

Refiner++

User Manual

## Content

---

1. How This Manual is Structured .....	1
2. Overview .....	2
How Refiner++ works .....	2
Consistency .....	3
Applications .....	5
3. Introduction to the System.....	6
Install and Start Refiner++ .....	6
The Refiner++ Graphical User Interface .....	7
The Refiner++ Data Types .....	8
Load an Existing, Consistent Refiner++ Dataset .....	8
Load an Existing, Inconsistent Refiner++ Dataset.....	8
Use Data from an Existing Database .....	8
4. Basic Use: Refiner++ as Knowledge Acquisition Tool.....	10
Structure the Knowledge .....	10
Add Cases .....	10
Check Consistency of a Dataset.....	10
Remove Inconsistencies.....	11
5. Advanced Use: Analyse Inconsistencies in a Database .....	12
Use and Exclude Refinement Strategies .....	12
Analyse Discriminatory Fields .....	12
Analyse Distribution of Values and Shelve Odd Cases .....	13
Refine Dataset .....	13
Re-Include Shelved Cases .....	13
6. Menu Functions.....	15
File .....	15
Edit.....	16
Data .....	16
View .....	17
7. Windows .....	19
Data Window .....	19
Results Window .....	23
Appendix – Refiner++ XML Schema.....	24

This manual has been written by Peter Troxler, partly based on previous material and articles written by Andy Aiken and Derek Sleeman.

Refiner++ was developed by Andy Aiken with support from Susan Frame and Sam Barker, based on the initial Refiner and Refiner+ algorithm developed by Sunil Sharma and Mark Winter respectively

Aberdeen, September 2004

## How This Manual is Structured

The *Overview* section explains how Refiner++ works. It gives examples of consistent and inconsistent datasets

*Introduction to the System* explains the Refiner++ Graphical User Interface (GUI). It describes how to install and start Refiner++ and how to open or import existing data into Refiner++.

*Basic Use* describes how Refiner++ is used to acquire a knowledge base, how the consistency of a dataset is checked and how inconsistencies are removed.

*Advanced Use* suggests strategies how to analyse inconsistencies and how to remove them.

The previous chapters, *Basic Use* and *Advanced Use*, rely on the last two chapters, the description of the *Menu Functions* and the description of the *Windows*.

The *Appendix* contains the XML Schema that defines the building blocks of a Refiner++ XML file (dataset).

## Overview

The purpose of Refiner++ is to assist an expert to build a consistent knowledge base of classified cases. In a consistent knowledge base the descriptions of the cases given by the expert match the categories inferred by the system.

Producing a knowledge base involves three stages:

1. Knowledge acquisition,
2. Encoding the knowledge, and
3. Debugging the KB.

Refiner++ can be used in all stages. Its main application however is stage 3, debugging.

Working with Refiner++, domain experts have to express their expertise in cases or examples. To describe the cases Refiner++ uses a series of features or descriptors, called fields. Every case has to fall into only one of a series of distinct categories. Refiner++ requires that the categories don't overlap.

So the domain expert has to provide:

- A series of categories,
- A series of fields (called features / descriptors), and
- A set of cases which are described using fields and which fit into categories.

### ***How Refiner++ works***

The domain expert adds a set of cases to a knowledge base. Each case can have an unlimited number of fields for its description. Fields can be numbers, strings (free text), Boolean values (yes/no), Taxons (i.e. a node in a taxonomy; for details see page 16) or Date-Time values. The expert also classifies the cases into categories.

From the case descriptions, Refiner++ automatically infers a description for each of the categories. It checks whether the

dataset is consistent, i.e. whether all cases are assigned by the system to the same categories given by the expert.

Refiner++ provides a number of strategies that have the potential to reduce, and eventually eliminate, the number of inconsistencies in the data set.

The domain expert can then choose one of the strategies. Refiner++ changes the dataset according to the strategy and re-checks the updated dataset. Refiner++ generates a new list of refinement strategies, if the dataset is still inconsistent.

### ***Consistency***

Refiner++ infers the descriptions of the categories automatically from the values the fields given for each case.

#### ***Example 1: This is a consistent dataset.***

Case	Field 1	Field 2	Category
1	A	7	I
2	B	7	II
3	A	5	I
4	B	5	II

#### ***Category Descriptions***

Category I: field 1 = 'A', field 2 = (5...7)

Category II: field 1 = 'B', field 2 = (5...7)

In this case, the discriminatory field is field 1.

#### ***Example 2: This is an inconsistent dataset***

Case	Field 1	Field 2	Category
1	B	7	I
2	B	7	II
3	A	5	I
4	B	5	II

**Category Descriptions**

Category I: field 1 = 'A' or 'B', field 2 = (5...7)

Category II: field 1 = 'B', field 2 = (5...7)

**Inconsistencies**

Case 2 should not match Category I but does,

Case 4 should not match Category I, but does.

Case 1 should not match Category II, but does.

**Example 3: This is a consistent dataset**

Case	Field 1	Field 2	Category
1	A	7	I
2	B	7	I, II
3	A	7	I
4	B	5	II

**Category Descriptions**

Category I: field 1 = 'A' or 'B', field 2 = 7

Category II: field 1 = 'B', field 2 = (5...7)

The discriminatory fields are field 1 and 2.

**Example 4: This is an inconsistent dataset**

Case	Field 1	Field 2	Category
1	A	7	I
2	B	5	I, II
3	A	7	I
4	B	5	II

**Category Descriptions**

Category I: field 1 = 'A' or 'B', field 2 = any value (5...7)

Category II: field 1 = 'B', field 2 = any value (5...7)

***Inconsistency***

Case 4 should not match category I, but does.

***Applications***

Refiner++ has been used in a variety of domains, including:

- Pain control
- Child psychology
- Intensive care
- Asthma

## Introduction to the System

### *Install and Start Refiner++*

You can download Refiner++ from the Refiner++ website at <http://www.csd.abdn.ac.uk/~aaiken/refiner/> or from the AKT technologies page at <http://www.aktors.org/technologies/refinerplusplus/>

Refiner++ is written in Java so should run on all platforms. To be able to run Refiner++, Java must be installed on the computer. Java is usually pre-installed on Macintosh and Unix platforms. If Java is not currently installed, it can be downloaded from the Sun Website.

To run Refiner++ on a UNIX or Windows computer, open up a terminal window or DOS prompt, and navigate to the directory that contains `refiner.jar`. Type the following command:

```
java -jar refiner.jar
```

To run Refiner++ on a Macintosh computer (OS 10 or higher) double click on its icon.

Please be patient while Refiner++ is starting up; the graphics libraries that Refiner++ uses may take a few minutes to load.

### ***The Refiner++ Graphical User Interface***



- [1] Menu
- [2] Buttons
- [3] Data part of the window with three tabs:
  - Fields (the field tab is active in the picture)
  - Cases
  - Categories
- [4] Results part of the window with two tabs:
  - Strategies (the strategies tab is active in the picture)
  - Inconsistencies

## ***The Refiner++ Data Types***

Refiner++ uses 5 data types:

- Numeric
- String
- Boolean
- Taxon (for details see Background Knowledge (Taxonomies) on page 16)
- DateTime

### ***Load an Existing, Consistent Refiner++ Dataset***

Import an existing, consistent Refiner++ dataset using the menu *File > Open...*

Check in the fields tab that all fields have been imported and recognized properly. Check in the cases tab that all cases have been imported properly. Check in the categories tab that Refiner++ has detected all the categories.

Validate your dataset (Ctrl-V, *Data > Validate*, or [>]-button). Refiner++ should not report any inconsistencies.

### ***Load an Existing, Inconsistent Refiner++ Dataset***

Instructions as above; in this case Refiner++ should now report inconsistencies.

## ***Use Data from an Existing Database***

### ***Prepare Database File for Import***

To prepare an existing database for Refiner++, transform your data in standard CSV (comma separated values) format. Make sure your fields are enclosed with "", particularly if they contain commas.

Your CSV file can contain any number of fields; Refiner++ will automatically detect the type of the data. Refiner++ takes each column in turn and tries to convert each cell into each data type it knows. If it's able to convert more than 95% of non-empty fields into a particular type, it assumes that's the type.

Refiner++ interprets the last field of every record in your CSV file as the category, so you will have to reorder the fields in your dataset if it is not in that format. If your dataset does not

contain categories, append an empty field at the end of each record in the dataset. You will have to add the categories later.

### ***Import CSV file***

Import the CSV file using the Import > from CSV option from the menu File.

Check in the fields tab that all fields have been imported and recognized properly. Check in the cases tab that all cases have been imported properly. Check in the categories tab that Refiner++ has detected all the categories.

## **Basic Use: Refiner++ as Knowledge Acquisition Tool**

### ***Structure the Knowledge***

Develop an initial description of the structure of your raw data. This is an offline task:

- What categories of cases / experiences do you expect?  
Add these in the categories tab.
- What are the characteristics (features, descriptors) of the cases / experiences?  
Add these in the fields tab.
- What are unique identifiers for your cases / examples (record numbers, names etc.)  
Add these in the fields tab, but remember to un-tick the box 'in use'.

*Note: Refiner is likely to pick up record numbers or other features that are unique per case as discriminatory fields and give you the impression of a consistent database.*

### ***Add Cases***

Go to the cases tab and start to enter your first few cases.

### ***Check Consistency of a Dataset***

After adding each case, validate your dataset (Ctrl-V, *Data > Validate*, or [*>*]-button). Refiner++ reports whether it has found any inconsistencies in the dataset. Initially you would not expect any inconsistencies, and Refiner++ will report this in a dialog box saying 'The database contains no inconsistencies'.

### ***Inconsistencies***

After a while your dataset might become inconsistent. Refiner++ will report a summary 'The database contains *N* inconsistencies; *M* strategies have been suggested.' The inconsistencies tab in the results window contains the inconsistencies that Refiner++ has detected.

## ***Strategies***

The strategies tab contains Refiner++'s suggestions of how to make the case base consistent. The 'configure allowed strategies' option from the data menu allows the user to control which strategies Refiner can suggest.

The list of suggested strategies is sorted based on three heuristics:

1. 'Remove Value' strategies are always first, because they always discriminate between two categories.
2. Strategies which are suggested multiple times are preferred over strategies which are suggested fewer times.
3. Strategies which affect fewer cases are preferred over strategies which affect more cases.

A strategy can be selected by double clicking it. If Refiner++ has suggested changing values, a dialog box will pop up and ask the user to select one. After executing a strategy, Refiner++ suggests that the user re-validates the dataset.

## ***Remove Inconsistencies***

Go to the strategies tab and see what strategies Refiner++ suggests. Select a strategy to make your dataset consistent again.

*Note: If you feel some of the strategies are generally not suitable for your dataset (e.g. 'Allow additional category') you can tell Refiner++ not to suggest these: Go to the menu Data > Configure allowed strategies and un-tick the corresponding boxes.*

## ***Undo / Redo***

*Refiner ++ has an undo / redo facility which allows you to restore the state of the knowledge base after a strategy has been performed. These faculties allow users to perform WHAT-IF experiments on the case base.*

## **Advanced Use: Analyse Inconsistencies in a Previously Defined Dataset**

Once you've acquired a knowledge base or imported a CSV file you are ready to analyse your data. The following tips might be helpful in producing a good result.

### ***Use and Exclude Refinement Strategies***

Define which strategies you don't want to use from the menu *Data > Configure* allowed strategies by un-ticking the corresponding boxes. You might want to exclude e.g. the 'change value strategy' if the values in the fields of your dataset have been validated.

### ***Analyse Discriminatory Fields***

Experiment with including or excluding fields in the analysis. Verify whether the fields you select are actually useful as discriminatory fields for Refiner++; while potentially discriminatory a field could contain no values that help Refiner++ discriminate between categories.

### ***Inferred by Refiner++***

Determine the fields Refiner++ identifies as discriminatory:

1. Run a first validation (Ctrl-V, *Data > Validate*, or [>]-button).
2. Select the fields tab and select from the menu *Fields > Discriminatory Fields > Show only Discriminatory Fields*
3. Compare the list of discriminatory fields with the expectations from the domain expert.

### ***Suggested by a Domain Expert***

Re-validate your dataset with only those fields active that the domain expert would expect to be discriminatory:

1. Select the fields tab and select from the menu *Fields > Discriminatory Fields > Show only Discriminatory Fields;*

un-tick the 'in use' box for all fields that should not be discriminatory.

2. Select from the menu *Fields > Discriminatory Fields > Show only Discriminatory Fields*; tick the 'in use' box for all fields that should be discriminatory.

*Note: Refiner is likely to pick up record numbers or other features that are unique per case as discriminatory fields and give you the impression of a consistent database (so you will need to shelve this field).*

3. Run a validation (Ctrl-V, *Data > Validate*, or [=]-button).

### ***Analyse Distribution of Values and Shelve Odd Cases***

Select a strategy to improve your raw dataset and validate repeatedly until the number of inconsistencies and/or strategies is fairly low:

1. Review your cases and look for stray values. In the fields tab click on any column heading to cycle through ascending, descending, and initial (import) sort order. Shelve cases with stray values by un-ticking the box 'in use'.
2. Analyse the distribution of values per categories and field. The 'Distribution Graphs' are a perfect tool to do this. To view a distribution graph, select the categories tab, right-click (or control-click) on a field for one category and select *Distribution graph > For this category* from the pop-up menu. Identify strange values, go back to the cases tab and shelve the corresponding cases by un-ticking the box 'in use'.

### ***Refine Dataset***

Once you are happy with the number of inconsistencies and/or strategies you can start to refine your dataset using the strategies Refiner++ offers. You might want to shelve cases that make the dataset inconsistent.

### ***Re-Include Shelved Cases***

Once your dataset is consistent, you can start to work on the shelved cases and add them one by one back into the dataset,

using Refiner++'s strategy suggestions to keep the dataset consistent at all times.

## Menu Functions

### *File*

#### ***New (Ctrl-N)***

Create a new case base

#### ***Open (Ctrl-O)***

Open a saved case base. The file has to be in XML format, as produced by Refiner++. For the specification of the structure (schema) see appendix.

### ***Import***

#### ***from CSV***

Import from CSV files

#### ***from data set***

Import from xml data set

### ***Export***

#### ***to CSV***

Export to CSV

#### ***Save (Ctrl-S)***

Save the current file

#### ***Save As...***

Save the current file under a new name and/or in a new location

### ***Exit***

Exit

## **Edit**

### **Undo Strategy (Ctrl-Z)**

Undo a previously run strategy, unavailable after editing data.

### **Redo Strategy (Ctrl-Y)**

Redo a previously undone strategy, unavailable after editing data.

## **Data**

### **Background Knowledge (Taxonomies)**

Manipulate background knowledge (Taxonomies)

This is a very basic tool to build simple taxonomic descriptions (features) of a case. The command opens a separate taxonomies window.

The Taxonomies window lets you add, remove or edit Taxonomies. The buttons OK and cancel leave the taxonomies window.

*Note: The add button only adds the top-level node of the taxonomy.*

To add child nodes select the top-level node and click edit. This opens a new window to edit the taxonomy. In the new window mark a node and

- click add to add a child node,
- click remove to remove the node,
- click OK to save changes,
- click cancel to exit without saving.

### **New Case**

Creates a new Case

### **New Category**

Creates a new Category

### ***New Field***

Creates a new Field

### ***Reset descriptions***

Regenerates category descriptions

### ***Validate (Ctrl-V)***

Validates the case base

### ***Configure allowed strategies***

Configure allowed strategies, ability to enable and disable certain strategy types:

- Add new field strategy
- Additional category strategy
- Change value strategy
- Exclude value strategy
- Reclassify case strategy
- Remove category strategy
- Shelve case strategy

### ***View***

The view menu controls which fields and which cases are displayed in the table. Fields can be discriminatory or non-discriminatory:

- Discriminatory: field is used to characterise categories
- Non-discriminatory: field is not used to characterise categories.

Fields and cases can be active or shelved:

- Active: field or case is taken into consideration when validating the dataset

- Shelved: field or case is not taken into consideration when validating the dataset

## ***Fields***

### ***Discriminatory Fields***

Show all: show all fields whether they are discriminatory or not

Only show discriminatory fields: show only fields that are discriminatory

Only show non-discriminatory fields: show only fields that are not discriminatory

### ***Shelved Fields***

Show all: show all fields whether they are active or shelved

Only show active fields: show only fields that are active i.e. used in validation

Only show shelved fields: show only fields that are shelved i.e. not used in validation

## ***Cases***

Show all: show all cases regardless of whether they are active or shelved

Only show active cases: show only cases that are active

Only show shelved cases: show only cases that are shelved

## Windows

### *Data Window*

#### *Fields Tab*

User can edit the details of a field by double clicking the appropriate row.

The resulting dialog box allows the changing of a field's name and/or type as well as the fields shelved status. For Numeric fields the user is provided with the ability to specify the number of decimal places and/or significant figures to use when rendering the values of that field. Or for taxonomies the user can select which taxonomy to use for that field.

Checking or unchecking the box in the 'in use' column of the table also changes a field's shelved status.

Fields can be renamed by right clicking on the field's table and selecting rename. Fields can also be removed by using the right click menu. Additional fields can also be added by using the right click menu.

Shelved fields are marked in light gray text.

#### *Cases Tab*

The Cases Tab allows "in place" editing of values.

The user has the capacity to edit all values (apart from the case number) in this table.

They can alter a case's shelved status again by checking or unchecking the box in the 'In Use' column or by selecting 'Shelve Case' from the right click menu. Shelved cases and fields are identified by light gray text.

To edit a value the user must double click on the appropriate cell. Numeric values are presented in their raw form for editing. A case's values can also be edited by selecting edit value from the right click menu. Values can be removed by selecting 'Remove value' from the right click menu, or in the case of dates by selecting an empty month.

The cases table also provides the ability to edit, rename, and remove fields using the right click menu.

The cases table allows the sorting of values within it. Sorting instructions can be placed on any column within the table. Sorting direction can be set by clicking the header of the column reader of the column to be sorted. Multiple sorts can be performed by holding down the control key when clicking a column header. Clicking a sorted column header will change the sorting direction. There are three options available, Ascending, Descending and default (default sort order is the order from the case base which is in order of case id).

Boolean values sort in the order 'Any', 'True', 'False'.

Taxonomy values sort in their hierarchical order. The children of each node sort in alphabetical order.

### ***Categories Tab***

From the right click menu on the category table the user has access to four functions:

#### ***Rename Category***

Rename the category; you can also do this by double-clicking on the category row.

#### ***Remove Category***

Deletes the category from the dataset. WARNING: this cannot be undone.

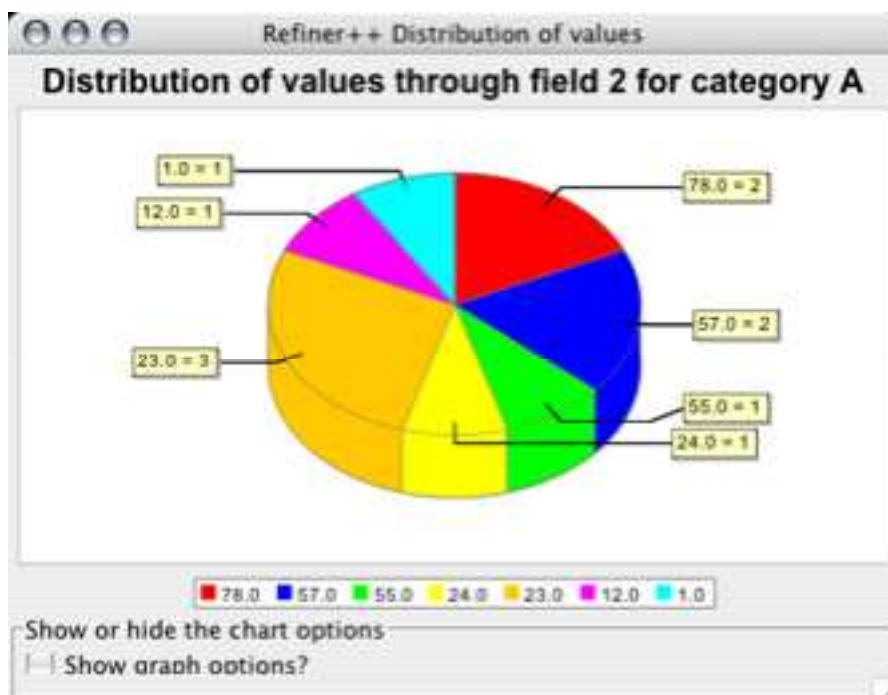
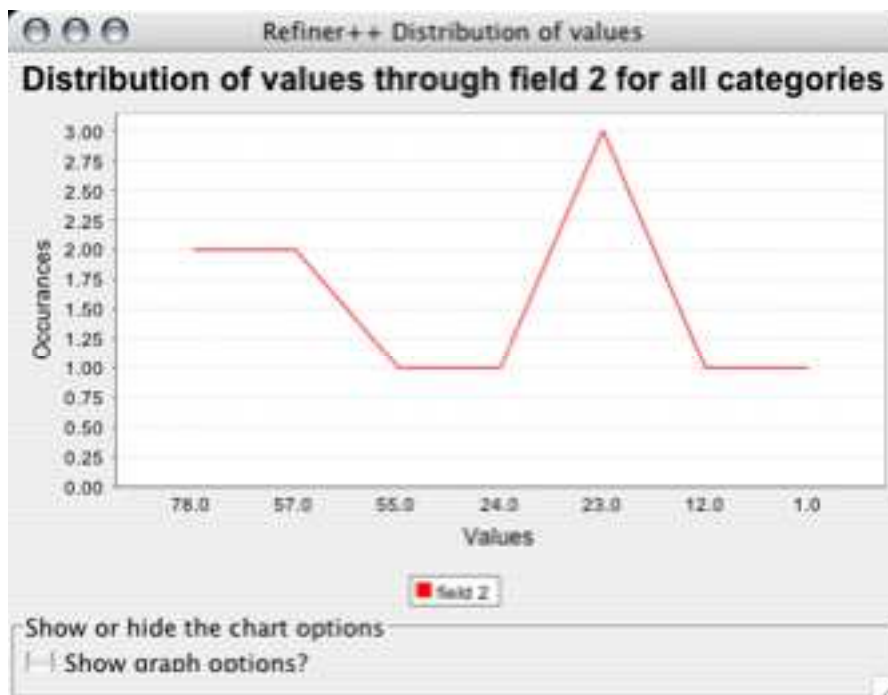
#### ***Reset Descriptions***

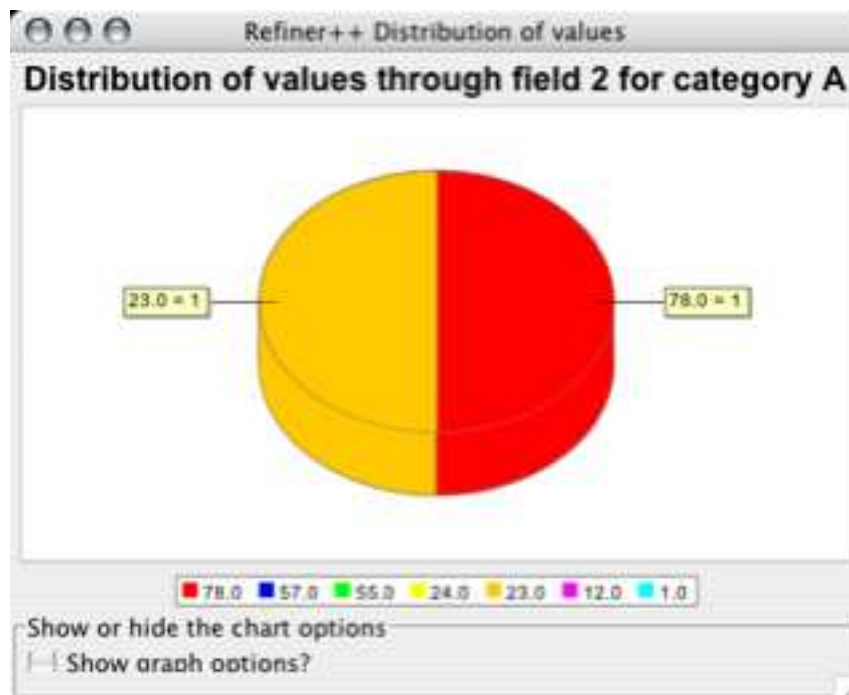
The descriptions for all categories are regenerated. This command is useful if no extra validation has been carried out after shelving cases or changing values.

#### ***Distribution Graphs (for all categories / for this category)***

Refiner++ provides graphs from two sources:

1. Distribution across all categories
2. Distribution for a particular category.





The distribution graph has several control options which allow the user to control the graph drawn:

The screenshot shows the "Show or hide the chart options" control panel. It contains several sections:
 

- Show or hide the chart options:** A checkbox for "Show graph options?" is checked.
- Select a chart type:** Two radio buttons are present: "Distribution chart" (selected) and "Pie Chart".
- Select a category to graph:** Two radio buttons are present: "Just graph 'field 1' for category: A" and "Graph 'field 1' for all categories" (selected).
- Select a sorting direction:** Two radio buttons are present: "Assending order" and "Desending order" (selected).
- Should <no data> enteries be counted:** A checkbox for "Count no data entries?" is checked.

The user can switch between a line chart and a pie chart of the data.

The user can also switch between displaying the distribution for one category (the one that was initially selected) and for all categories.

The user can select the sort order of values.

The distribution graph can also include '<no data>' entries when graphing all categories.

## ***Results Window***

### ***Inconsistencies Tab***

This tab lists all the inconsistencies that Refiner++ detected in the last validation run.

### ***Strategies Tab***

This tab lists all the strategies that Refiner++ suggests to reduce inconsistencies.

Double click on a strategy to apply it to the dataset. If Refiner++ suggests multiple values in the Replace Value strategy, a dialogue box will ask the user to select one particular value that will be used for all cases.

## **Appendix – Refiner++ XML Schema**

{Yet to be produced by Andy}