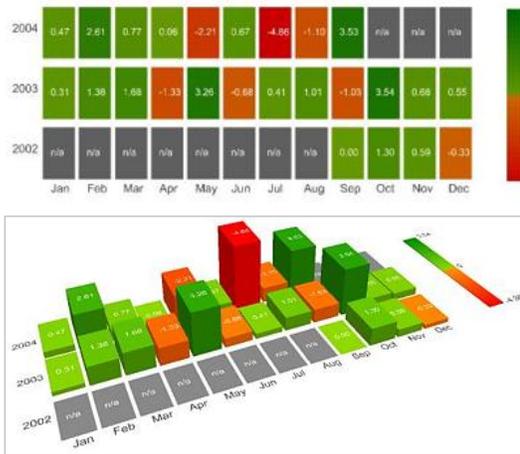


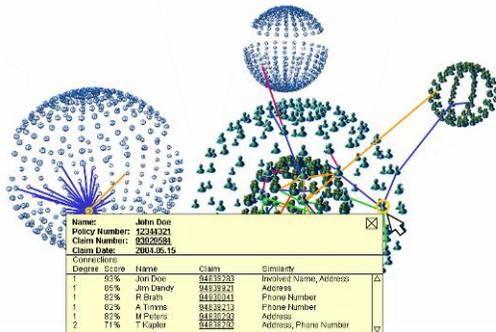
Using O4X™  
 For Your Business Data Visualization Solutions  
**A Technical Overview**

# Introduction

Uncharted™ ([www.uncharted.software](http://www.uncharted.software) formerly Oculus Info Inc.) is the leading provider of innovative business visualization software and professional services for Fortune 500 companies and third-party software providers. Our innovative, multi-dimensional, interactive user interfaces enable our client base to achieve far more comprehension and more rapid understanding of their complex data. This enhanced interaction offers considerable business value. For example:

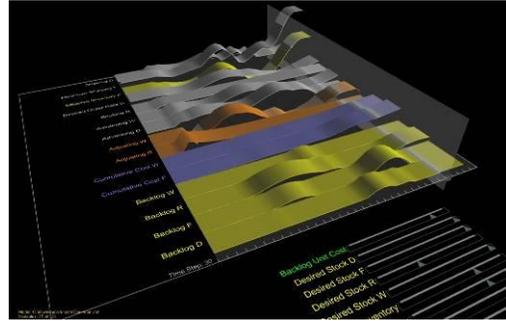


**Figure 1A & 1B: O4X is used in finance for monitoring, analysis and communication. The interactive "heat grid" visualization (above) reveals cells of high volatility (in dark red and green) which is further accentuated when smoothly animated from the 2D "top down" view to the 3D perspective view.**



**Figure 2: O4X is used for displaying complex data such as networks and relationships. Above, a visualization of potential fraud rings uses point and click to follow the linkages between**

**suspect individuals and drill-down to record details.**



**Figure 3: A visualization of a systems dynamics model of an electronics fabrication plant. The user adjusts input parameters (slider bars at bottom right, e.g. orders, raw materials) and all the dependent functions immediately update (ribbon strips in center, e.g. cost, production). The user can easily evaluate "what-if" scenarios on the fly.**

The principals at Uncharted have a long and distinguished history of designing and developing high value, business visualization solutions. The performance, quality and leadership of our group is recognized within the visualization and general business community.

In addition to providing tailored consulting services to our clients, Uncharted has persistently sought out new opportunities to develop our own software toolkits that facilitate custom business data visualization solutions where existing solutions are simply not enough.

On December 29, 2002, Uncharted released the first version of O4X™, a 2D and 3D business visualization development toolkit. O4X has evolved with newer features and has been used in a variety of real-world applications and demanding environments.

O4X can be used to create visualizations ranging from clean, highly interactive visuals for broadcast and communication purposes to rich visualizations of complex business data for data analysis and exploration.

# O4X

O4X is a business data visualization-focused library that is built on top of Microsoft DirectX. It encompasses a set of classes to help create high performance, real-time rendered visualization applications and takes advantage of the many features offered by DirectX. O4X provides efficient rendering of large data sets and sophisticated interactivity to build powerful solutions supporting new methods of human information interaction. O4X can be used with any of the .NET supported languages, including C# and VB.NET, and with WPF or Windows Forms to build interactive 2D and 3D data visualizations.

## Features

O4X contains a multitude of data-visualization-specific classes for creating, positioning and manipulating graphical objects (or groups of objects) in a 3D scene. But it is not limited to 3D. With O4X, developers can also create pure 2D or mixed 2D/3D data visualizations.



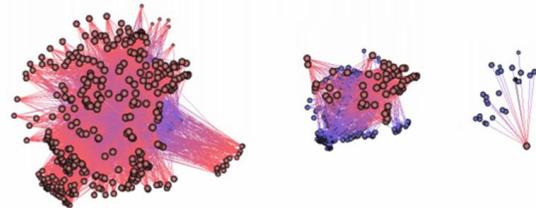
**Figure 4: The smooth scrolling, real-time data driven, 2D logo stock ticker at NASDAQ MarketSite in New York's Times Square is built with O4X.**

One of the primary advantages of using the O4X library is that no low-level graphics programming theory, specialized DirectX knowledge or understanding of graphics hardware is required. Object creation, scene rendering and viewer navigation is managed by the library, providing the ability for developers to create interactive visualization applications with significantly greater ease and speed.



**Figure 5: A fully interactive 3D scatter plot with 300,000 datapoints in O4X.**

Programmers experienced with low-level graphics programming have the option of extending O4X using DirectX. Since it interoperates with pure DirectX, advanced developers can create their own Direct3D graphic objects and add these to the scene.



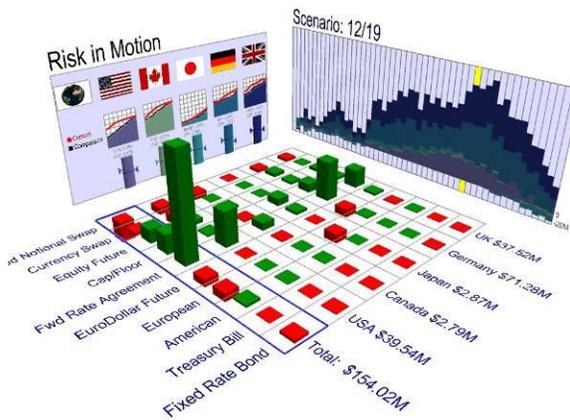
**Figure 6: A visualization of network security alarms as links between computers (attacking computers shown as red spheres, targets shown in blue). The visualization interoperates with other applications including an external data source and other COM-based components.**

## Language Neutral

Because the Microsoft .NET Framework is language neutral, developers are not tied to any specific language when working with O4X.

## Data-source Neutral

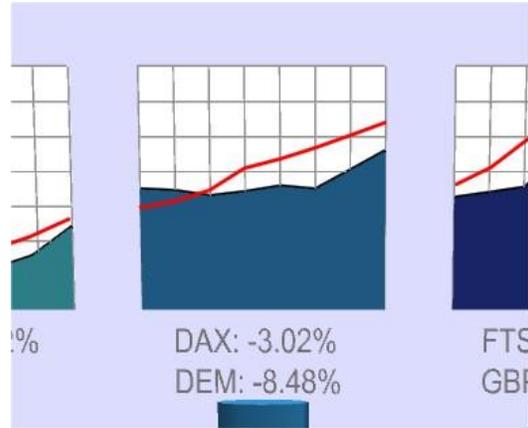
O4X is not constrained to a single data source. Everything from text and XML files, to Microsoft Excel spreadsheets and any database can be accessed via standard data connection methods. This open data architecture ensures connectivity to critical legacy and pre-calculated data as well as real-time data feeds.



**Figure 7: Risk in Motion visualization, built with O4X. This enterprise risk management application visually integrates data from multiple sources into a single scene.**

## Viewer

At the core of the O4X library is the Viewer, which renders the scene. This has been implemented as a Windows Forms control and can be added to any form or used in a composite custom user control. Since the Viewer is a Windows Forms control (like any other standard control) starting to build a visualization application is as simple as dragging and dropping the Viewer onto a form or a WPF control. The Viewer has many different properties (size, position, background color, viewpoint position/direction, etc) and manages all the necessary low-level DirectX RenderState settings. The Viewer also contains and manages its own high performance rendering loop that redraws the scene only when needed and optimizes how objects are given to the video card. It also performs all scene-graph management by only drawing what needs to be drawn at any given time. These features insulate the programmer from the complexity of developing a 'raw' DirectX application.



**Figure 8: A zoomed-in view from the Risk in Motion application provides additional details.**

## Viewer Interaction

The viewer has built-in and transparent navigation controls that permit simplified and intuitive interaction with the scene, including panning, zooming and rotating. By clicking and dragging on the scene, for example, the user can manipulate the viewpoint to highlight and view any position for visual drilldown and analysis. Viewer navigation can also be controlled via Viewpoint objects, allowing the developer to programmatically set up custom viewpoints, or for cycling through for an automated presentation.

## Shapes

O4X supports a number of different shape types that can be used as a template for single or multiple visual elements. These templates include:

- **Quad**
- **Text**
- **Line**
- **Polygon**
- **Cube**
- **Sphere**
- **Cone**
- **Cylinder**
- **Face** (a quad for making a surface)
- **Glyph**

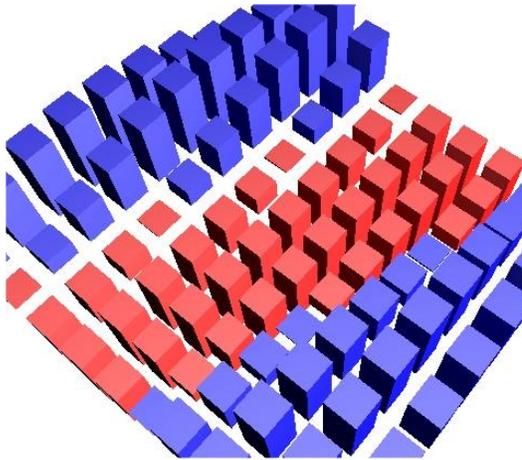
A glyph shape is a custom-made object, created in a 3D modeling package and exported to a Microsoft .X File. This enables complex and detailed models to be presented in a visualization application, without having to resort to building it piecemeal out of simple geometric shapes.



**Figure 9: Signatures from a pen are converted into lines and drawn in real-time on the large screen as the person is signing.**

## Visuals

Visuals are abstract representations of graphic objects used in O4X. Since any Shape can be applied to a visual, changing its appearance simply involves applying a different Shape to the instance. To accommodate multiple instances of a graphical object, the library has **VisualSet** and **VisualSet2D** classes that simplify and optimize the creation of multiple instances of a similar shape.



**Figure 10: VisualSet2D (10 x 10 elements) with a Cube shape. Different properties can be set for each element of a VisualSet2D (color, height etc.)**

The combination of Shapes and VisualSets enable the developer to easily switch between different display types, e.g. a

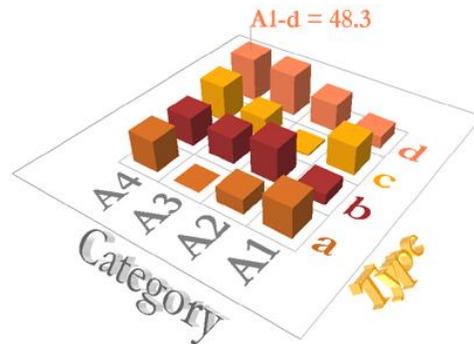
switch from a bar to a line chart is a change of a shape from a quad to line; or a switch from a 3D bar to a 3D surface is a change from a cube to a surface.



**Figure 11: A VisualSet with Cube shape for each city in the US. Polygon and Line shapes are used for visuals that define the map.**

## Text

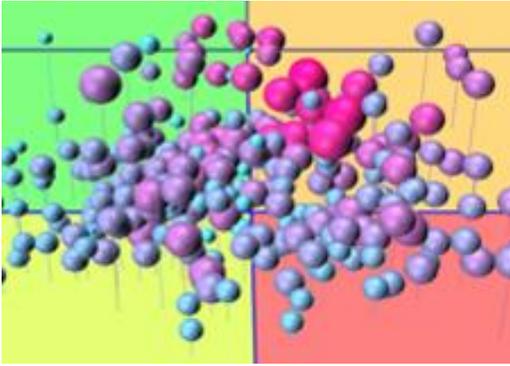
O4X supports 2D text, 3D extruded text, and Screentext (text placed in the overlay). Any font that is available on the system can be presented and placed in the scene.



**Figure 12: A variety of text styles on a 3D bar chart.**

## Lighting

O4X contains several different types of lights including directional, point, spot, and headlight. By adjusting their position, direction and color, the developer can customize and fine-tune lights for the desired visual effect. Multiple lights with different properties can be added to the scene for emphasis and effect.



**Figure 13: Close-up of a scatterplot made with a VisualSet set to a Sphere shape. Note material effects, such as specular highlights, can be used.**

## Materials

Materials can be created and applied to a Shape to change the appearance of an object, e.g. shiny, transparent. Different materials define how light affects the appearance of an object, and can be used for aesthetics, for highlighting or de-emphasizing an object, or indicating data uncertainty.

## Textures

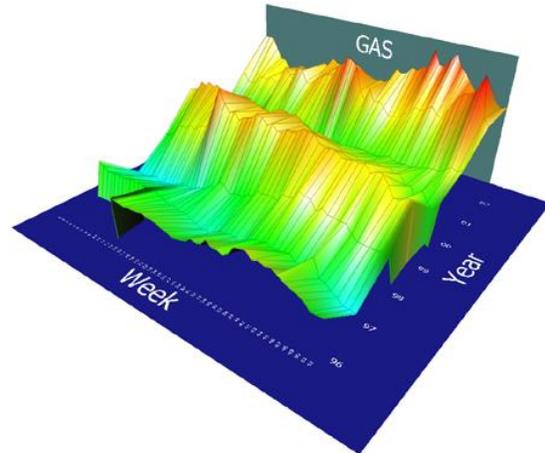
Textures are like wallpaper applied to a surface. For example, satellite imagery can be applied as a texture on top of a mesh or 3D surface representing 3D terrain. O4X supports textures in JPEG, Windows bitmap (BMP), portable network graphics (PNG) formats and more.

## Visual layouts

A Visual represents one or more instances of a similar piece of geometry, arranged according to a user-specified layout. There are two layout manager classes in O4X to help manage regularly structured placement of graphical objects:

- **Row layout** - for incrementally placing geometry along a single dimension e.g. one row of bars.
- **Grid layout** - for row and column placement.

Additionally, a visual object can be placed arbitrarily anywhere in 3D space if managed layout is not required.



**Figure 14: A surface (of prices) is presented by using a VisualSet2D (7 x 52 elements) with a Face shape set in a Grid Layout.**

O4X can use a layout engine to help configure the layout of graphic elements such as the squarified tree map visual layout. The layout can be used by O4X visualizations or any .NET based renderer.

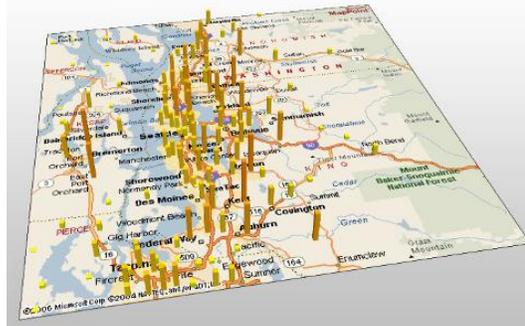


**Figure 15: A treemap layout. Boxes are sized based on data and layout fills the entire space. Note the logos applied as textures.**

Developers are not limited to using the built-in layouts. O4X visual elements expose position, size & rotation properties which allows any layout imaginable, including the use of 3rd party layout libraries, such as graph layouts.

## Maps

O4X has a visual component that makes building 2D and 3D zoom-able tile-based map applications easy. It can connect to an OSM based tile service, and has a simple API to allow developers to easily add other visual elements on top of their map.



**Figure 16: O4X map displaying counts per zipcode in the Pacific Northwest.**

## Groups

Multiple graphic objects can be placed in a 'Group' container with O4X. This is a non-visible scene component that can contain multiple scene components, other groups and lights. By placing scene components into a group, they can be grouped together in the landscape and logically treated as a single object. This facilitates object oriented visual design, by enabling all the sub-components of a complex object to be managed as a single entity. For example, the developer can create a 'BarGraphGroup' object that contains all the visual elements that are necessary to display a bar graph (title, axis labels, bars, grid lines, border etc.) All items contained within a group are translated, rotated and scaled according to the containing group.

## Animation

By adding the standard Windows Forms Timer object, a scene can be used to present data that is animated through time. The visuals in the scene can be adjusted with only one call to the build method on the viewer. The scene can be smoothly animated at frame rates of 50 frames per

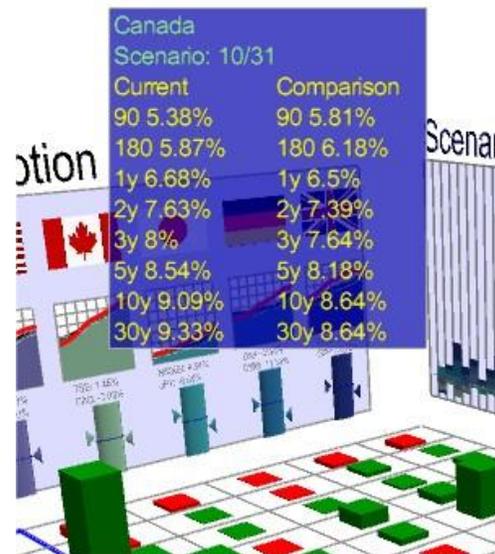
second or more (depending on the complexity of the scene).

## Mouse Interaction

Graphics created with O4X can respond to Windows events (mouse over, mouse click, mouse move etc.) for selection. This makes any object in the scene available for use as a controller for other objects or as a trigger for events and interaction. For example, tapping on a label could be used as a drilldown to filter out only data related to items similar to that class of items.

## Data-rich Tooltips

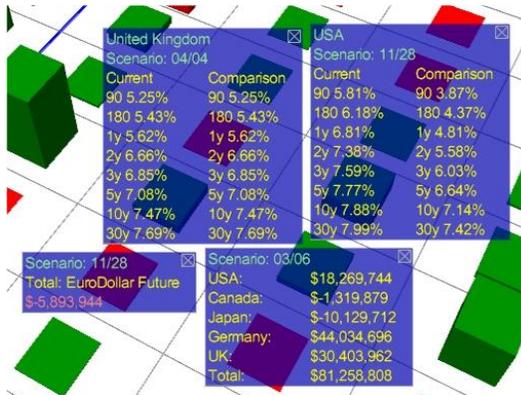
Tooltips are 2D graphics that are presented in the overlay (on top of the scene), typically when the user hovers over a piece of geometry. These can be very effective for presenting additional drill-down details on graphical representations of data. Text data can be laid out in row/column format to assist in readability and alignment of non-fixed width fonts. The content and appearance of the tool-tip – font, text color, background and opacity can all be customized.



**Figure 17: A data rich drilldown Tooltip displaying two rows of detail information.**

## Tooltip Notes

These are extensions of the tool tip object that allow multiple tool-tips to persist in the overlay for visual reporting for ad hoc annotating, or side-by-side comparisons. Tooltip notes can be re-positioned in the scene by dragging on them with the mouse, and can be closed by clicking on their 'X' box button.



**Figure 18: Multiple Tooltip Notes can be positioned and overlaid onto a scene.**

## Microsoft .NET GDI+ Integration

Lightweight 2D graphics created with Microsoft GDI+ can be mixed with added to the scene. For example, GDI+ graphics defining a simple pie chart can be mapped onto a quad in the 3D scene (or, added to the 2D overlay) as a texture.

## WPF Integration

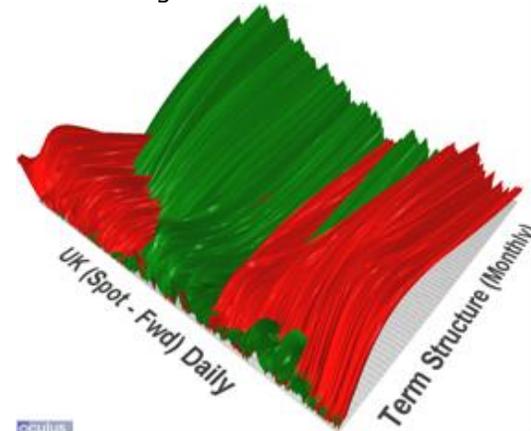
Using a WPF WindowsFormHost the O4X Viewer control can be integrated into a WPF based application.

## Excel Visualizer

### Excel Framework

Excel is a highly popular tool for collecting, organizing, analyzing and modeling data. It is frequently a desired data source for visualization. Uncharted Excel Visualizer Framework option provides a starting point for creating interactive visualization directly tied to an Excel spreadsheet.

Excel Visualizer will run on any Excel spreadsheet out-of-box (Excel 2007 or newer): it does not require the spreadsheet to be organized in any way. It is inherently scalable to large amounts of data using the core O4X engine.



**Figure 19: Excel Visualizer view of 36,000 cells directly from an Excel spreadsheet.**

Using the Excel Visualizer Framework as a starting point, the visualization development effort is faster. The Excel Visualizer source code can be adapted to meet specific needs, such as trading spreadsheets for price modeling; risk spreadsheets; etc.

## Deployment Strategies

### Strategy 1: Standalone Client Application

This is the standard implementation, where an application is deployed to the client via an installer. Built with Windows Forms or

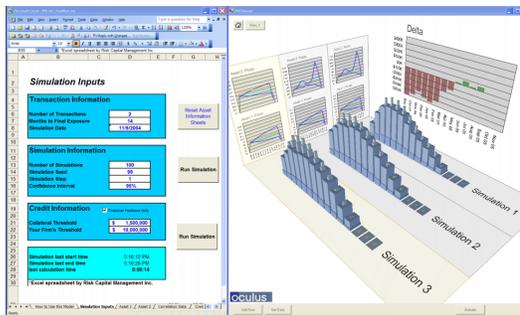
WPF, this solution could be displayed either in a window or full screen for a more immersive application.

## Strategy 2: Integrated into Other Applications

O4X can be integrated into other applications, either developed internally, or popular third party programs.

## Strategy 3: Office Add-in

O4X can be used as part of an Office add-in. For example, O4X visualizations can be created as a plug-in to Microsoft Excel, providing a rich interactive interface for directly interacting with Excel spreadsheets; and immediately seeing changes when the spreadsheet updates.



**Figure 20: An Excel add-on visualization which is updated as the data in Excel is recalculated. O4X is popular with risk modeling spreadsheets, such as the one above from Risk Capital Inc. ([www.riskcapital.com](http://www.riskcapital.com)): the visualizations enable the viewer to see the large amount of data generated by different models that would be otherwise difficult to see in grids of numbers or impossible to depict with the spreadsheets built-in charts.**

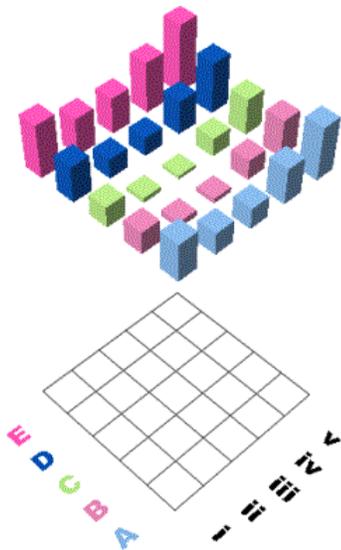
## Benefits

O4X provides a simple flexible interface for creating tailored visualization solutions. The library allows the developer to focus on the business solution without getting bogged down in the details of low level device and graphics manipulation. This ease of use is combined with the flexibility to develop truly customized solutions which would not be available from ready-made charts or pre-packaged visualizations.

**Flexibility:** O4X provides a simple consistent interface that can be used to create a wide range of visualizations. For example, a simple 3D bar chart, rather than being a single monolithic class, can be made with four instantiations:

- A VisualSet2D of Cubes
- A VisualSet2D of Lines
- Two VisualSets of Text

With only these three classes and the associated methods, complete flexibility is provided - flexibility that cannot be achieved with a monolithic component class with hundreds of methods.



**Figure 21:** In O4X a bar chart is created with four instantiations of three classes.

Other alternatives, such as a low level API like DirectX or OpenGL, requires additional specialized programming effort, such as rendering optimization, scene management and viewer navigation. O4X represents an effective middle ground.

**Easy Data Management:** With O4X data is kept entirely separate from the visual elements. Using O4X means that data is not repeated multiple times throughout the application, nor does the data need to specially transformed to fit the specific format required by the visualization. Data can be managed centrally and it is easy to attach and update multiple visualization elements to the data. Furthermore, O4X does not prescribe any pre-set data model or data format, giving the developer flexibility to use the most appropriate data model for the data.

**Easy Interactions:** The O4X library makes it easy to extend and enhance visualizations with rich interactions, through events such as click; click and drag; and drag and drop to enable features such as animation, tooltips, and drill-down.

**Easy Graphics Management:** O4X facilitates the management of detailed visualizations and the creation of re-usable visual components with techniques such as Layouts, VisualSets, Grouping and OO-techniques (e.g. sub-classing), which promotes easy-of-programmability, performance and scalability.

## Conclusion

The O4X™ business data visualization development toolkit is a state-of-the-art class library that leverages the power of Microsoft DirectX.

Combining the power and ease-of-use of the O4X toolkit with the proven, widely deployed Microsoft environment ensures that this large client base can now take advantage of new methods of human information interaction, while maintaining their investment in Microsoft-based applications and tools.

The power inherent in multi-dimensional user interfaces, allows for far more

comprehension and better understanding of complex data. This enhanced interaction adds up to huge revenue potential and considerable business value.

## Professional Services

Uncharted provides services above O4X to assist our customers get maximum value from visualization. Uncharted is a full service company: in addition to training and knowledge transfer, Uncharted professional services help our customers get the right visual designs to meet their business objectives and user capabilities; design application architectures to integrate within the enterprise environment; and develop prototypes, pilots and full production visualization applications.

## Contacts

For more information on O4X and other Uncharted visualization development toolkits, please email us at [info@uncharted.software](mailto:info@uncharted.software) or contact:

### **Richard Brath**

Partner and Director of Business Development  
Telephone: (416) 203-3003, ext. 242  
[richard.brath@uncharted.software](mailto:richard.brath@uncharted.software)  
Toronto Office

### **William Wright**

Sr. Partner and President  
Telephone: (416) 203-3003, ext. 227  
[bill.wright@uncharted.software](mailto:bill.wright@uncharted.software)  
Toronto Office

## Office Location

Uncharted Software Inc.  
2 Berkeley St.  
Suite 600  
Toronto, Ontario M5A 4J5

Telephone: (416) 203-3003  
Facsimile: (416) 203-0646

[www.uncharted.software](http://www.uncharted.software)

### Legal Notice:

Any redistribution or reproduction of any materials or information contained herein is strictly prohibited. This publication and the information may be used solely for personal, informational, non-commercial purposes, and may not be modified or altered in any way. Recipient may not remove any copyright or other proprietary notices contained in the documents and information. Throughout this paper, trademarked names are used. Rather than put a trademark symbol in every occurrence of a trademarked name, we state that we are using the names only in an editorial fashion and to the benefit of the trademark owner with no intention of infringement of the trademark.

Microsoft, DirectX and .NET are registered trademarks of Microsoft Inc. O4X is a trademark of Uncharted Software Inc. Other company names, brand names or product names mentioned herein may be trademarks and/or registered trademarks of their respective owners and companies.