

BIRT 2.0 Chart User Interface Functional Specification

New Chart UI Specification

This document specifies the functionality of the new user interface being provided for building charts within BIRT v 2.0.

Document History

Version	Changes	Date
Draft 3	Moved the Property Editor and Popup Menu sections into a separate document Changes made based on feedback	09/06/2005
Draft 2	Added clarification regarding type selection UI behavior. Updated Screenshots Modified / Updated descriptions of Type and Data sheets	08/30/2005
Draft 1	Initial Draft	

Table of Contents

Introduction	1
<i>References:</i>	<i>1</i>
New User Interface Components	2
<i>Chart Builder v 2.0</i>	<i>2</i>
Open Issues	34

Introduction

The BIRT Chart library provides the charting functionality for the BIRT Report Designer. The primary interface between the user and the chart engine is the 'Chart Builder'. The chart builder is a user interface that allows visual creation and editing of charts embedded into the report design. The primary purpose of the new user interface to be introduced in v 2.0 of BIRT is to make chart creation and editing an easy to follow, task-oriented, process. To enable this, the new UI will be very user friendly and will be organized to only show users properties and options that are relevant to the task at hand. This document is intended to provide a description of how this will be look and how it will work.

References:

Chart FAQ:

<http://www.eclipse.org/birt/wiki/index.php?n=FAQ.Charts>

Chart UI Feature Spec.

<http://www.eclipse.org/birt/release20specs/BPS15%20-%20New%20Chart%20User%20Interface%20Feature%20SPEC%201.1.pdf>

Chart UI Design Mockups.

<Include reference to UI Design Document.>

Property Editor and Popup Menu Functional Spec.

<Include reference to Property Editor and Popup Menu Functional Spec.>

New User Interface Components

For BIRT v 2.0, significant changes will be made to the way users create and modify charts in reports. The more visible of these changes will be in the form of a new User Interface for the chart builder as well as additional UI components and options for accessing the various task related UI for charts within the report.

The UI changes can be considered as three distinct but related components:

- a. [Chart Builder v 2.0](#)
- b. [Chart-specific Property Editor Sheets](#)
- c. [Popup Menu Items for Charts](#)

Each of these will provide access to specific tasks in the chart creation and editing process and will make it easier for users to get tasks done without wading through unnecessary steps.

The following sections will give detailed descriptions on each of the new UI components. (NOTE: The sections related to Property Editor Sheets and Popup Menu have been moved into a separate document.)

Chart Builder v 2.0

Task-based Wizard Framework

The new Chart Builder User Interface will be built up on a framework that will provide infrastructure support for the specific UI elements. This framework will be generic and will be built with a view to creating different task-oriented wizards. As such, the framework will provide a mechanism to cleanly and simply add tasks as well as customize the list of tasks shown for any particular invocation.

The general appearance of a task-oriented wizard will be as follows:

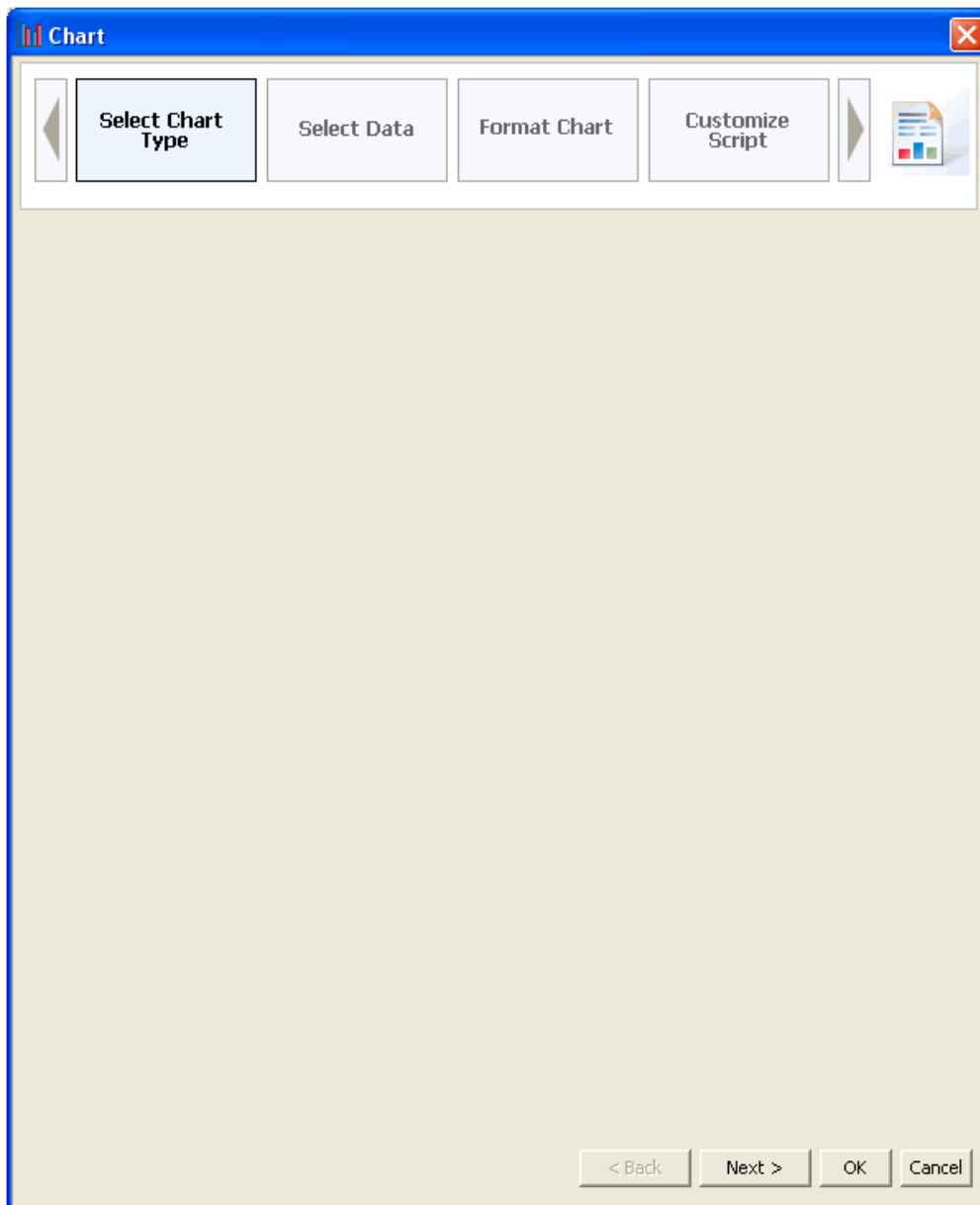


Figure 1. BIRT 2.0: Task-based Wizard Framework

The wizard will consist of a series of buttons along the top that will represent all available tasks currently accessible. It will be possible to enable/disable tasks by specifying an identifying label. If a task with that label is registered and available in the current invocation, it will be enabled / disabled by a call to the framework. Tasks in each of these states will be distinguished by the appearance of their entries in the task list. To allow navigation between tasks when more tasks are available than can fit in the UI, the task list will feature two 'scroll buttons' that will allow users to scroll along the list to see additional available tasks.

In addition to the task list along the top, the framework will provide a set of buttons along the bottom (Previous, Next, Ok and Cancel). The enabling of the Previous & Next buttons will be handled by the Framework. The 'OK' button will be labeled as 'Finish' for all but the last task. It will be possible to enable / disable the 'Finish' button through a call to the Wizard Framework.

To give an additional layer of usability, the Framework will provide default implementation of two classes that represent common task types; Simple Task (where all task activities occur on a single sheet) and Compound Task (where a task may involve changing loosely related properties spread over multiple sheets). The Simple Task UI container will be a thin wrapper that will allow setting of a single composite that occupies its entire area. The

Compound Task UI container will include a navigation mechanism (e.g. a tree) along the left and a container area along the right. This container will allow addition of a group of sub-tasks and sheets associated with each through registration API provided by the UI container.

The tasks available in the Chart Wizard will be as follows (**NOTE:** The screenshots for all sheets have not been updated to reflect the new functionality provided by the framework.):

Task – Chart Type Selection

The first task in the creation of a chart is the selection of the chart type. This selection in many ways defines what the chart looks like and will be driven to a great extent by the type of data to be represented.

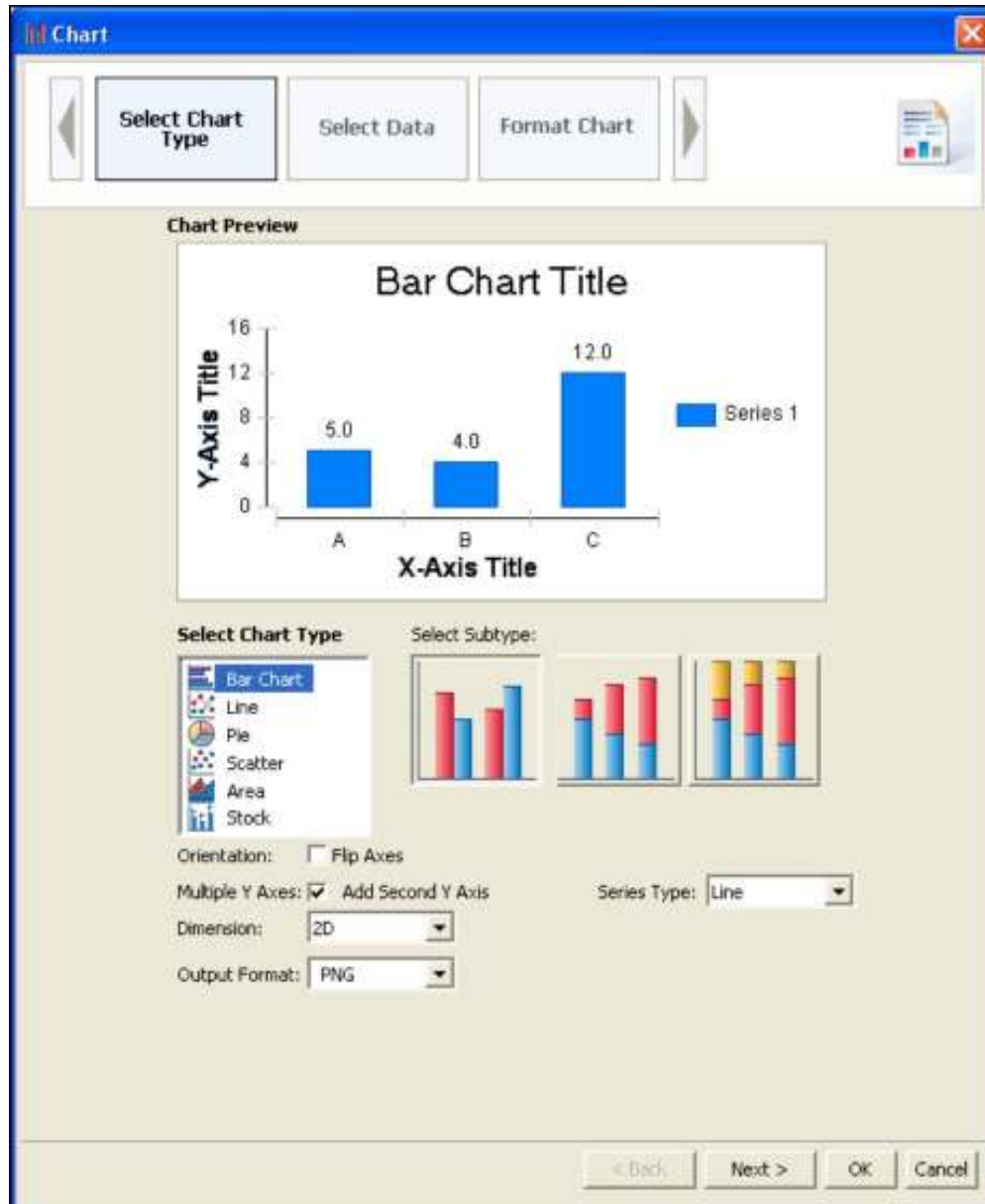


Figure 2. BIRT 2.0: UI for Type Selection Task

The sheet will contain the following UI elements:

Chart 'Live Preview' – This component will display the current chart. Changing the chart type or any other property in the wizard will cause the chart to be updated to reflect what the final chart would look like.

Chart type selection list – The chart type selection list will display all available chart types in the order of popularity. The most commonly used chart types will show up first. This list will be hard coded and will not change. Custom chart types (if implemented) will show up at the bottom of the list.

Chart subtype buttons – For each chart type, there will be one or more subtypes. The user will be able to select the subtypes using an array of toggle buttons which will display icons representative of the charts. The button list will change depending on the dimensions and the type. (Some chart types will only support some subtypes when rendered in 2D for example).

Orientation option – The chart orientation will be configurable through a checkbox. The checkbox will be enabled or disabled depending on whether a chart type supports transposition. Checking or un-checking this option will cause the button list to be updated to reflect the appearance of the chart.

Multiple Y Axes – Some chart types will support overlay axes (in the form of a second Y axis). This option will be enabled for charts with axes only. Selecting or unselecting this option will cause the button list to be changed to reflect the appearance of the chart.

Overlay series type – This drop-down will allow users to specify the series type to be used to display overlay data. This option will only be enabled if the Multiple Y Axes checkbox is selected. The default new series will be the same type as the standard series for this chart type.

Chart Dimension – Most chart types will support rendering in multiple forms (as 2D charts, 2D with depth or 3D). The chart dimension drop-down will allow users to change the dimensions based on a list of supported dimensions for each chart type. Changing this value will cause the chart subtype button list to be updated to reflect the appearance of the charts.

Output Format – This option will set the default output format for the chart in the generated report. This will only be a hint. The actual output format will depend on the report output format.

Restrictions

The series types available for the Overlay series will only be those that are support combination. I.e. it will not be possible to add a Pie Series as the overlay series for a Bar Chart.

Task – Chart Data Binding

The next and perhaps the most important task in creating a chart is the data binding. This is where the source of the chart data as well as the links between the various series' and individual elements (expressions) from the dataset are defined. The data binding sheet will allow users to see the live data in tabular form as well as associate columns or expressions with each series in the chart.

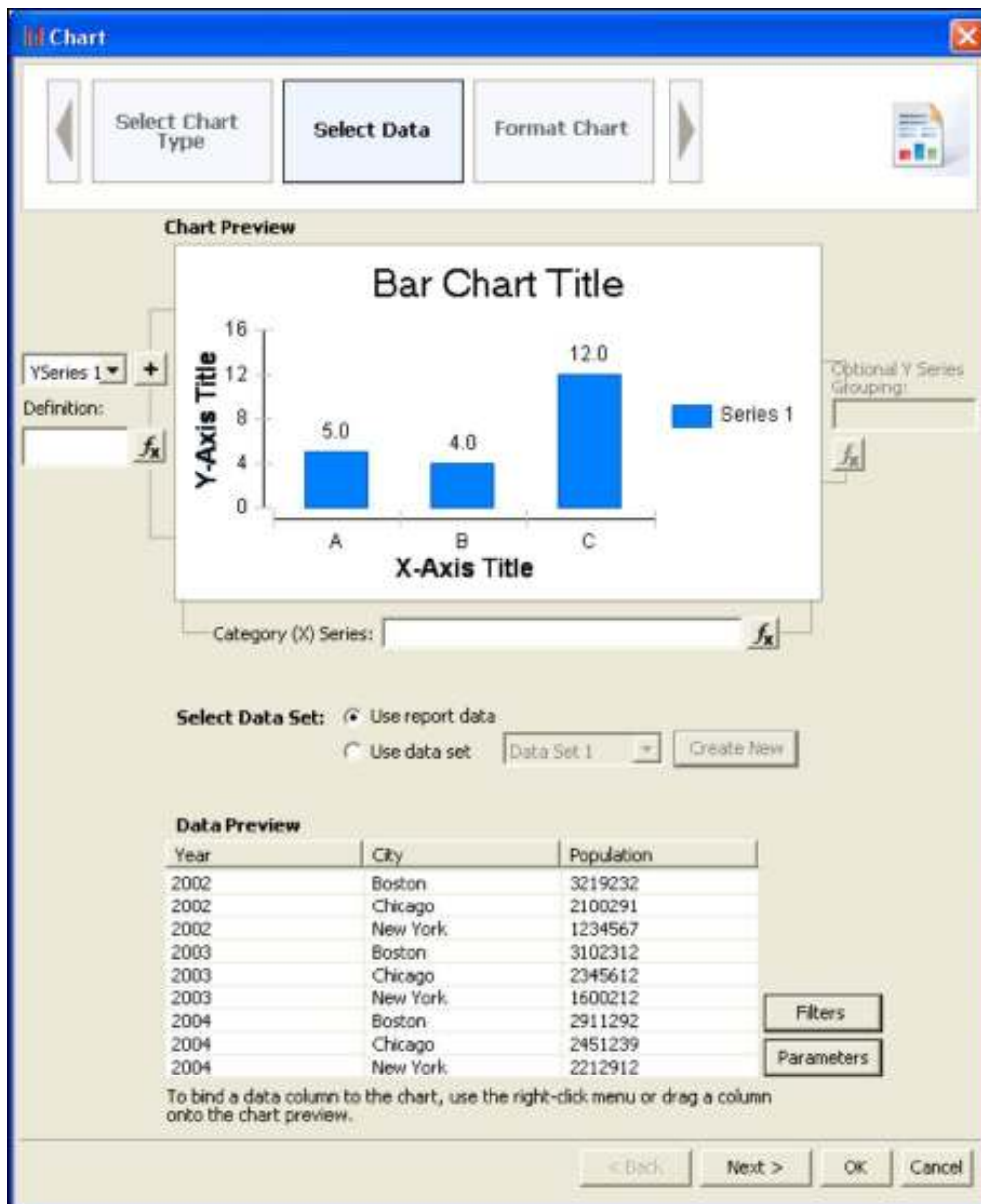


Figure 3. BIRT 2.0: New Data Binding Sheet

Dataset Binding – The first step in binding the data with the chart involves specifying the source of the chart data. In BIRT, this would be either the report data or a dataset.

Choosing 'Use report data' would cause the chart to inherit the data binding from its container in the report design. This option would only be available if the chart's container in the report structure has some inheritable data.

Choosing 'Use data set' would allow the users to specify the dataset they want with the chart. They could at this point either select from the available datasets already defined in the report or select 'None' in the drop down and click on the 'Create New' button to create a new dataset and bind it with the chart.

Dataset view – Choosing an existing dataset would cause the current data for the dataset being displayed in a table in the data binding sheet. It will be possible for users to drag column headings from this table to one of the series data fields to bind the column with the series. Doing so will cause the heading of the table to reflect that it is bound to a series.

Filter data – A button next to the data table will invoke a separate UI to specify filters to be applied at the chart level. Clicking on this button will invoke the dataset filter UI.

Data Parameters – Another button will be provided to set the values for dataset parameters. This button will be disabled if the selected dataset does not have any parameters defined.

X / Base Series data – A text field will allow specification of an expression for the base (X) series.

Y / Orthogonal Series data – A pair of drop-down lists will show the available Y and Overlay series. A button next to the drop-down will alternately add or remove a series. The UI for data definition of the Y and Overlay series will be obtained from the appropriate registered UI providers for this series type (SeriesUIComposite extension point implementations).

Overlay Series data – A list will be provided to specify up to two (2) expressions, one each for each overlay series.

Series Grouping key – A set of text fields will be provided to specify the (optional) grouping key for the Y series and the Overlay series. The text fields for Overlay series will only be available if multiple Y axes have been enabled.

Each of the above components will be accompanied by a button that will invoke the BIRT Expression Builder UI. The user will be able to use the expression builder to select a data field or enter an expression to be associated with the series.

NOTE: The actual UI (in terms of layout and options available) will be different depending on the currently selected chart type.

Restrictions

- A maximum of two (2) Y series and two (2) Overlay series can be defined in the data sheet. These numbers cannot be transferred from one type to the other (3 Y series and 1 Overlay series are not permitted).
- Only one (1) X series can be defined in the data sheet.

Task – Chart Formatting

The formatting task is probably the most time consuming task of the chart creation process. It can also be the simplest. This task is expected to be an optional task where the user would leave the default formatting for basic charts but spend a long time tweaking the appearance of the more important charts.

The UI for this task will be made up of a navigation tree along the left that will allow access to any of the main chart components and a content area on the right that will show the basic (and most often modified) properties for a selected node from the tree and the preview of the chart.

The more advanced of the properties for the selected node will be accessible through a 'More Options...' toggle button that will popup a dialog docked to the right side of the main wizard. The popup dialog will hold additional properties for the selected node and will contain 'Apply', 'Ok' and 'Cancel' buttons.

Clicking on 'Apply' in the popup dialog will cause the changes made in the dialog to be applied to the chart. Clicking on 'Ok' will have the same result as the 'Apply' button but will in addition close the popup dialog. Clicking on 'Cancel' will cause all the changes made since the popup dialog was opened, or 'Apply' was clicked on, to be lost and the dialog will be closed. Clicking on the 'More Options...' button or changing the tree selection when the popup is shown will have the same effect as clicking on Cancel in the popup dialog.

The following elements will be available as nodes in the navigation tree for the formatting task:

Chart

This node will represent the chart object as a whole and will allow formatting of general properties related to the chart.

Basic Properties

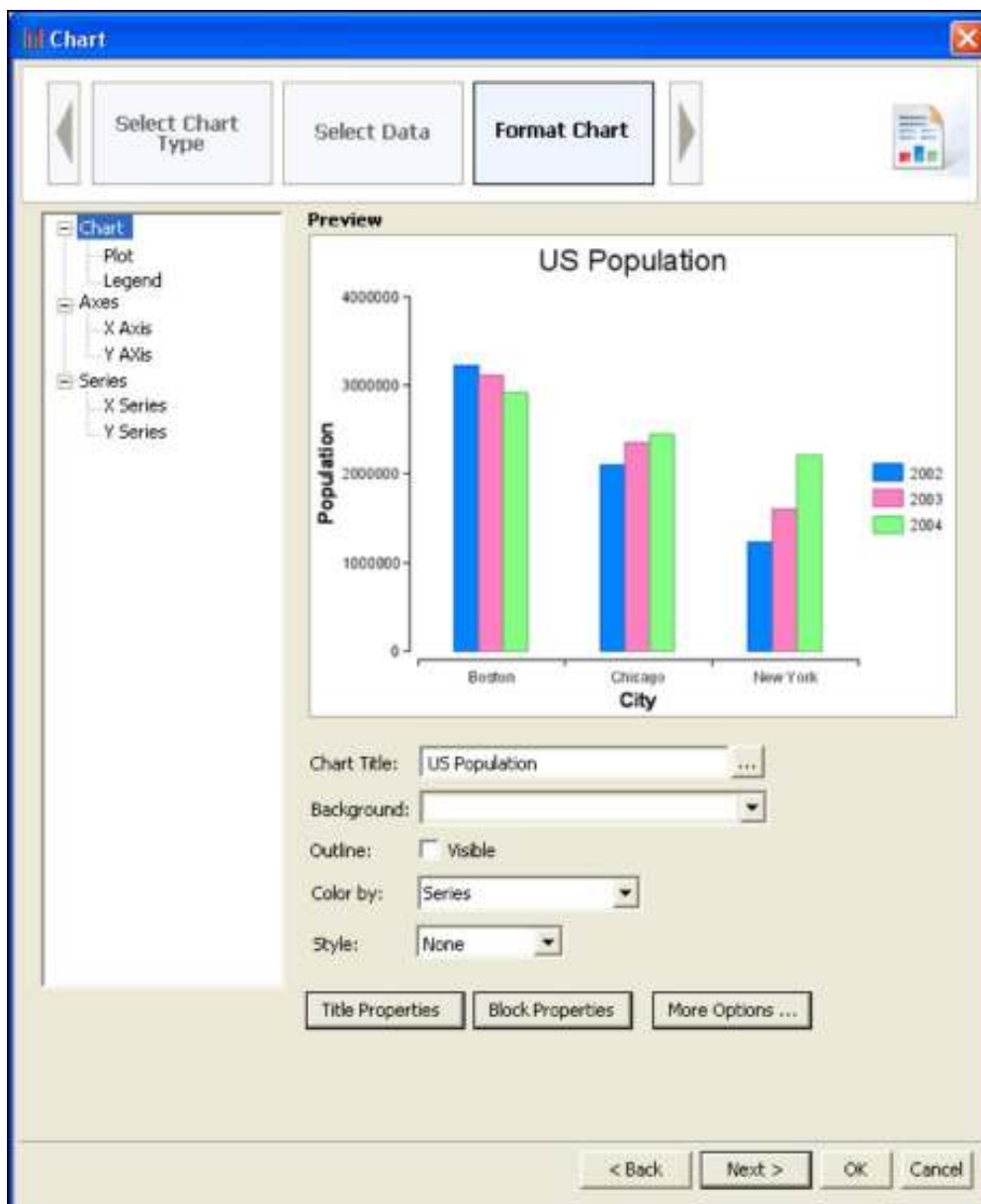


Figure 4. BIRT 2.0: Basic Chart Properties Sheet

Advanced Properties

Figure 5. BIRT 2.0: Advanced Chart Properties Popup

Misc. Popups

Title Properties

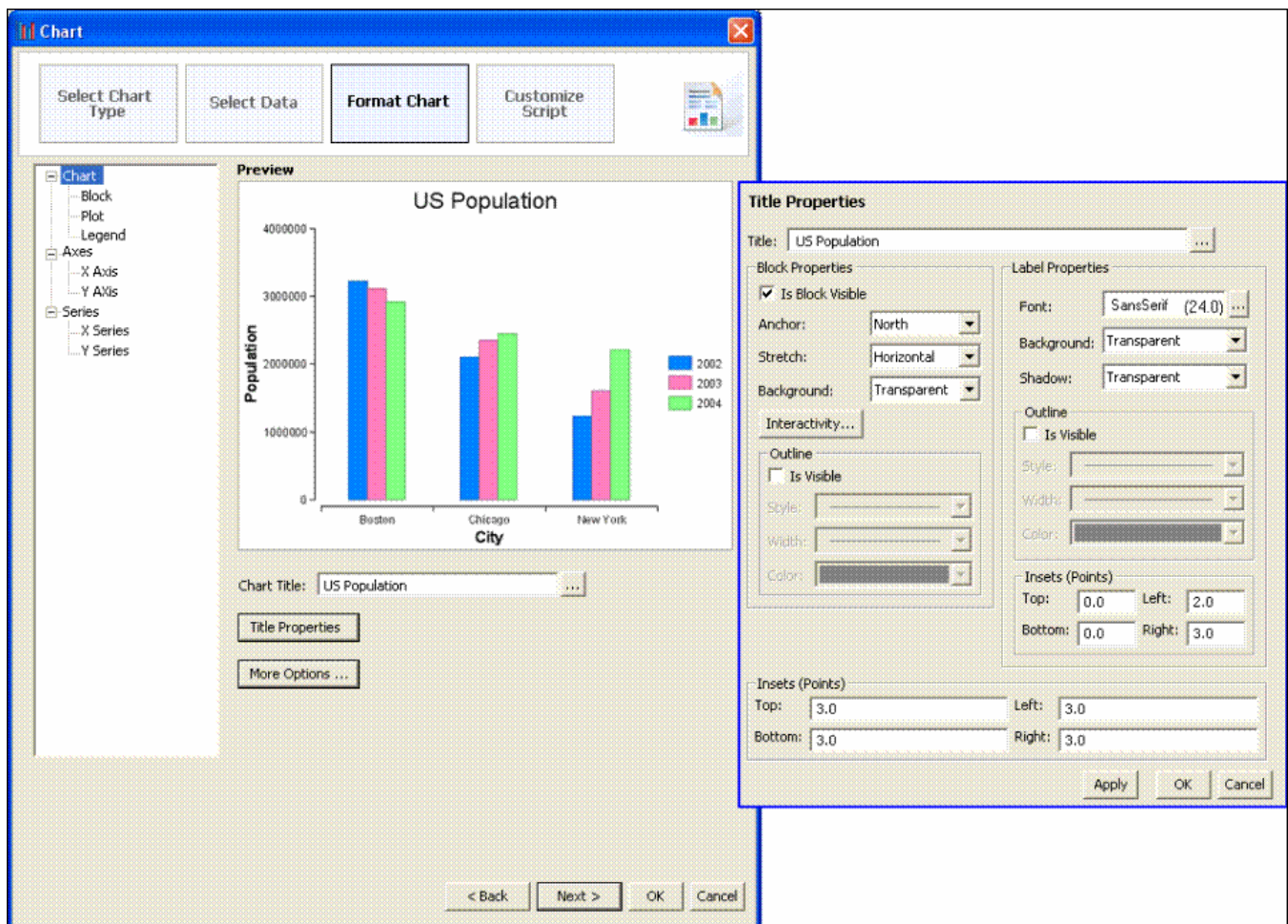


Figure 6. BIRT 2.0: Chart Title Properties Popup

Block Properties

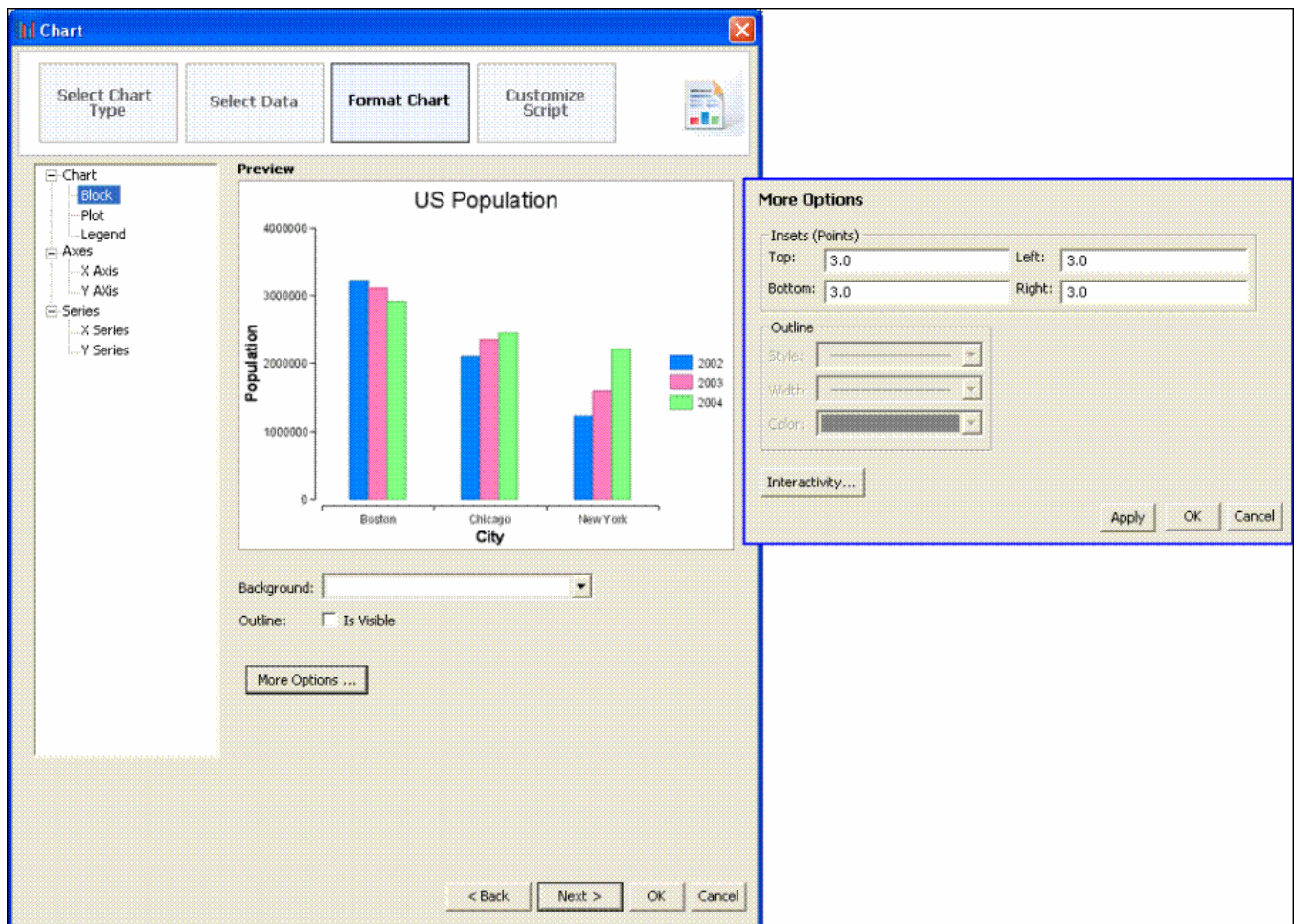


Figure 7. BIRT 2.0: Chart Block Properties Popup

Chart – Plot

This node will represent the plot area of the chart object and will allow formatting of properties related to the plot and its client area.

Basic Properties



Figure 8. BIRT 2.0: Chart Plot Basic Properties Sheet

Advanced Properties

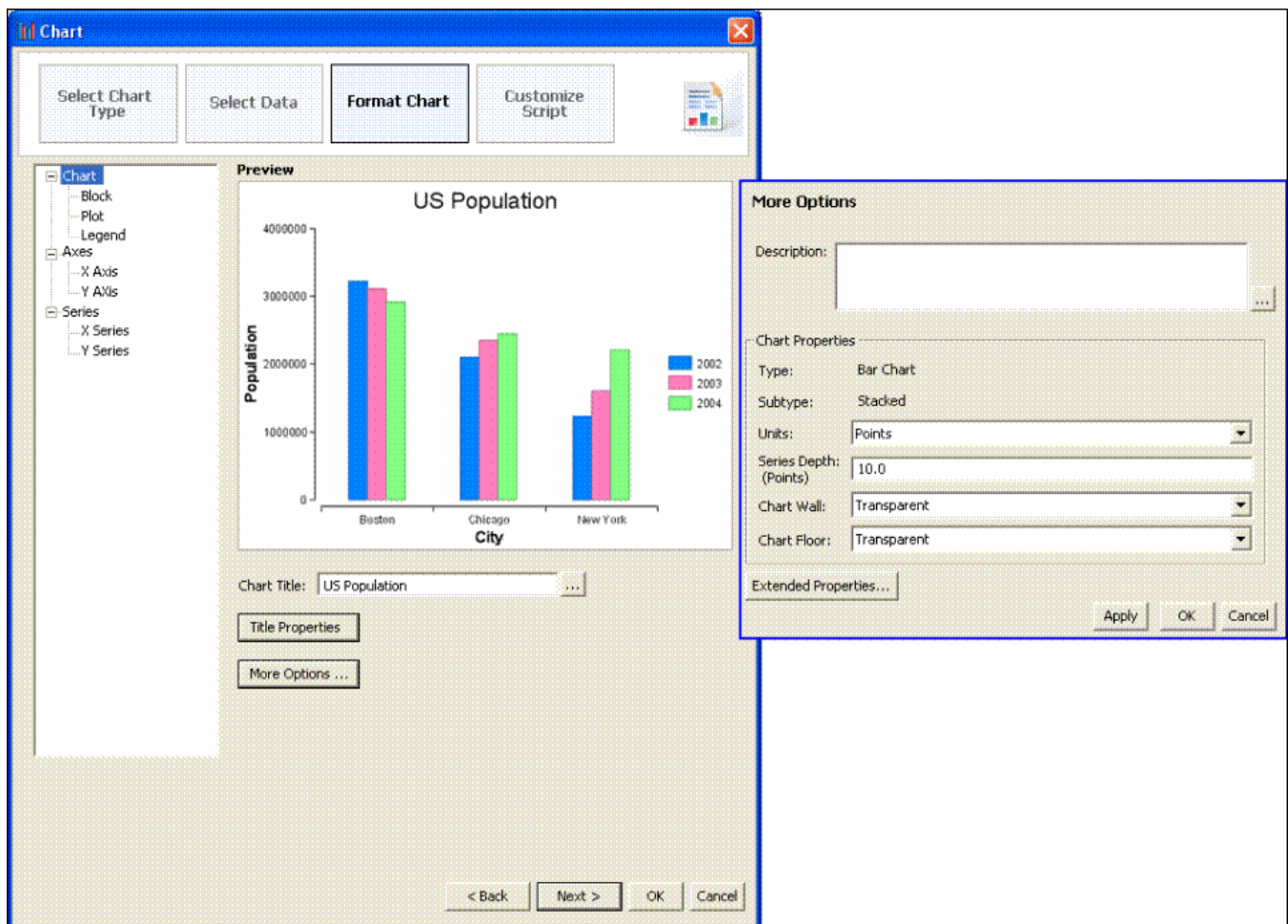


Figure 9. BIRT 2.0: Chart Plot Advanced Popup

Chart – Legend

This node will represent the legend area and will allow formatting of properties related to the legend and its client area.

Basic Properties

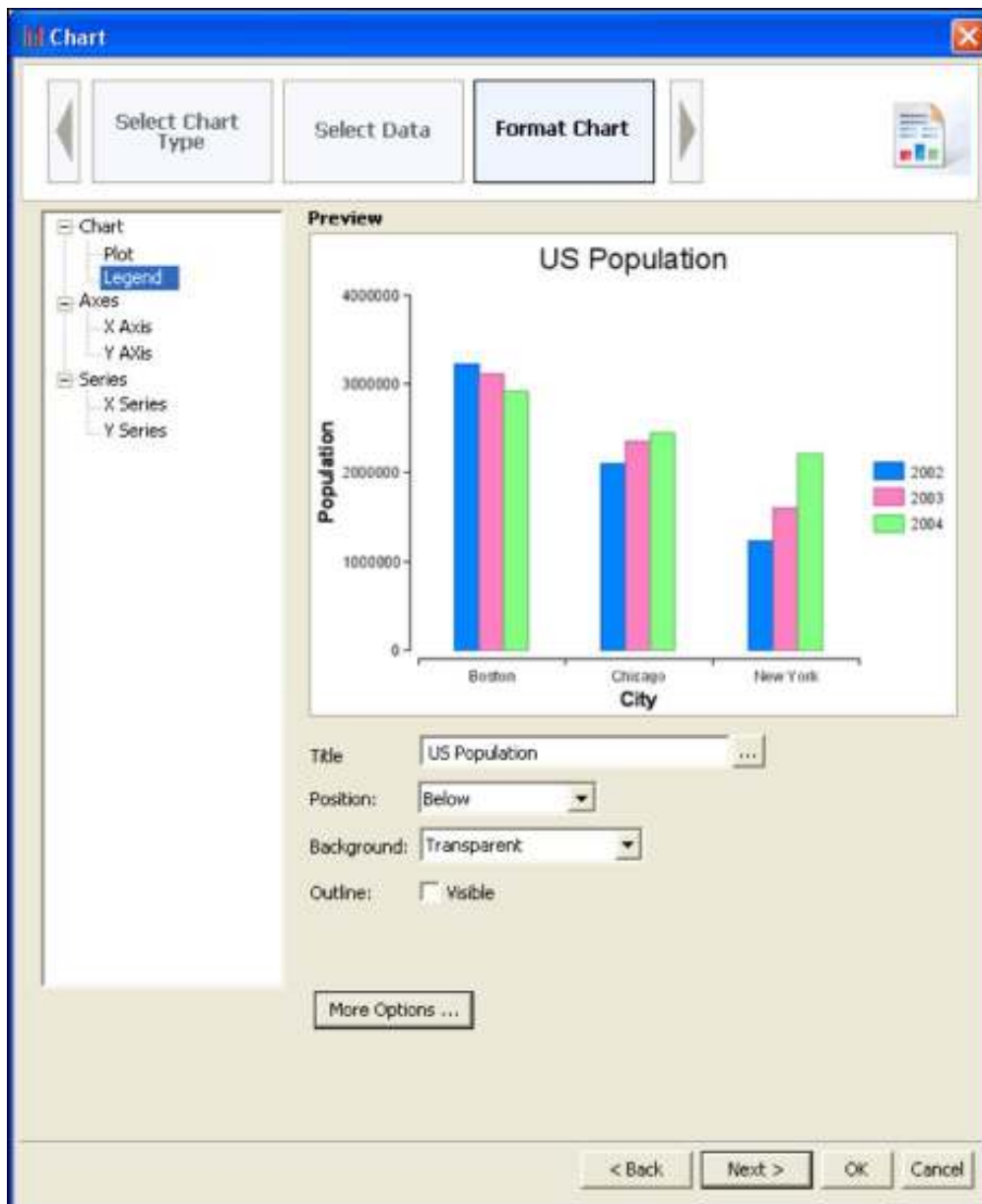


Figure 10. BIRT 2.0: Chart Legend Basic Properties Sheet

Advanced Properties

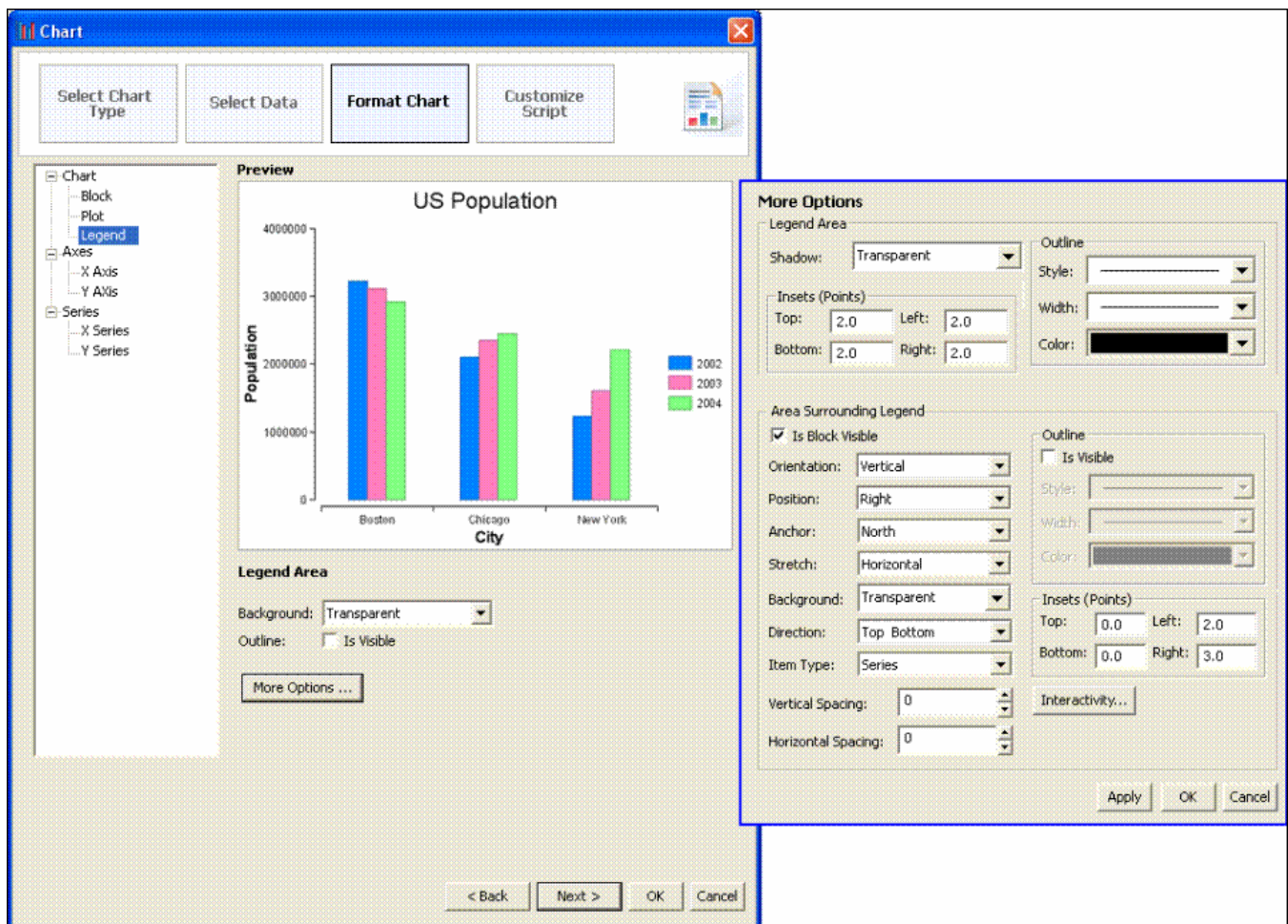


Figure 11. BIRT 2.0: Chart Legend Advanced Popup

Axes

This node will represent the axes in the chart and will allow formatting of general properties related to them.

NOTE: The Axes and Axes - * nodes will only be present in the tree if the chart is a chart with Axes. Also, the UI to control rotation along an axis will only be present if the chart is a '2D with depth' or '3D' chart.

Basic Properties

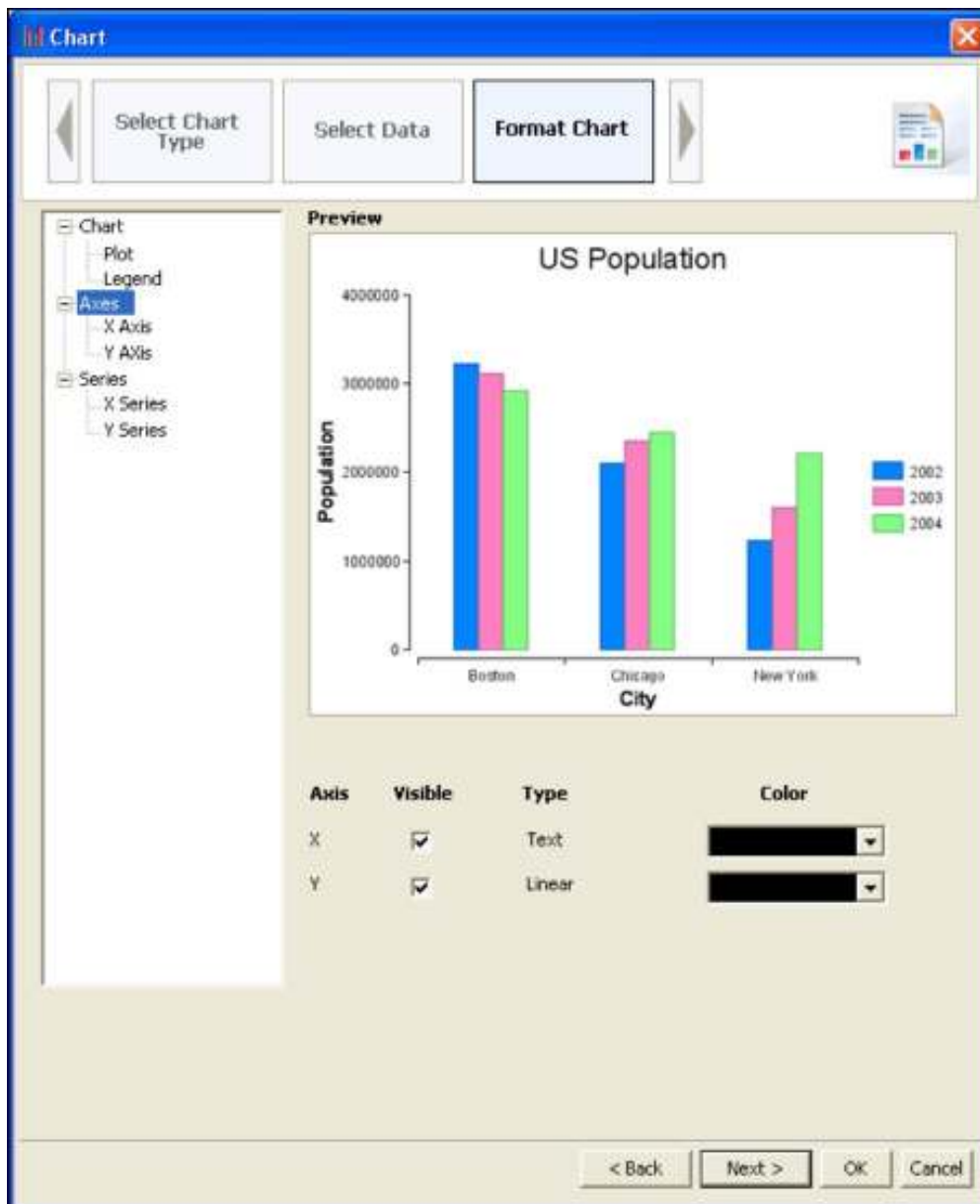


Figure 12. BIRT 2.0: Axes Basic Properties Sheet

Axes – X Axis

This node will represent the X Axis of the chart and will allow formatting of general properties related to it. **NOTE: If multiple X Axes are supported, there will be a limit of two. In which case, there will be two nodes in the tree for X Axes: X Axis 1, X Axis 2. The UI for both nodes will be the same.**

Basic Properties



Figure 13. BIRT 2.0: X Axis Basic Properties Sheet

Advanced Properties

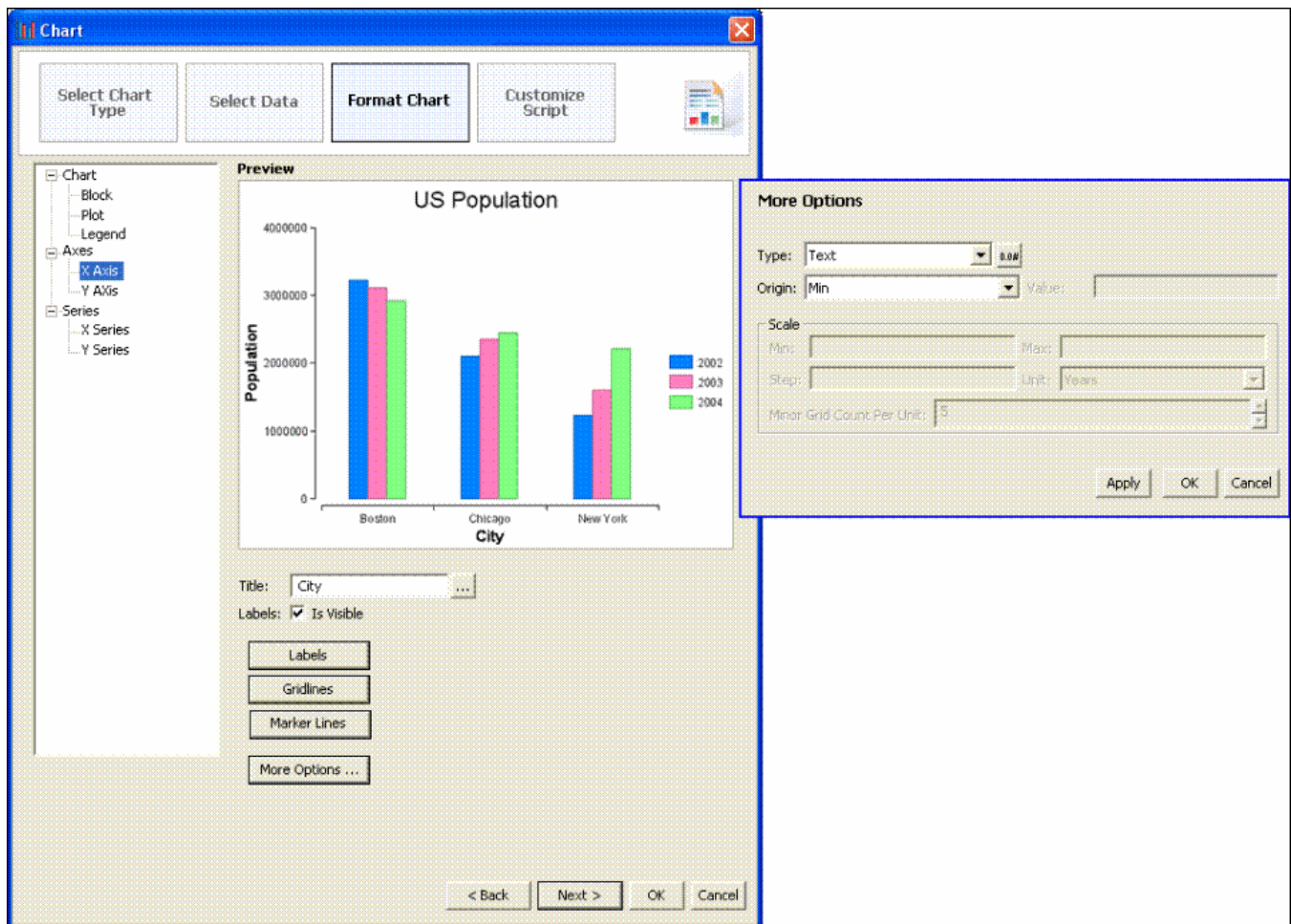


Figure 14. BIRT 2.0: X Axis Advanced Popup

Misc. Popups

Label Properties

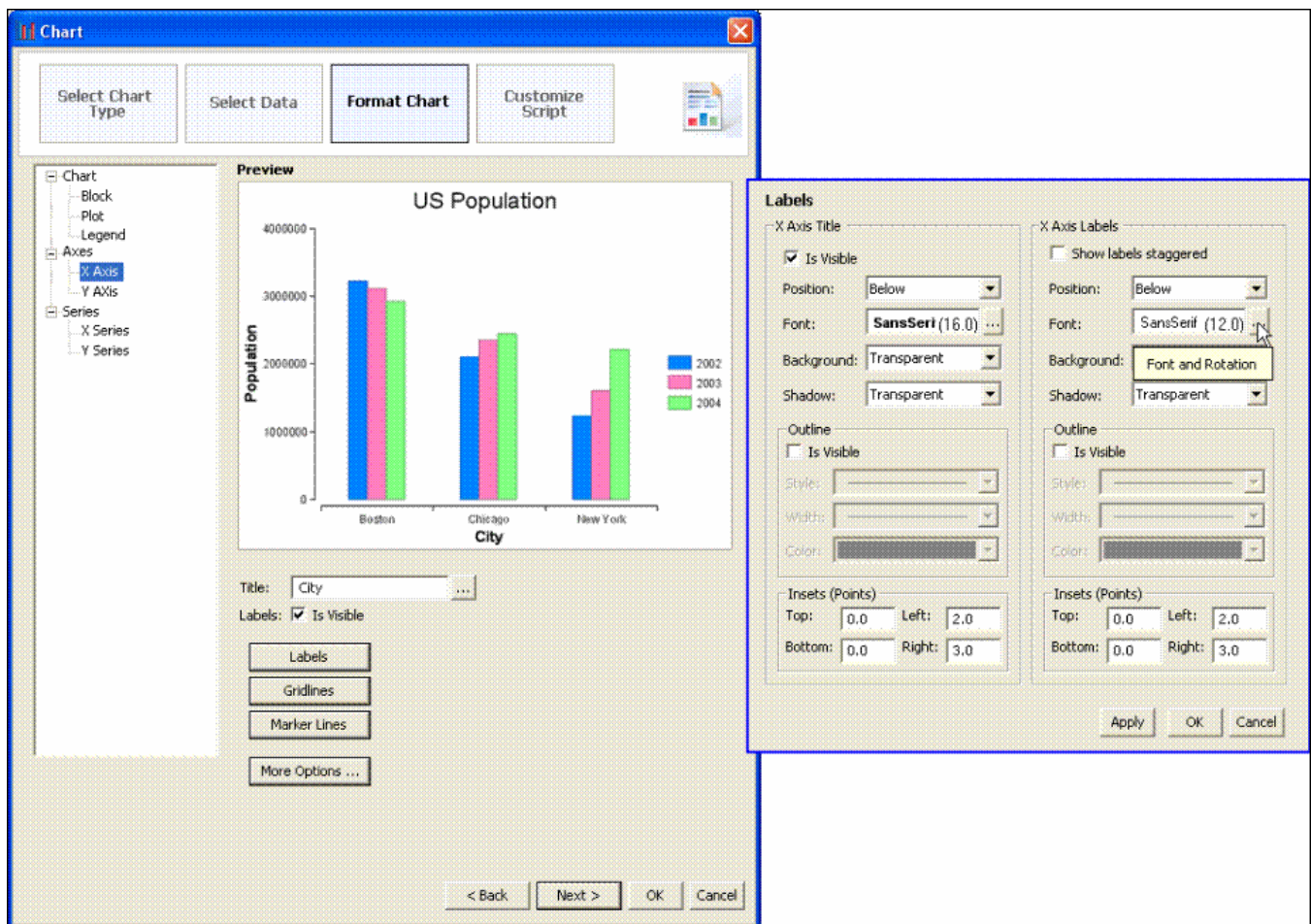


Figure 15. BIRT 2.0: X Axis Label Properties Popup

Gridline Properties

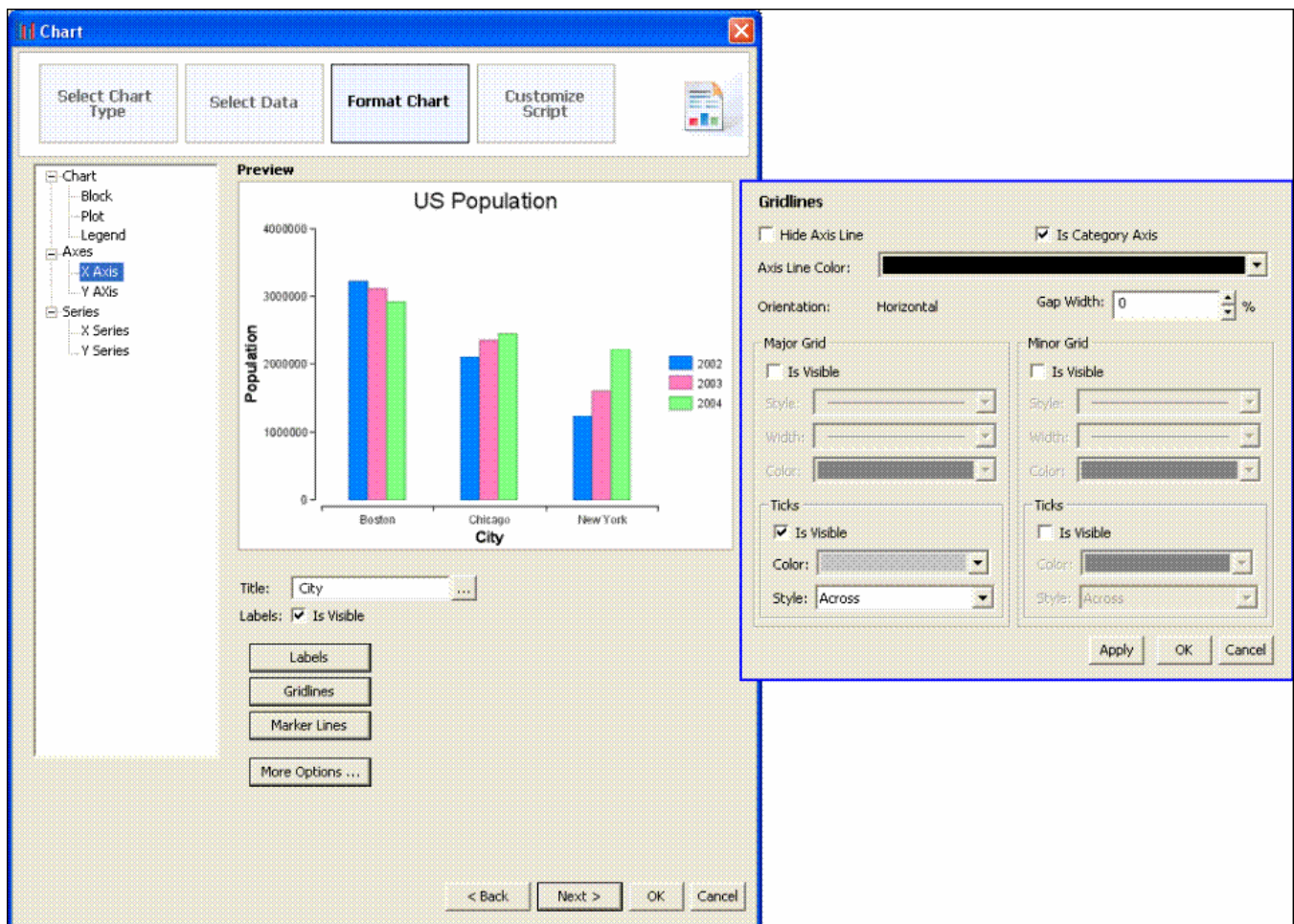


Figure 16. BIRT 2.0: X Axis Gridline Properties Popup

Marker Properties

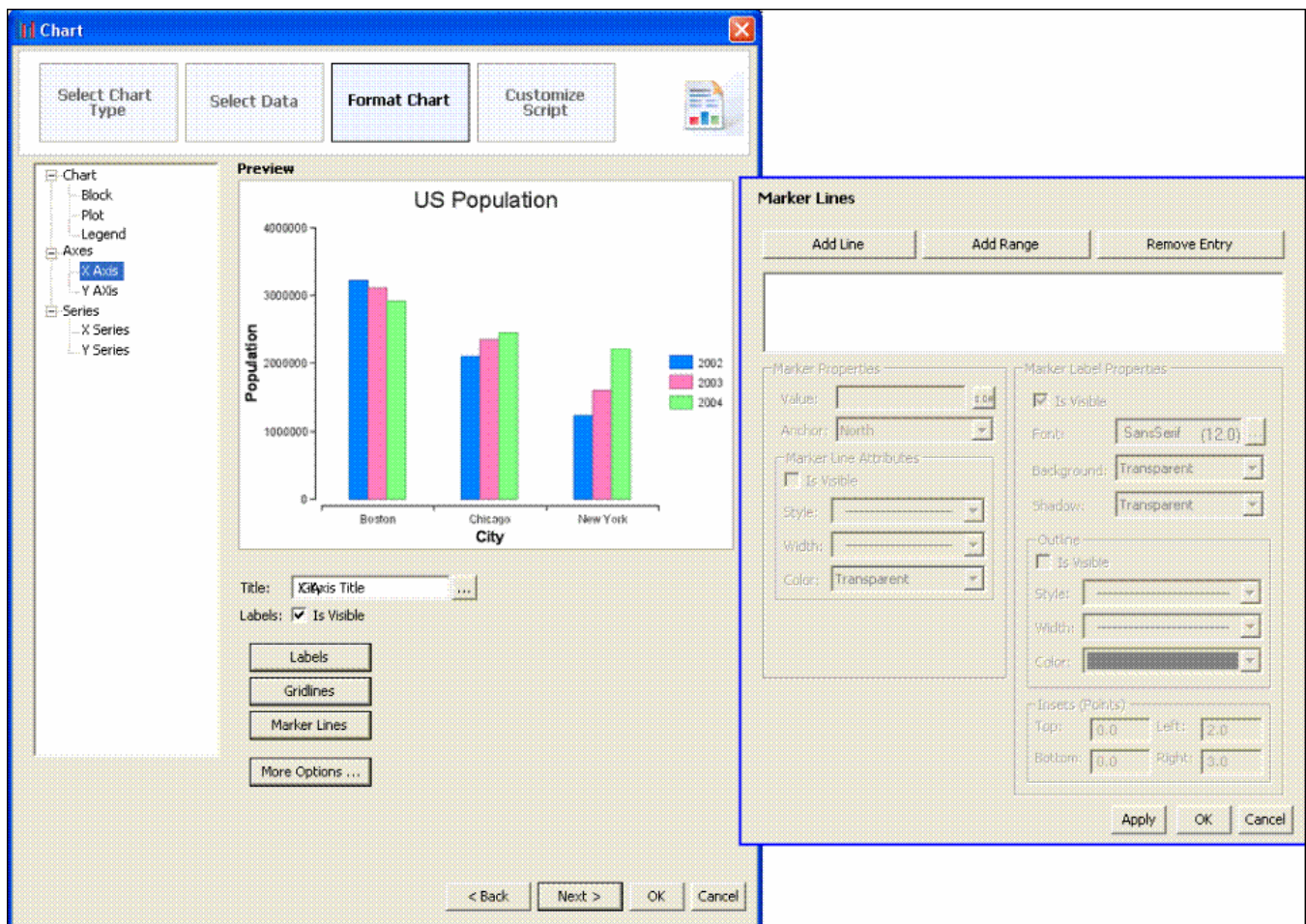


Figure 17. BIRT 2.0: X Axis Marker Properties Popup

Axes – Y Axis

This node will represent the Y Axis of the chart and will allow formatting of general properties related to it. **NOTE:** There will be a limit of up to two Y Axes. One will be the primary Y Axis and the other, the Overlay Axes. To support this, there will be two nodes for these in the navigation tree: Y Axis, and Overlay Axis. The UI for both nodes will be the same. The presence of the Overlay Axis node in the tree will be determined by whether or not the user has enabled Overlay Axes in the type selection screen.

Basic Properties

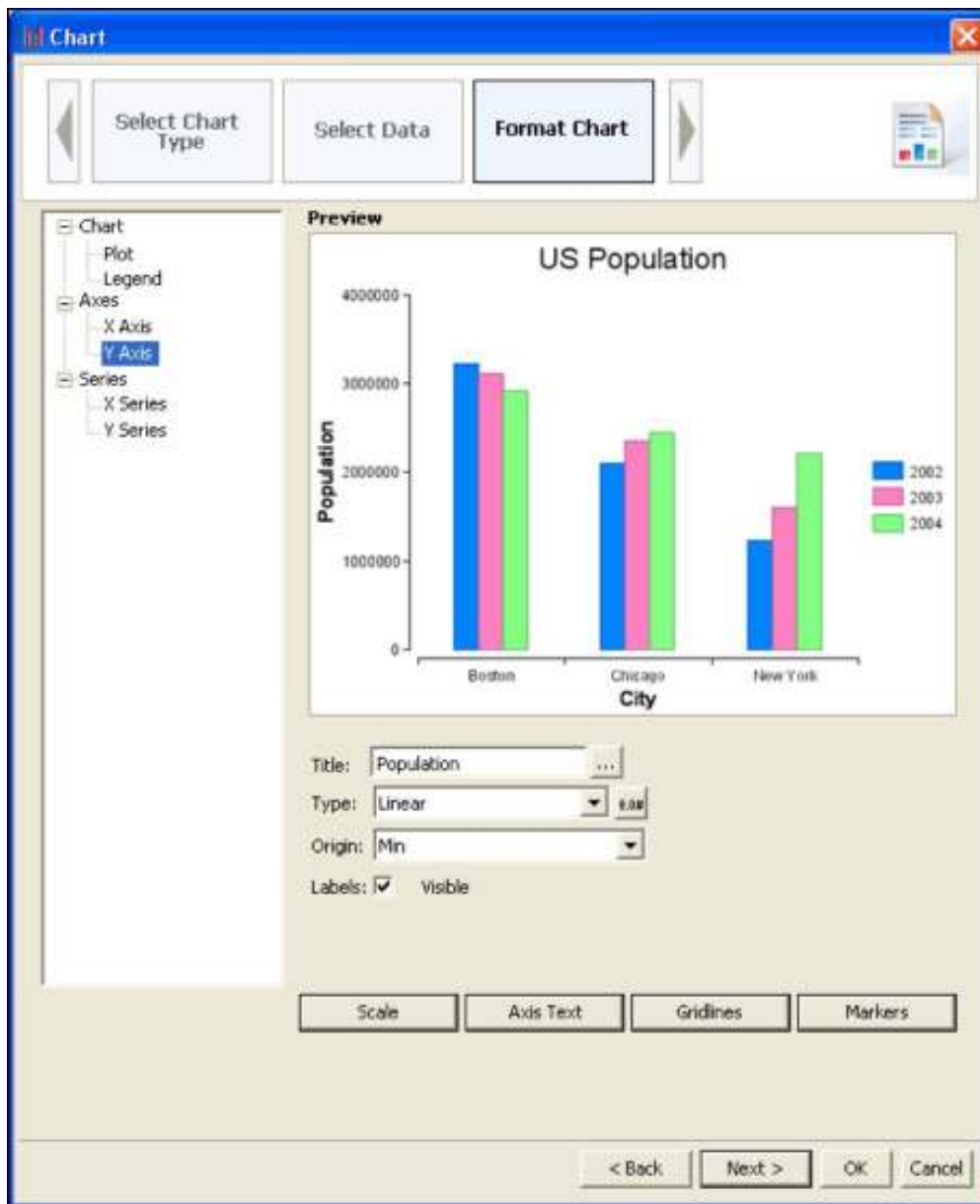


Figure 18. BIRT 2.0: Y Axis Basic Properties Sheet

Misc. Popups

Axis Scale

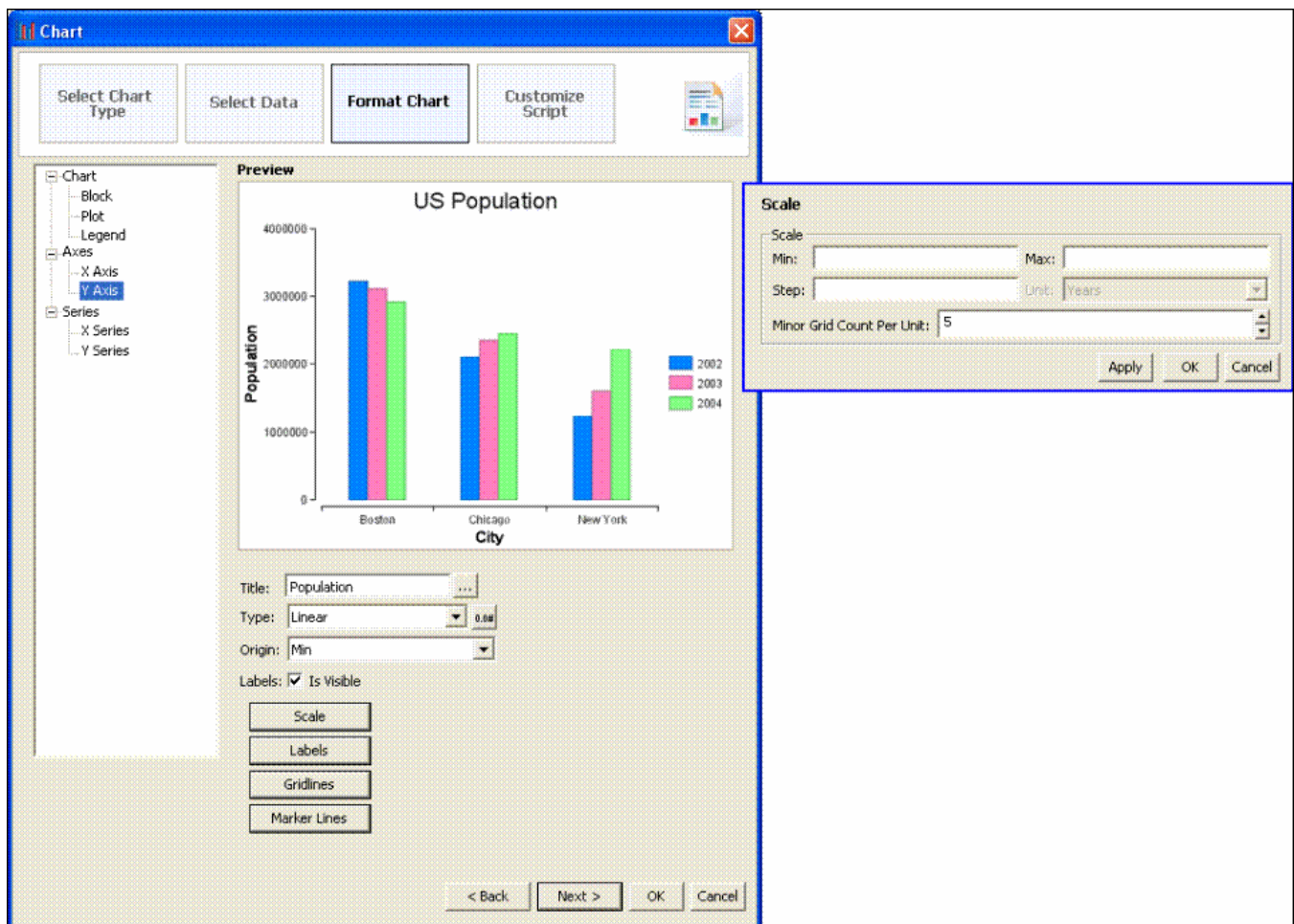


Figure 19. BIRT 2.0: Y Axis Scale Popup

Label Properties

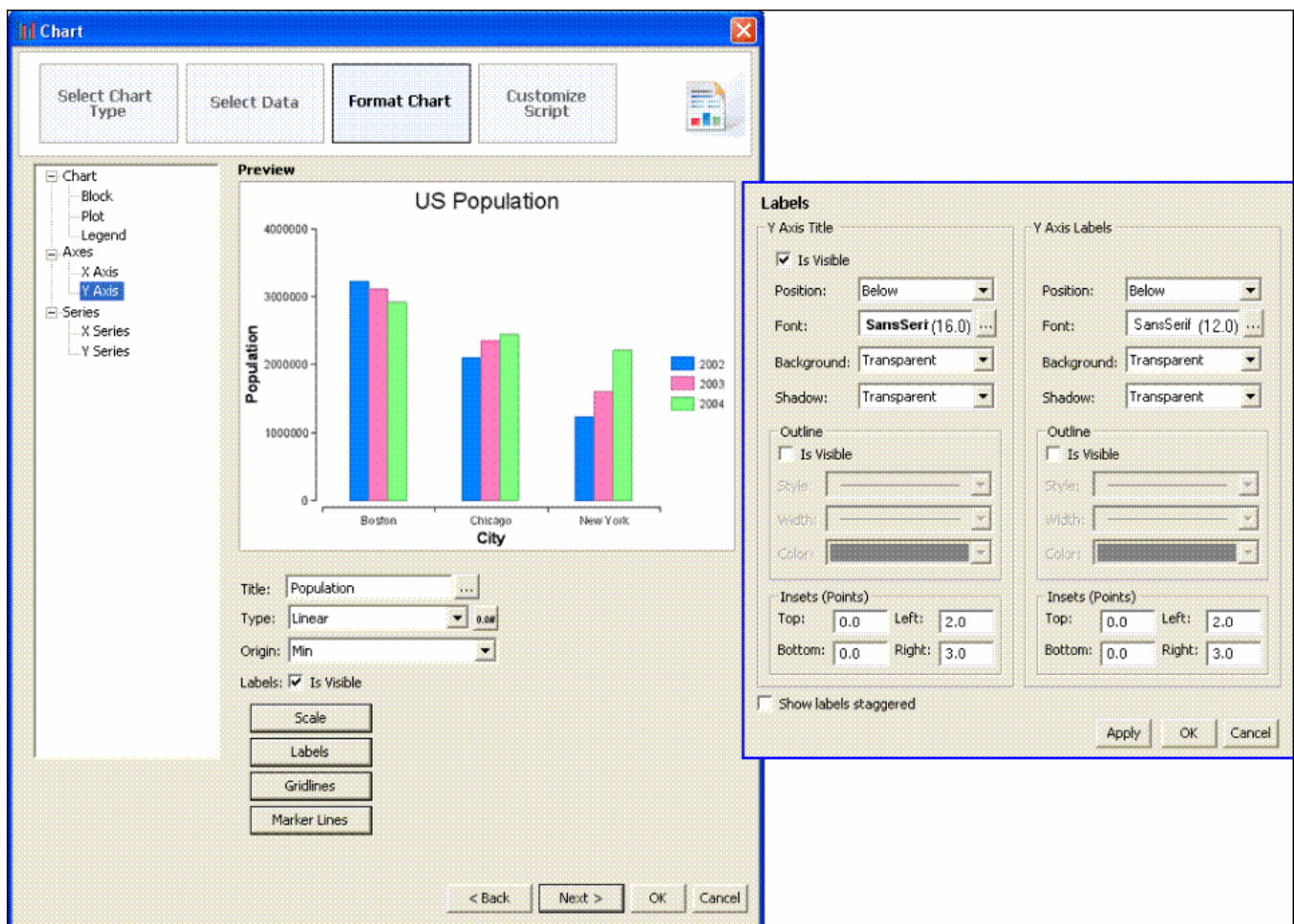


Figure 20. BIRT 2.0: Y Axis Label Properties Popup

Gridline Properties

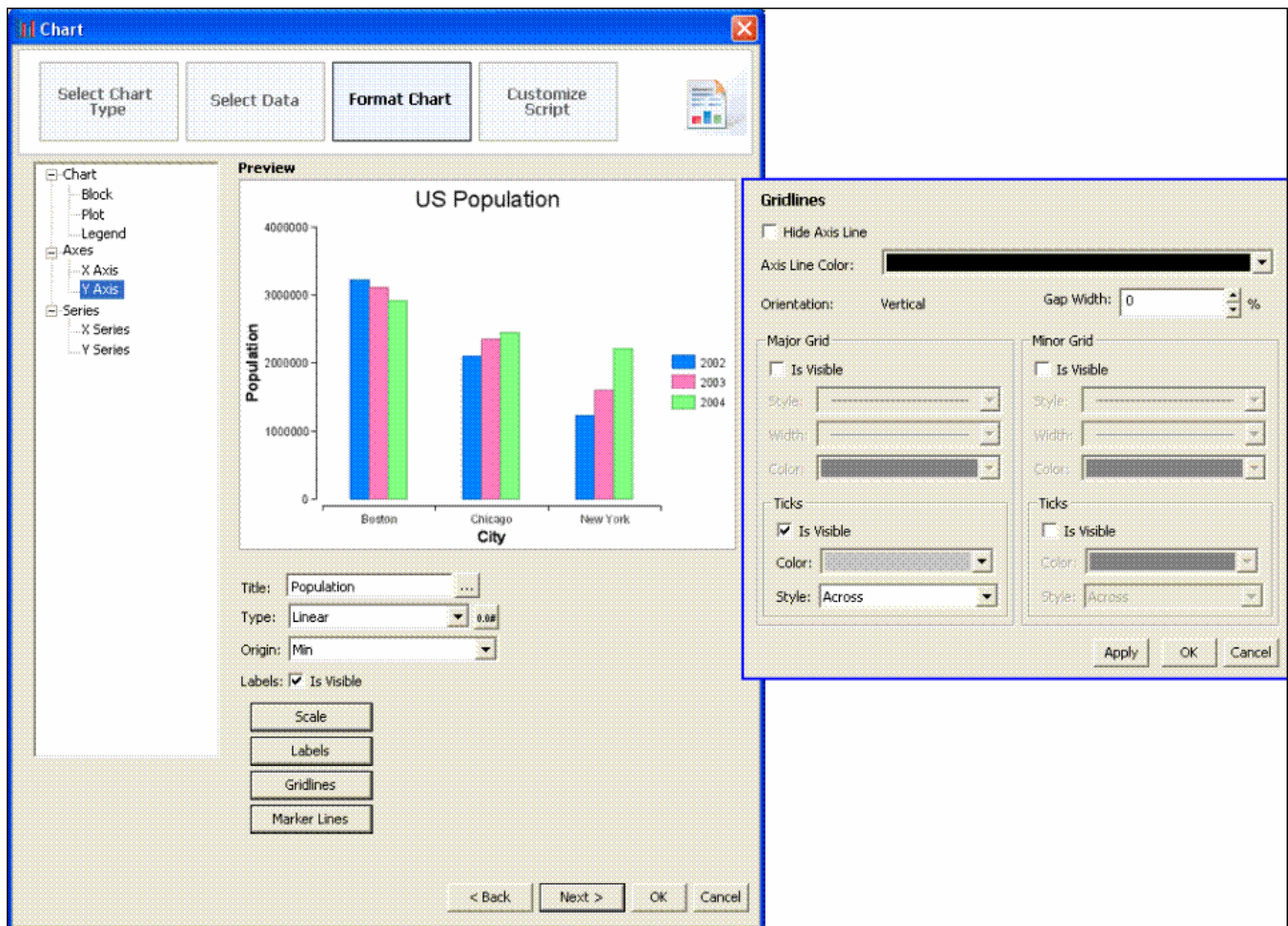


Figure 21. BIRT 2.0: Y Axis Gridline Properties Popup

Marker Properties

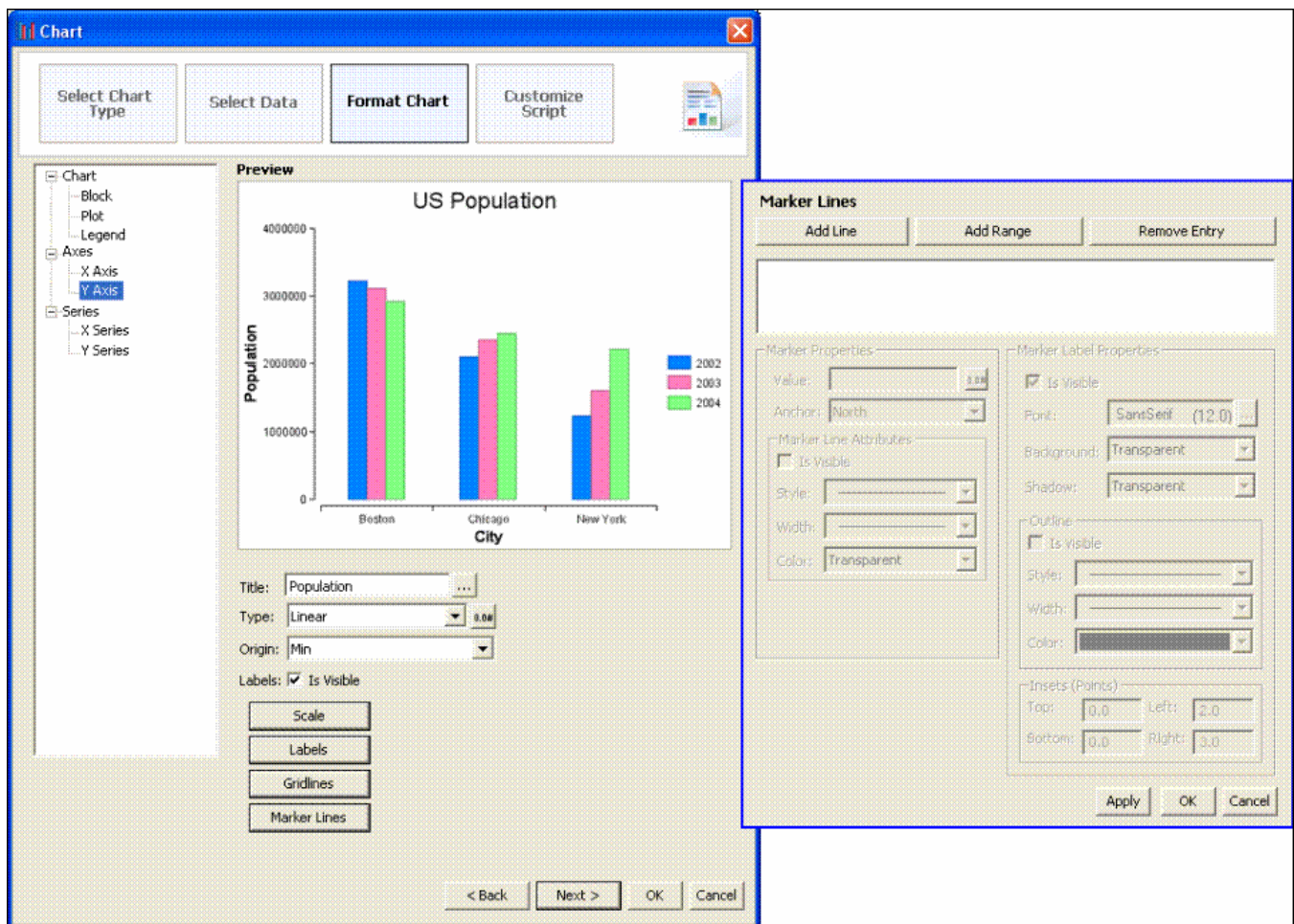


Figure 22. BIRT 2.0: Y Axis Marker Properties Popup

Series

This node will represent all the series in the chart. It will display information about the series and allow some common operations on the series.

Basic Properties

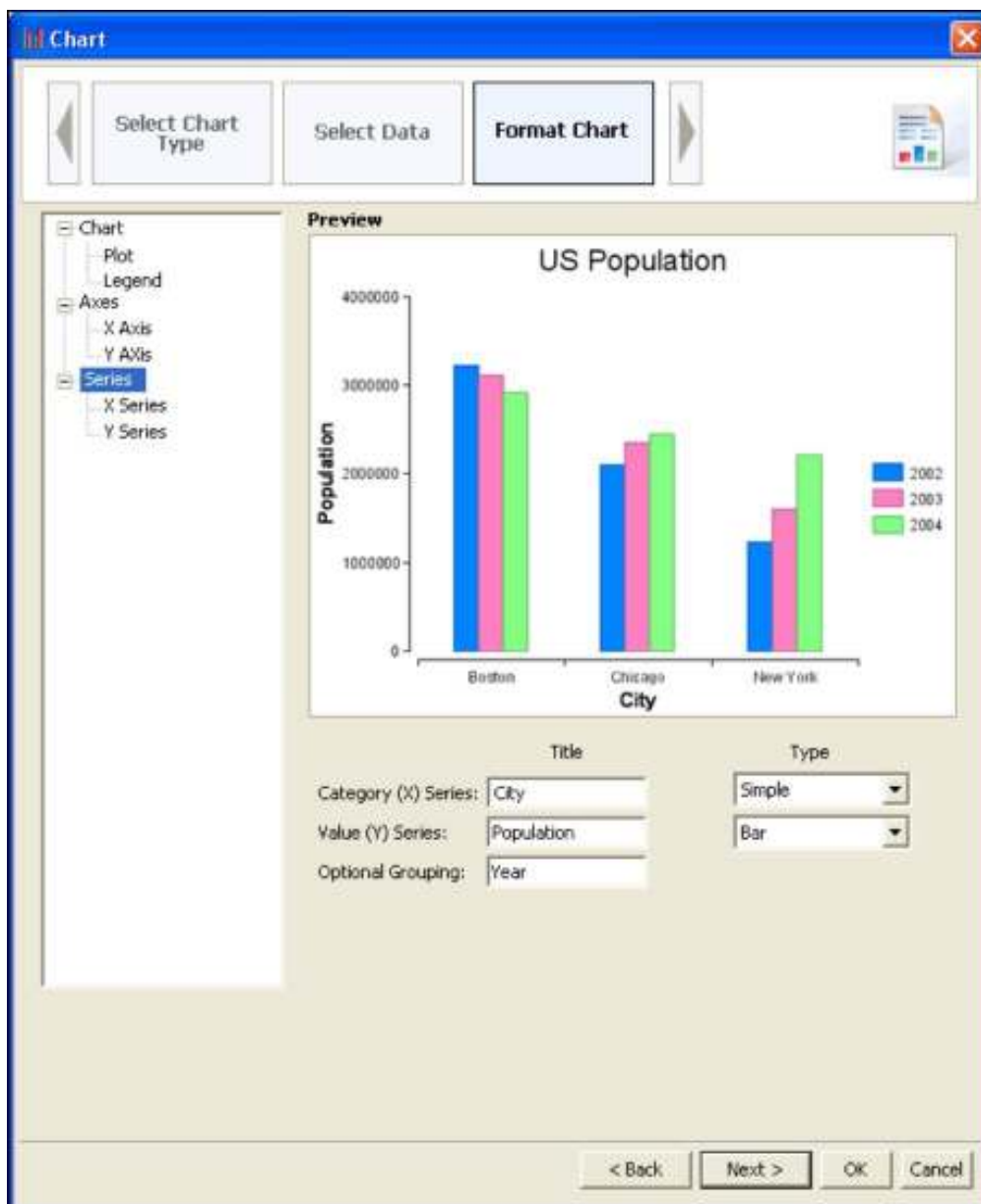


Figure 23. BIRT 2.0: Series Properties Sheet

Series – X Series

This node will represent the X series in the chart.

Basic Properties



Figure 24. BIRT 2.0: X Series Basic Properties Sheet

Advanced Properties

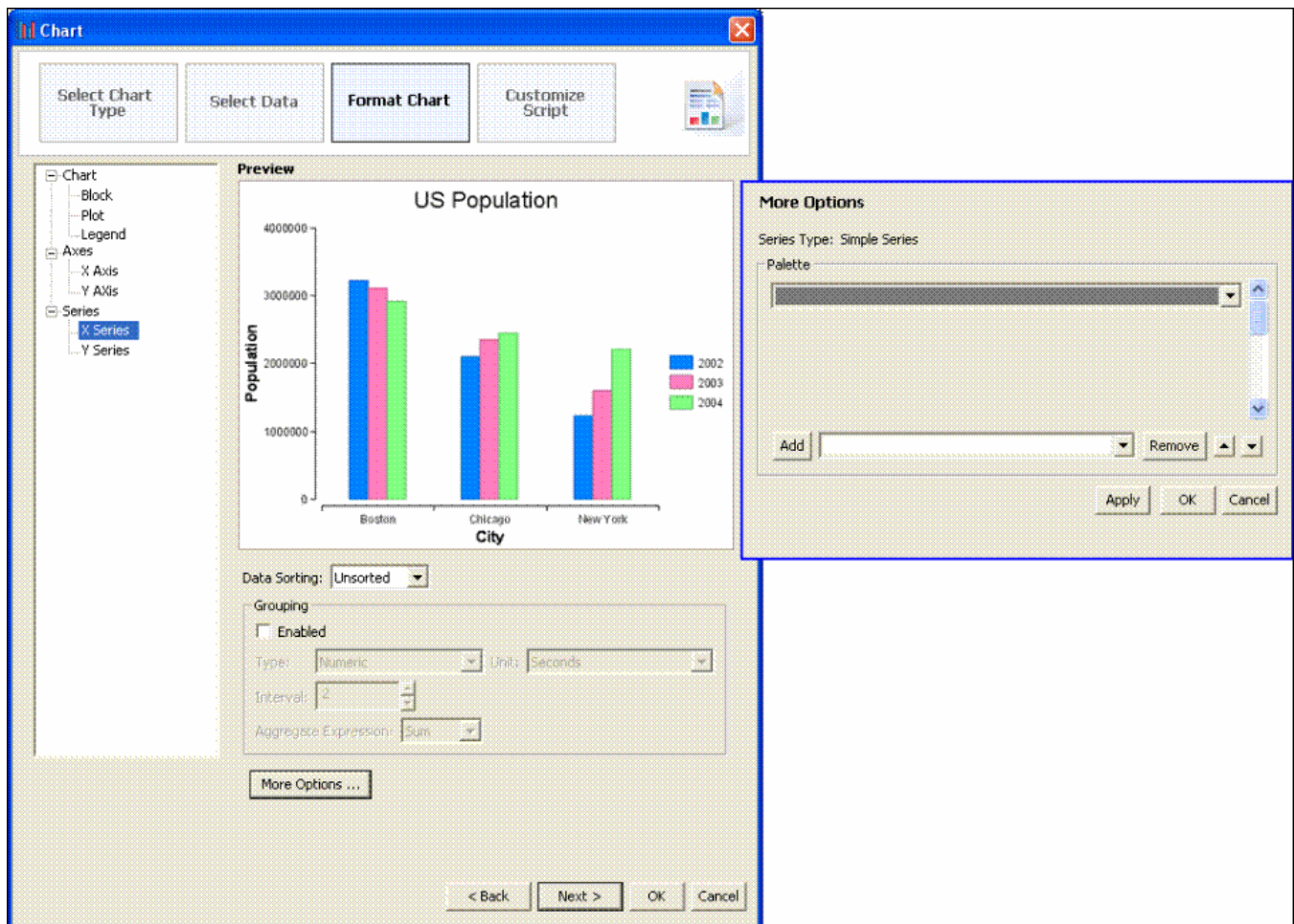


Figure 25. BIRT 2.0: X Series Advanced Popup

Series – Y Series

This node will represent the Y Series in the chart. **NOTE: There will be a limit of up to two Y Series AND two Overlay Series. To support this, there will be two nodes for each in the tree: Y Series 1, Y Series 2, Overlay Series 1 and Overlay Series 2. The UI for all these nodes will be the same. The presence of these nodes in the tree will be determined by whether or not the user has enabled Overlay Axes and whether a data definition has been associated with each in the data binding sheet.**

Basic Properties

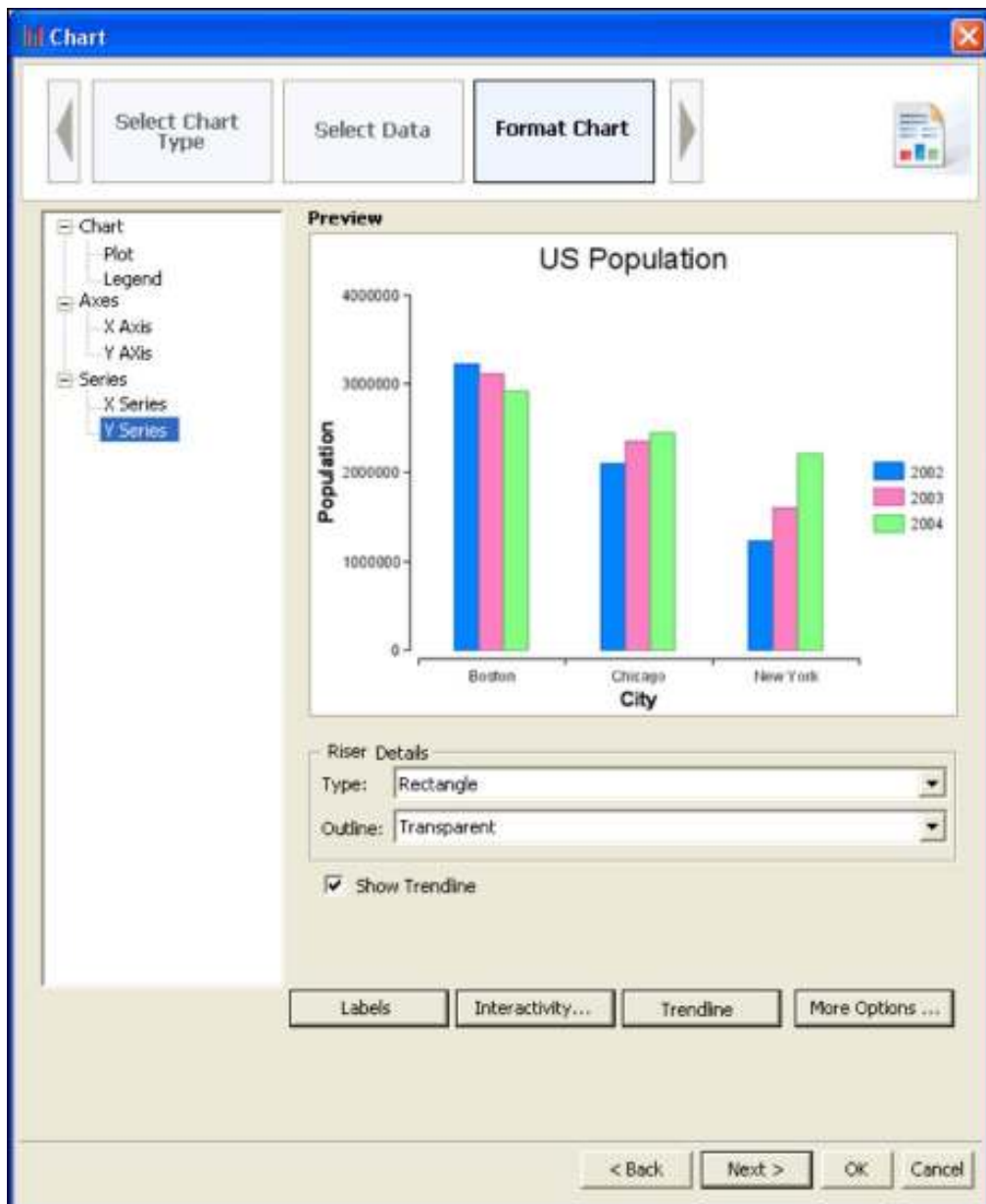


Figure 26. BIRT 2.0: Y Series Basic Properties Sheet

Advanced Properties

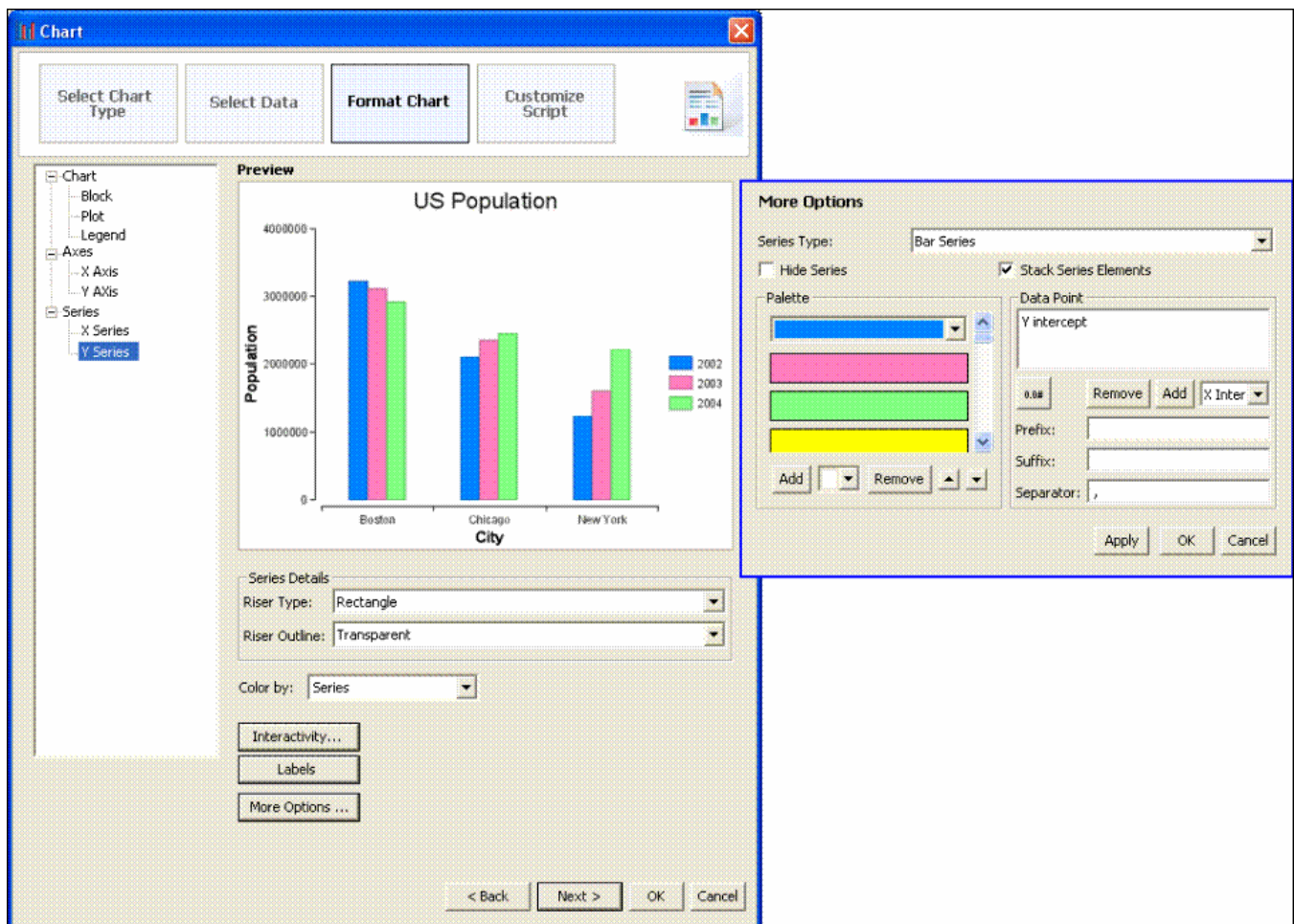


Figure 27. BIRT 2.0: Y Series Advanced Popup

Misc. Popups

Label Properties

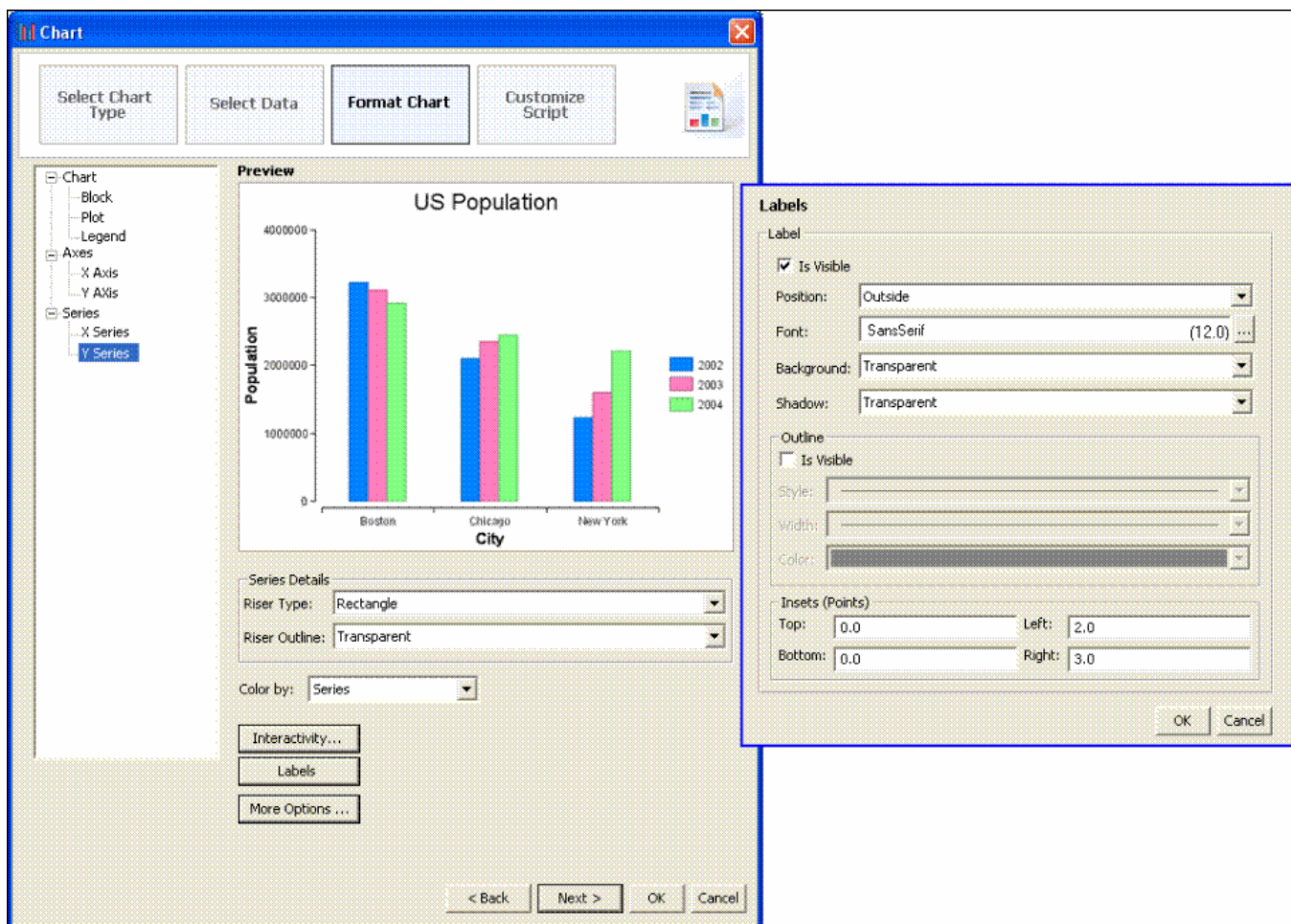


Figure 28. BIRT 2.0: Y Axis Label Properties Popup

Interactivity

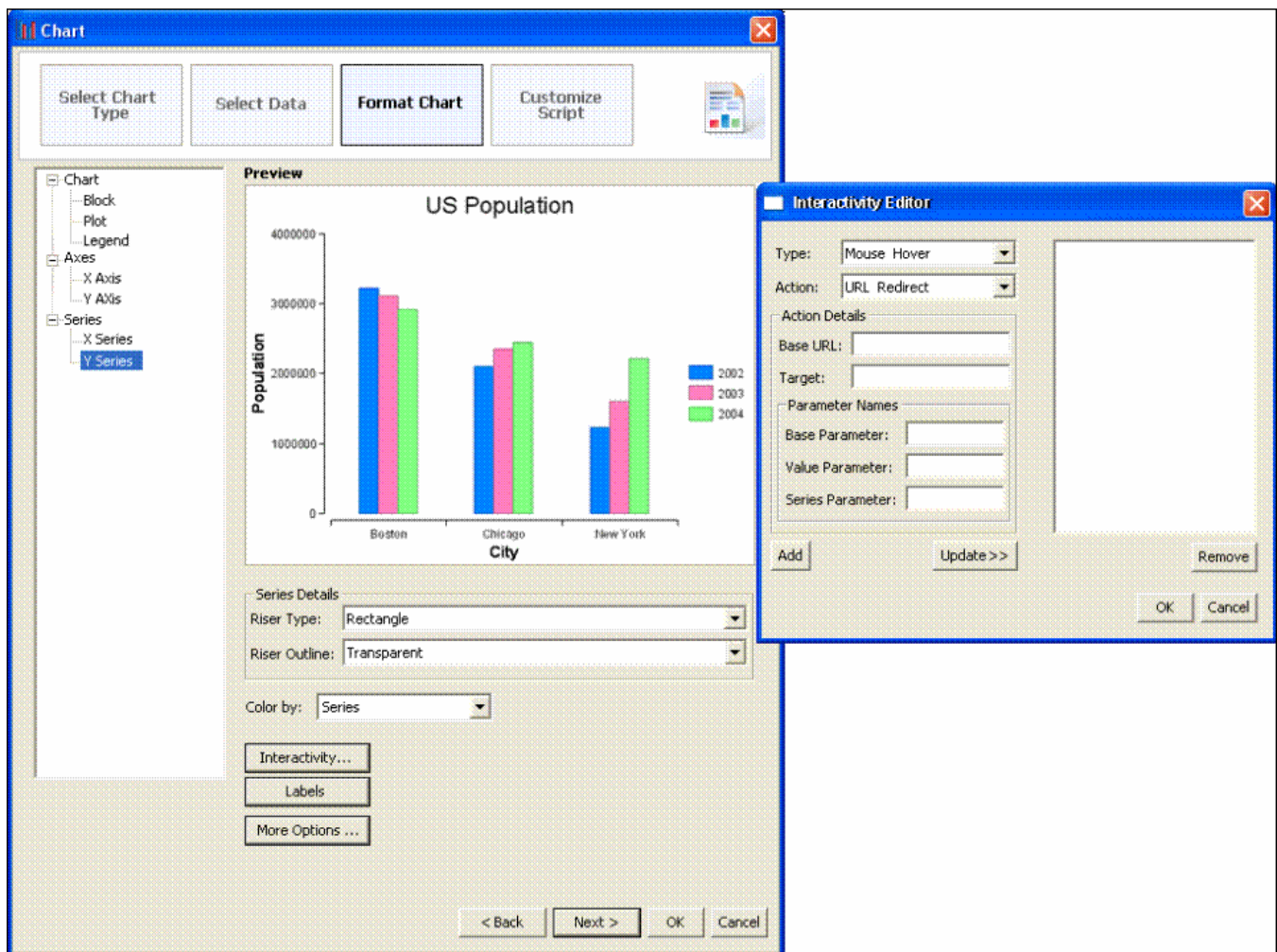


Figure 29. BIRT 2.0: Y Series Interactivity Popup

Trendline Properties

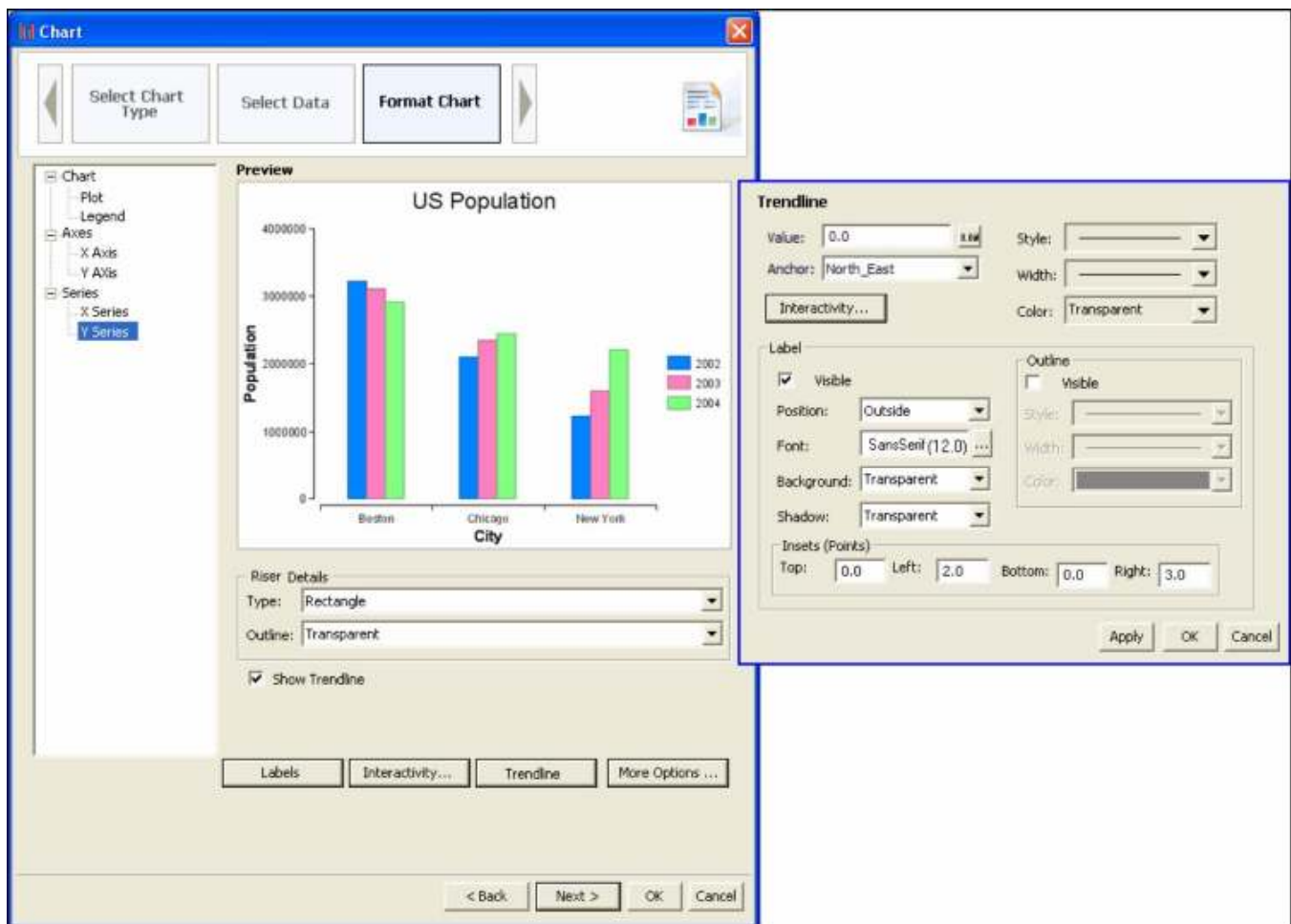


Figure 30. BIRT 2.0: Y Axis Trendline Properties Popup

Extensibility

While the new UI will be significantly more easy to use, it will still support straight-forward extension.

New Extension Points

'tasks' Extension Point

A new extension point 'tasks' will be defined by the framework plug-in. This extension point will be created to allow for registering custom tasks to be used by various task-oriented wizards. The properties available for the extension point will be as follows:

'taskID'* – A string representing a unique task name. This will identify the task to be performed. The taskID will be unique and it will not be possible to register multiple tasks with the same ID.

'classDefinition'* - A fully qualified class name of a class that will provide the UI for the task. This class will be required to implement the ITask interface. Refer to the [framework](#) section above for details of this functionality.

'description' – Optional descriptive text to explain the functionality provided by this task.

* - These properties will be required.

'taskWizards' Extension Point

A new extension point 'taskWizards' will be defined by the framework plug-in. This extension point will be created to allow definition of task-oriented wizards and to specify the tasks used by each.

'wizardID' – A string representing a specific wizard. Although the ID will uniquely identify a wizard, it will be possible to add tasks to the wizard by adding extensions that refer to an already registered wizard. Each subsequent registration will serve to add more tasks to the wizard. **(If a task has already been added to the wizard, it will not be added again.)**

'tasklist' – A comma separated list of taskIDs. These tasks will automatically be added to the wizard upon invocation. The sequence of the tasks will be the same as that specified in this list. It will also be possible to add tasks to a wizard dynamically as part of the wizard framework.

'positionbefore' – A task ID before which the previously specified list of tasks are to be displayed in the wizard.

* - These properties will be required

Support for Old Extension Points

'uisheets' Extension Point

The **'uisheets'** extension point will be supported but in a slightly limited form. It will be used to populate the navigation tree in the formatting task sheet only. All extensions will be displayed under a special 'Custom Properties' node of the tree. The structure of the nodes represented by the extensions will be maintained.

'types' Extension Point

This extension point will be supported with some new functionality added to the interface. If the existing extension is created by implementing the interface, it will need to be modified to implement the additional methods. If it extends the default implementation, no changes will be needed. It will still be suggested that the new functionality be reviewed so that its relevance to each extension can be examined.

'changelisteners' Extension Point

This extension point will NOT be supported. The functionality that could be extended and controlled using this extension point will not be available in the new User Interface.

'seriescomposites' Extension Point

This extension point will be supported and will have the same use as it did in version 1.0 viz. to provide UI components to set attribute and data properties for custom series types. The location and use of these custom composites however will be a little different in the new UI. In this case, the data composite will be used in the data screen to allow setting of the data definition for the series components. The attribute composite will be used in the advanced popup for the series.

Open Issues

1.