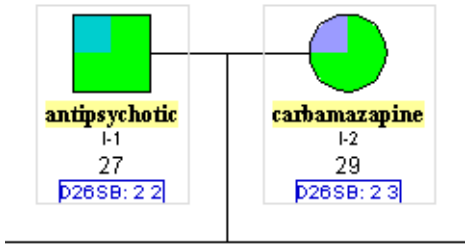
a Click the Select Color or Select Virtual Color button.

b Select the color, and then click the OK button.



Modifying Backgrounds

You can modify the background fill and outline color for labels, indexes, properties, alleles, and haplotypes.



To modify backgrounds

1. Select an annotation type from the list of annotations.
2. Complete the following steps to select a background or outline color:
   a. Click the Select Background Color or Select Outline Color button.

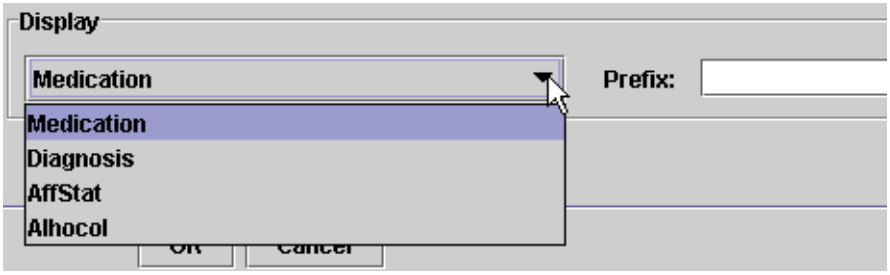
b Select the color, and then click the OK button.

3 Select the Display Background or Display Outline checkbox.

To modify data to display

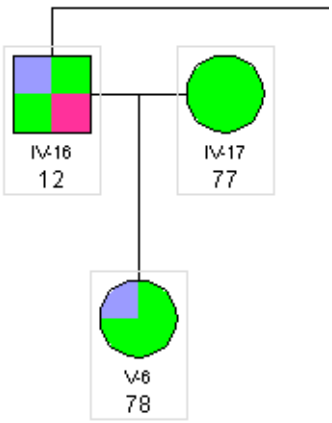
1 Select an annotation type from the list of annotations.

2 Select the data to display according to the type of annotation.



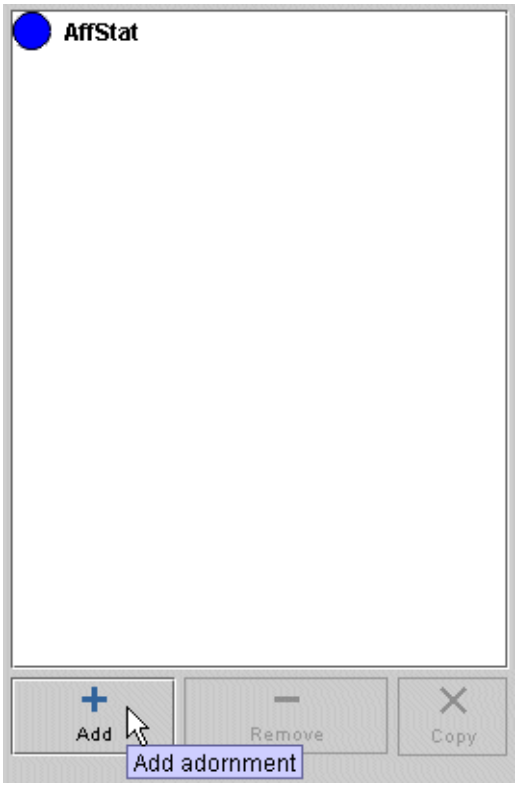
Adding and Modifying Adornments

You can modify the colors and regions for symbol adornments.

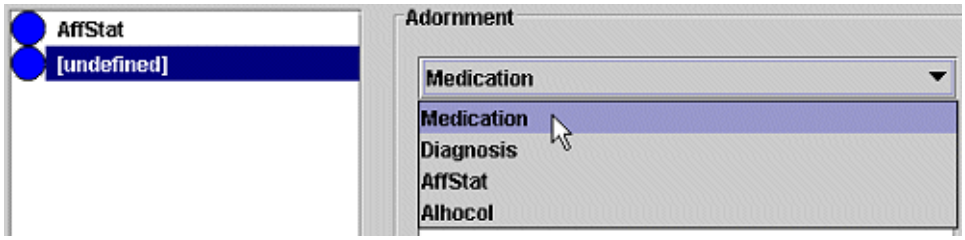


To add an adornment

1 After selecting Symbol from the list of annotations, click the Add button under the list of adornments.



2 Select an adornment type from the drop-down list.

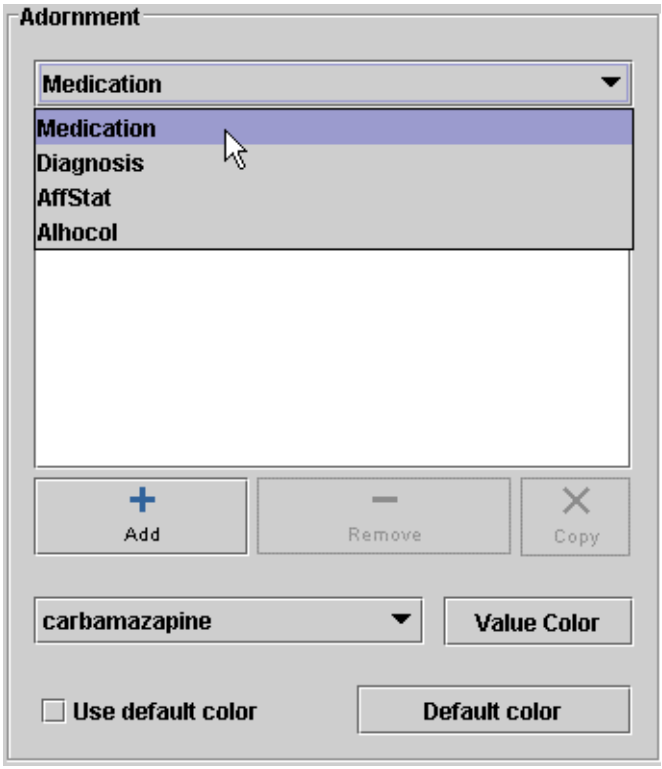


3 Modify the adornment attributes depending on the adornment type.

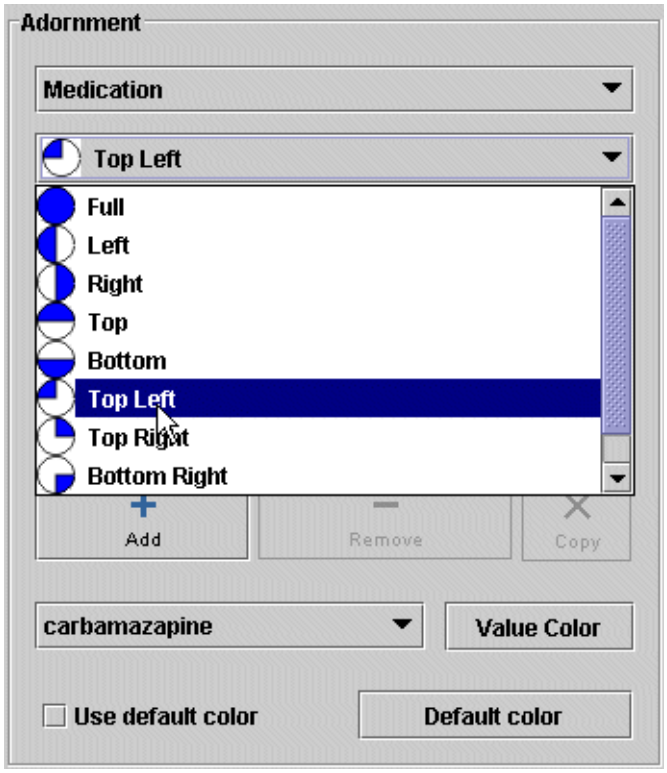
You modify adornment attributes by selecting an adornment from the list and then modifying the region, value, and color.

To modify adornments

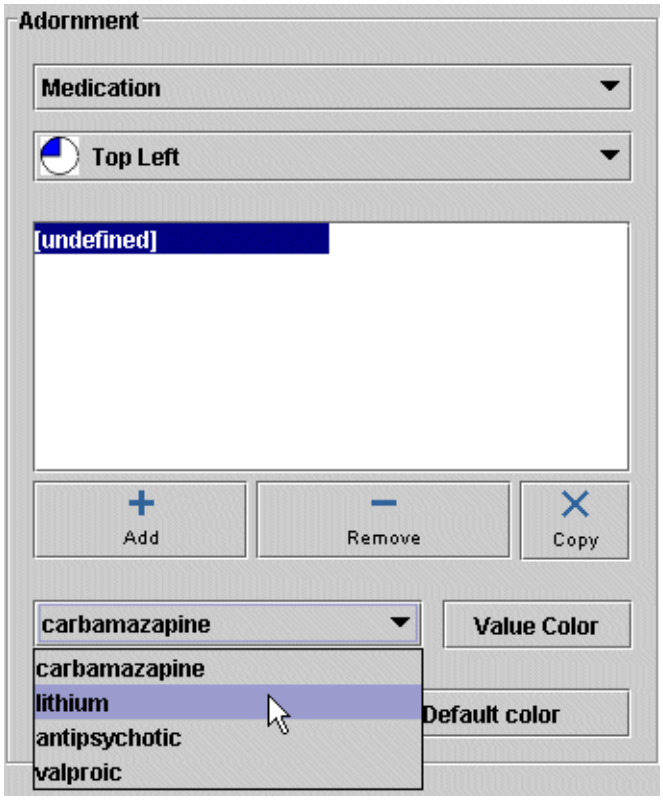
1 Select an adornment from the list of adornments.



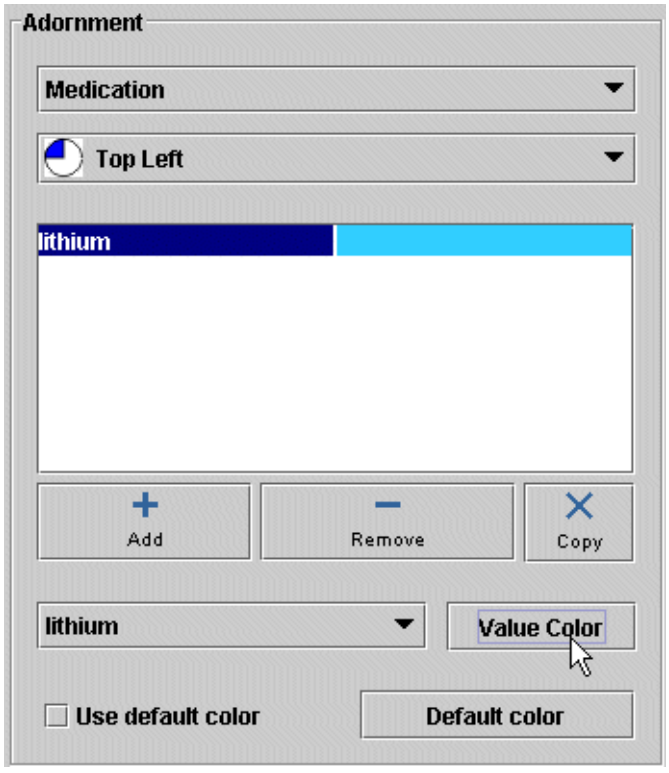
2 Select the region of the symbol to display this attribute from the drop-down list.



3 Select a value for this attribute by clicking the Add button, and then selecting a value from the drop-down list.

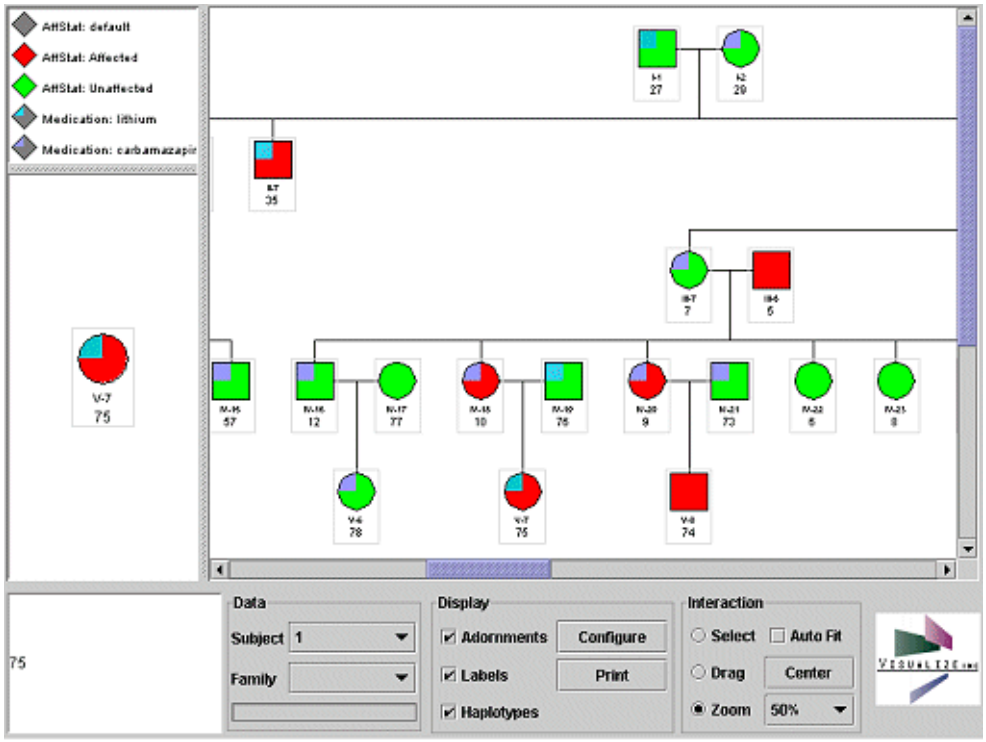


4 Select a color for this attribute by clicking the Value Color button, selecting a color, and then clicking the OK button.



5 Click the OK button.

Repeat steps 3 through 5 to add additional values.

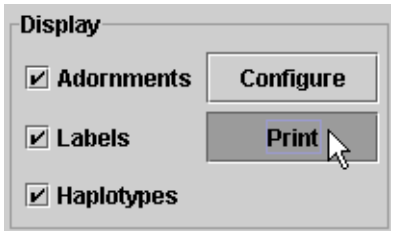


Printing Pedigrees

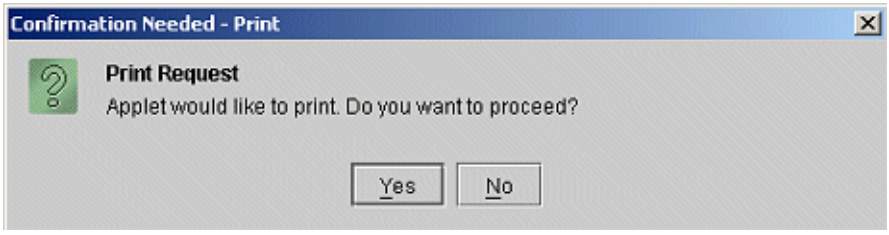
You can print the entire pedigree by using the Print button and your browser's Print function.

To print a pedigree

1 Click the Print button.



2 In the Confirmation dialog, click the Yes button.



3 In your browser's Print windows, select the print options, and then click the OK buttons.

4 In the Confirmation dialog, click the Yes button.

7 XML Reference

DataVista Pedigree XML Elements

This appendix organizes a list of elements within each XML block (top-level elements):

- <licensing>–details licensing information for DVP.
- <options>–defines interaction aspects of DVP.
- <viewer>–defines pedigree appearance.
- <dataset>–defines the pedigree data, including the family structure, phenotypic, and genotypic data.

Elements appear in the list in the order in which they should be appear in the configuration file.

Licensing Block Elements

Table 7-1 <licensing> Elements

Element	Attributes	Values	Parents	Children	Description
<keyword>		String	<licensing>		License keyword.
<license>		String	<licensing>		License code.
<servers>		List of strings	<licensing>		Servers covered under the license.
<expiration>		String	<licensing>		License expiration date.

Options Block Elements

Table 7-2 <options> Elements

Element	Attributes	Values	Parents	Children	Description
<handlers>			<options>	<handler>	The handlers.
<handler>			<options><handlers>	<linktarget>, <url-property>, <name>	The behavior of the applet when a subject is right-clicked and the desired handler selected.
	type	hyperlink	<options><handlers><handler>		Causes the browser to jump to a web page related to the selected subject.
<linktarget>		String	<options><handlers><handler>		The behavior of the target web page. Valid for hyperlink handler type.
<url-property>		String	<options><handlers><handler>		The name of the property that determines the address of the target Web page.
<name>		String	<options><handlers><handler>		The name of the handler that is displayed in the context menu.

Viewer Block Elements

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<pedigree-renderer>			<viewer>	<subject>, <detailed-subject>, <mating>, <lineage>, <overlapped-children-spacing>, <overlapped-mating-spacing>, <row-spacing>, <column-spacing>	Contains the pedigree display configuration.
<subject>		subelements	<viewer><pedigree-renderer>	<insets>, <background-color>, <virtual-color>, <outline-color>, <selected-color>, <highlighted-color>, <annotations>	Defines how to display subjects.
<insets>		integer (top) integer (left) integer (bottom) integer (right)	<viewer><pedigree-renderer><subject>		The padding to surround the subject.
<background-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject>		The color of the background area behind a subject.
<virtual-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject>		The outline color that should be employed when the subject is virtual.
<outline-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject>		The color of the outline.
<selected-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject>		The color of the outline when the subject is selected.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<highlighted-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject>		The color of the outline when the subject is highlighted.
<annotations>			<viewer><pedigree-renderer><subject>	<annotation>	Defines how to display subject data.
<annotation>		subelements	<viewer><pedigree-renderer><subject><annotations>	<marker-name>, <font>, <haplotype>	The annotations.
	centered	true | false	<viewer><pedigree-renderer><subject><annotations><annotation>		Defines whether or not to place the annotations centered.
	type	haplotype | index | property | symbol | label | allele	<viewer><pedigree-renderer><subject><annotations><annotation>		The type of annotation.
	annotation type=haplotype		<viewer><pedigree-renderer><subject><annotations><annotation>	<font>, <text-color>, <insets>, <background-color>, <outline-color>, <draw-background>, <draw-outline>, <mutation-color>, <indeterminate-color>, <show-alleles>, <id>, <haplotype-color>	Defines haplotypes.
<font>		subelements	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>	<family>, <size>, <style>	The font.
<family>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype><font>		The font family.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<size>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype><font>		The font size.
<style>		italic | bold | plain	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype><font>		The font style.
<text-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		The text color.
<insets>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		The padding to surround this annotation.
<background-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		The background color.
<outline-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		The outline color.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<draw-background>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		Defines whether the background should be colored.
<draw-outline>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		Defines whether an outline should be drawn around the haplotype.
<mutation-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		The color of the X displayed for mutation sites.
<indeterminate-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		The color of the vertical bar displayed for indeterminate values.
<show-alleles>		true | false	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		Defines whether to show alleles.
<id>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		The id of the haplotype to display.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<haplotype-color>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=haplotype>		The haplotype color.
	annotation type=index	subelements	<viewer><pedigree-renderer><subject><annotations><annotation>	<print-generation>, <print-index>, <font>, <color>, <virtual-color>, <insets>, <background-color>, <outline-color>, <draw-background>, <draw-outline>	The index display.
<print-generation>		true | false	<viewer><pedigree-renderer><subject><annotations><annotation type=index>		Defines whether the generation index will be displayed.
<print-index>		true | false	<viewer><pedigree-renderer><subject><annotations><annotation type=index>		Defines whether the individual index will be displayed. The individual index is the position of the subject from left to right.
<insets>		integer (top) integer (left) integer (bottom) integer (right)	<viewer><pedigree-renderer><subject><annotations><annotation type=index>		Four numbers defining the size of the padding around the index.
<background-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=index>		The index background color.
<outline-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=index>		The index outline color.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<font>		subelements	<viewer><pedigree-renderer><subject><annotations><annotation type=index>	<family>, <size>, <style>	The index font.
<family>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=index><font>		The index font family.
<size>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=index><font>		The index font size.
<style>		italic | bold | plain	<viewer><pedigree-renderer><subject><annotations><annotation type=index><font>		The index font style.
<color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation>type=index		The index color.
<virtual-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=index>		The color of the text when the subject is virtual.
<draw-background>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=index>		Defines whether the background should be colored.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<draw-outline>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=index>		Defines whether an outline should be drawn around the index.
	annotation type=property		<viewer><pedigree-renderer><subject><annotations><annotation>	<property-name>, <prefix>	The display of properties.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=property>		The name of the property whose value should be displayed.
<prefix>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=property>		The text that should be displayed before the property value.
	annotations type=symbol	subelements	<viewer><pedigree-renderer><subject><annotations><annotation>	<insets>, <size>, <background-color>, <outline-color>, <legend-font>, <legend-color>, <legend-insets>, <adornments>	The symbols.
<insets>		integer (top) integer (left) integer (bottom) integer (right)	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>		Four numbers defining the size of the padding around the symbol.
<size>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>		The symbol size.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<background-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>		The symbol background color.
<outline-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>		The symbol outline color.
<legend-font>		subelements	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<family>, <size>, <style>	The legend font.
<family>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><legend-font>		The legend font family.
<size>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><legend-font>		The legend font size.
<style>		italic | bold | plain	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><legend-font>		The legend font style.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<legend-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>		The legend color.
<legend-insets>		integer (top) integer (left) integer (bottom) integer (right)	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>		Four numbers defining the size of the padding around the legend entry.
<adornments>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<full>, <left>, <right>, <top>, <bottom>, <topleft>, <topright>, <bottomright>, <bottomleft>	The adornments.
<full>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “full” region.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>I<full>		The adornment property name.
<use-default-color>		true | false	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><full>		Defines whether to use the default adornment color.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><full>		The adornment color.
	value	string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><full><color>		The value of the color.
<left>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “left” region.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><left>		The adornment property name.
<use-default-color>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><left>		Defines whether to use the default adornment color.
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><left>		The adornment color.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
	value	string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><left><color>		The color value.
<right>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “right” region.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><right>		The adornment property name.
<use-default-color>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><right>		Defines whether to use the default adornment color.
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><right>		The adornment color.
	value	string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><right><color>		The color value.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<top>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “top” region.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><top>		The adornment property name.
<use-default-color>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><top>		Defines whether to use the default adornment color.
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><top>		The adornment color.
<bottom>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “bottom” region.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottom>		The adornment property name.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<use-default-color>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottom>		Defines whether to use the default adornment color.
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottom>		The adornment color.
<topleft>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “topleft” region.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><topleft>		The adornment property name.
<use-default-color>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><topleft>		Defines whether to use the default adornment color.
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><topleft>		The adornment color.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
	value	string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><topleft><color>		The color value.
<topright>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “topright” region.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><topright>		The adornment property name.
<use-default-color>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><topright>		Defines whether to use the default adornment color.
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><topright>		The adornment color.
	value	string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><topright><color>		The color value.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<bottomright>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “bottomright” region.
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottomright>		The adornment property name.
<use-default-color>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottomright>		Defines whether to use the default adornment color.
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottomright>		The adornment color.
	value	string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottomright><color>		The color value.
<bottomleft>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol>	<property-name>, <use-default-color>, <color>	The adornment in the “bottomleft” region.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<property-name>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottomleft>		The adornment property name.
<use-default-color>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottomleft>		Defines whether to use the default adornment color.
<color>			<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottomleft>		The adornment color.
	value	string	<viewer><pedigree-renderer><subject><annotations><annotation type=symbol><bottomleft><color>		The color value.
	annotation type=label		<viewer><pedigree-renderer><subject><annotations><annotation type=label>	<font>, <color>, <virtual-color>, <insets>, <background-color>, <outline-color>, <draw-background>, <draw-outline>	The annotation text.
<font>			<viewer><pedigree-renderer><subject><annotations><annotation type=label>	<family>, <size>, <style>	The font to be used in displaying the label text.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<family>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=label><font>		The label text font family.
<size>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=label><font>		The label text font size.
<style>		italic | bold | plain	<viewer><pedigree-renderer><subject><annotations><annotation type=label><font>		The label text font style.
<color>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=label>		The label text font color.
<virtual-color>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=label>		The label text font virtual color.
<insets>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=label>		The padding to surround the label text.
<background-color>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=label>		The label text font background color.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<outline-color>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=label>		The label text font outline color.
<draw-background>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=label>		Defines whether the background should be colored.
<draw-outline>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=label>		Defines whether an outline should be drawn around the label.
	annotation type=allele		<viewer><pedigree-renderer><subject><annotations><annotation>	<marker-name>, <font>, <family>, <size>, <style>, <color>, <virtual-color>, <insets>, <background-color>, <outline-color>, <draw-background>, <draw-outline>	The display of allele data.
<marker-name>		String	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		The name of the marker whose allele should be displayed.
<font>		<font> subelements	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		The font to be used in displaying the text.
<family>		string	<viewer><pedigree-renderer><subject><annotations><annotation type=allele><font>		The text font family.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<size>		integer	<viewer><pedigree-renderer><subject><annotations><annotation type=allele><font>		The text font size.
<style>		italic | bold | plain	<viewer><pedigree-renderer><subject><annotations><annotation type=allele><font>		The text font style.
<color>		integer (red) integer (green) integer (blue) integer (triplet)integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		The color of the text.
<virtual-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		The color of the text when the subject is virtual.
<insets>		integer (top) integer (left) integer (bottom) integer (right)	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		The padding to surround the allele.
<background-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		The background color.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<outline-color>		integer (red) integer (green) integer (blue) integer (triplet)	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		The outline color.
<draw-background>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		Defines whether the background should be colored.
<draw-outline>		Boolean	<viewer><pedigree-renderer><subject><annotations><annotation type=allele>		Defines whether an outline should be drawn around the allele.
<detailed-subject>			<viewer><pedigree-renderer>		The detail subject information.
<mating>			<viewer><pedigree-renderer>	<color>	The appearance of the line connecting subjects in a mating.
<color>		string	<viewer><pedigree-renderer><mating>		The color of the line connecting subjects in a mating.
<lineage>			<viewer><pedigree-renderer>	<color>	The appearance of the line connecting children to their parents.
<color>		string	<viewer><pedigree-renderer><lineage>		The color of the line connecting children to their parents.
<overlapped-children-spacing>		integer	<viewer><pedigree-renderer>		The space inserted between lines that connect overlapping children.
<overlapped-mating-spacing>		integer	<viewer><pedigree-renderer>		The space inserted between lines that connect overlapped matings.

Table 7-3 <viewer> Elements

Element	Attributes	Values	Parents	Children	Description
<row-spacing>		integer	<viewer><pedigree-renderer>		The space between generational rows.
<column-spacing>		integer	<viewer><pedigree-renderer>		The space between columns.

Dataset Block Elements

Table 7-4 <dataset> Elements

Element	Attributes	Values	Parents	Children	Description
<updateinterval>		integer	<dataset>		The interval, in milliseconds, between updates to the Pedigree data.
<pedigree>			<dataset>		Defines pedigree data.
	source	static | servlet | csv | jdbc |	<dataset><pedigree>		The type of datasource.
	source=static or servlet		<dataset><pedigree source=static or servlet>	<config>, <configxml>, <verbose>, <compressed>, <twin-group-field>, <family-group-field>, <subject>	Indicates a static or servlet datasource.
<config>			<dataset><pedigree source=static or servlet>		Defines the XML configuration.
<configxml>			<dataset><pedigree source=static or servlet>		The XML to be passed to the data servlet.
<verbose>			<dataset><pedigree source=static or servlet>		Whether the logging will be verbose.
<compressed>			<dataset><pedigree source=static or servlet>		Whether data will be compressed when sent from servlet to client.
<twin-group-field>		integer	<dataset><pedigree source=static or servlet>		The index of the field that contains the twin group for a subject.
<family-group-field>		integer	<dataset><pedigree source=static or servlet>		The index of the field that contains the family group that this subject is a member of.
<subject>		subelements	<dataset><pedigree source=static or servlet>	<label>, <description>, <parents>, <gender>, <properties>, <living>, <proband>, <twin-group>, <genotype>, <haplotype-assignment>	Defines subject data.

Table 7-4 <dataset> Elements

Element	Attributes	Values	Parents	Children	Description
	id	integer	<dataset><pedigree source=static or servlet><subject>		The id of the subject.
<label>		String	<dataset><pedigree source=static or servlet><subject>		The text that should be employed in labeling the subject in the pedigree view.
<description>		String	<dataset><pedigree source=static or servlet><subject>		Defines descriptive text pertaining to the subject.
<parents>		List of subject ids	<dataset><pedigree source=static or servlet><subject>		The ids of the parents.
<gender>		male | female | unknown	<dataset><pedigree source=static or servlet><subject>		The subject's gender. When omitted, the gender defaults to unknown.
<properties>		subelement	<dataset><pedigree source=static or servlet><subject>		The properties assigned to this subject.
<property>		subelements	<dataset><pedigree source=static or servlet><subject><properties>	Any	The value of a property.
<living>		true | false	<dataset><pedigree source=static or servlet><subject>		Defines whether the subject is living.

Table 7-4 <dataset> Elements

Element	Attributes	Values	Parents	Children	Description
<proband>		true | false	<dataset><pedigree source=static or servlet><subject>		Defines whether the subject is the proband.
<twin-group>		String	<dataset><pedigree source=static or servlet><subject>		The name of the twin group to which this subject belongs.
<family-group>		string	<dataset><pedigree source=static or servlet><subject>		The family group that this subject is a member of.
<genotype>		subelements	<dataset><pedigree source=static or servlet><subject>	<allele>	The genotype of the subject, defined as a collection of alleles.
<mates>		string	<dataset><pedigree source=static or servlet><subject>		The subject's mate's ids.
<haplotype-assignment>		subelements	<dataset><pedigree source=static or servlet><subject>	<chromosome>	The haplotype to which each allele should be assigned.
	id	integer	<dataset><pedigree source=static or servlet><subject><haplotype-assignment>		The haplotype id.

Table 7-4 <dataset> Elements

Element	Attributes	Values	Parents	Children	Description
<chromosome>			<dataset><pedigree source=static or servlet><subject><haplotype-assignment>		Defines haplotype assignment for a given chromosome.
	source=csv		<dataset><pedigree source=csv>	<delimiters>, <label-field>, <id-field>, <gender-field>, <living-field>, <proband-field>, <url-field>, <twin-group-field>, <family-group-field>, <parent-fields>, <properties>, <alleles>, <debug>, <elide-indeterminate-haplotypes>, <haplotype>	Defines that the datasource is a csv file.
<delimiters>		string	<dataset><pedigree source=csv>		The characters that separate the data.
<label-field>		integer	<dataset><pedigree source=csv>		The index of the label field (starting at 0).
<id-field>		integer	<dataset><pedigree source=csv>		The index of the id field (starting at 0).
<gender-field>		integer	<dataset><pedigree source=csv>		The index of the gender field (starting at 0).
<living-field>		integer	<dataset><pedigree source=csv>		The index of the living field (starting at 0).
<proband-field>		integer	<dataset><pedigree source=csv>		The index of the proband field (starting at 0).
<url-field>		integer	<dataset><pedigree source=csv>		The index of the url field (starting at 0).
<twin-group-field>		integer	<dataset><pedigree source=csv>		The index of the field that contains the twin group for a subject.
<family-group-field>		integer	<dataset><pedigree source=csv>		The index of the field that contains the family group that this subject is a member of.
<parent-fields>		integer	<dataset><pedigree source=csv>		The index of the parent fields (starting at 0).

Table 7-4 <dataset> Elements

Element	Attributes	Values	Parents	Children	Description
<properties>		subelements	<dataset><pedigree source=csv>	<property>	The property fields.
<property>		subelements	<dataset><pedigree source=csv><properties>	<field>, <name>	The mapping of field index to property name.
<field>		integer	<dataset><pedigree source=csv><properties><property>		The field index to be used to define the value of a property.
<name>		string	<dataset><pedigree source=csv><properties><property>		The name of the property.
<alleles>		subelements	<dataset><pedigree source=csv>	<allele>	The mappings of field index to marker name.
<allele>			<dataset><pedigree source=csv><alleles>	<fields>, <name>	The mapping of field index to marker name.
<fields>		integer	<dataset><pedigree source=csv><alleles><allele>		The field indices that should be used to define the allele values.
<name>		name	<dataset><pedigree source=csv><alleles><allele>		The name of the marker.
<debug>		true | false	<dataset><pedigree source=csv>		Enables debug information to be output to the console.
<elide-indeterminate-haplotypes>		true | false	<dataset><pedigree source=csv>		Defines whether the haplotype designation of alleles whose haplotype is indeterminate should be elided or interpolated from surrounding haplotype assignments.
<haplotype>		List of marker ids	<dataset><pedigree source=csv>		The haplotype.

Table 7-4 <dataset> Elements

Element	Attributes	Values	Parents	Children	Description
	id		<dataset><pedigree source=csv><haplotype>		The haplotype id.
	calculate		<dataset><pedigree source=csv><haplotype>		Defines whether the haplotype should be calculated.
	source=jdbc		<dataset><pedigree source=jdbc>	<url>, <query>, <server>, <driver>, <user>, <password>, <label-field>, <id-field>, <gender-field>, <living-field>, <proband-field>, <url-field>, <twin-group-field>, <family-group-field>, <parent-fields>, <properties>, <alleles>, <debug>, <elide-indeterminate-haplotypes>, <haplotype>	Defines that the datasource is jdbc.
<url>		string	<dataset><pedigree source=jdbc>		Names the datasource file.
<query>		string	<dataset><pedigree source=jdbc>		The SQL query.
<server>		string	<dataset><pedigree source=jdbc>		The URL of the JDBC server.
<driver>		string	<dataset><pedigree source=jdbc>		The name of the JDBC driver class.
<user>		string	<dataset><pedigree source=jdbc>		The user name.
<password>		string	<dataset><pedigree source=jdbc>		The password.
<label-field>		integer	<dataset><pedigree source=jdbc>		The index of the label field (starting at 0).
<id-field>		integer	<dataset><pedigree source=jdbc>		The index of the id field (starting at 0).
<gender-field>		integer	<dataset><pedigree source=jdbc>		The index of the gender field (starting at 0).
<living-field>		integer	<dataset><pedigree source=jdbc>		The index of the living field (starting at 0).

Table 7-4 <dataset> Elements

Element	Attributes	Values	Parents	Children	Description
<proband-field>		integer	<dataset><pedigree source=jdbc>		The index of the proband field (starting at 0).
<url-field>		integer	<dataset><pedigree source=jdbc>		The index of the url field (starting at 0).
<twin-group-field>		integer	<dataset><pedigree source=jdbc>		The index of the field that contains the twin group for a subject.
<family-group-field>		integer	<dataset><pedigree source=jdbc>		The index of the field that contains the family group that this subject is a member of.
<parent-fields>		integer	<dataset><pedigree source=jdbc>		The index of the parent fields (starting at 0).
<properties>		subelements	<dataset><pedigree source=jdbc>	<property>	The property fields.
<property>		Any subelement	<dataset><pedigree source=jdbc><properties>	<field>, <name>, and so on	The mapping of field index to property name.
<field>		integer	<dataset><pedigree source=jdbc><properties><property>		The field index to be used to define the value of a property.
<name>		string	<dataset><pedigree source=jdbc><properties><property>		The name of the property.
<alleles>		subelements	<dataset><pedigree source=jdbc>	<allele>	The mappings of field index to marker name.
<allele>			<dataset><pedigree source=jdbc><alleles>	<fields>, <name>	The mapping of field index to marker name.
<fields>		integer	<dataset><pedigree source=jdbc><alleles><allele>		The field indices that should be used to define the allele values.

Table 7-4 <dataset> Elements

Element	Attributes	Values	Parents	Children	Description
<name>		name	<dataset><pedigree source=jdbc><alleles><allele>		The name of the marker.
<debug>		true | false	<dataset><pedigree source=jdbc>		Enables debug information to be output to the console.
<elide-indeterminate-haplotypes>		true | false	<dataset><pedigree source=jdbc>		Defines whether the haplotype designation of alleles whose haplotype is indeterminate should be elided or interpolated from surrounding haplotype assignments.
<haplotype>		List of marker ids	<dataset><pedigree source=jdbc>		The haplotype.
	id		<dataset><pedigree source=jdbc><haplotype>		The haplotype id.
	calculate		<dataset><pedigree source=jdbc><haplotype>		Defines whether the haplotype should be calculated.

Index

T

U

V

X

Z