DRAKES BAY COMPANY

US6538654 Claim Chart

Adobe Flash – Stage3D

DRAKES BAY COMPANY

Disclaimer

The patent right(s) referenced in these materials are offered for acquisition through a sale as of the date of this publication. The patent owner may, at any time, in its sole discretion and without prior notice, modify, substitute or withdraw this offering, and may modify any and all terms and conditions related to the acquisition including, but not limited to, pricing. Neither these materials nor the patent-related information referred to in these materials is intended to constitute legal analysis, notice or accusation of infringement, or to claim or assert that any party has violated any intellectual property rights or law. Prospective purchasers must rely on their own examination and evaluation of the patent(s), including but not limited to review of associated patent applications, patent family members, file histories, and other relevant information, in determining the value or applicability of the patent(s) with respect to any activities, products, or services. Any purchase is to be made independent of these materials. Neither the patent owner nor Drakes Bay Company makes any representation or warranties regarding the patent(s), the information contained in or referred to in these materials, or any other information supplied in connection with this offering. The patent owner and Drakes Bay Company expressly disclaim any warranty of merchantability or fitness for a particular purpose in connection therewith. The patent owner, in its sole discretion, may reject any and all offers for any reason. No obligations or commitments, whether express, implied or otherwise, shall be enforceable against the patent owner unless and until a final, written agreement is executed by and between the patent owner and a prospective purchaser.

Contents

Bibliographic Data

US6538654: System And Method For Optimizing 3D Animation And Textures.

Issue Date: March 25, 2003

Filing Date: December 23, 1999

Priority Date: December 24, 1998

Inventors: Anthony Rose, Andrew D. Davie, Alexis Vuillemin.

Original Assignee: B3D Inc.

Current Assignee: B3D Inc.

Independent Claims: 9 (Claim #1, #6, #7, #8, #10, #11, #12, #14 and #15)

Total Claims: 15

Applications: The invention can be used in following application areas :

- In 3D animation object development and rendering software
- Animation film making studios.

Summary: The subject patent relates to 3D animation software. It discloses a method for determining and removing the animation data of 3D objects (actors) which are outside the view frame.

U.S. Family Members: 5

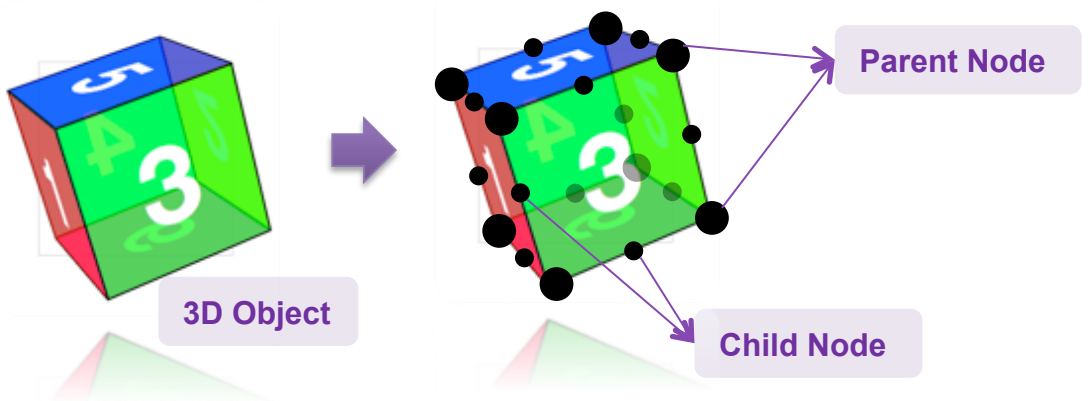
Non-U.S. Family Members: 0

DRAKES BAY COMPANY

Invention Overview

Subject patent describes the process of creating the animation data for 3D objects (Actors, Props, Cube etc.) in an optimized manner. The diagrams below illustrate the method proposed in the claimed invention.

Step 1: Modeling: Creation of nodes in 3D-Object



The 3D object, a cube, is divided into sub-parts such as the six faces of the cube. Each sub-part is made up of nodes (vertices & edges) which govern geometry of the object. There are two types of nodes –

- **Parent Node – All 8 vertices of cube (shown as big black dots).**
- **Child Node – 12 edges of the cube (represented by small black dots)**

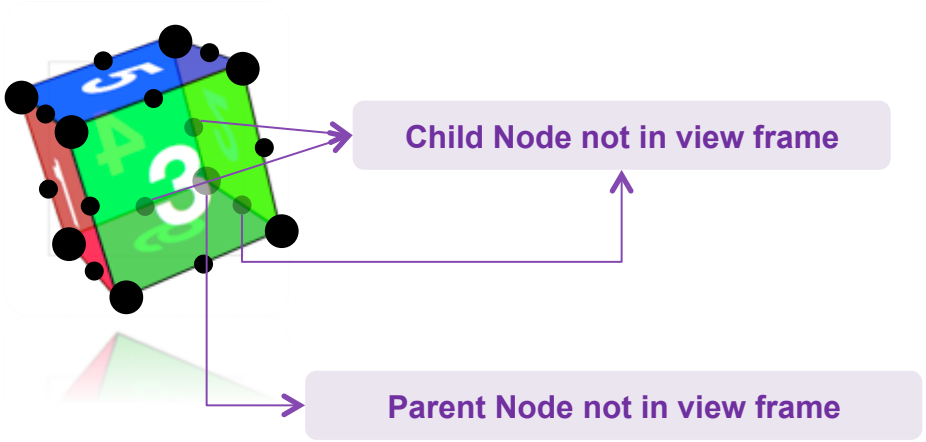
Animation data is applied to the parent node of 3D object and same animation data is inherited by respective child node.

***Note**: This section is provided by the analyst for better visualization of the claimed invention.*

Information on Next Slide

Invention Overview

Step 2: Determination and Analysis of View Frame



View frame is analyzed to determine nodes that are not visible on the screen of the viewer.

As evident from the image, face 6 of cube (on the bottom) and the nodes associated with it (vertex & edges) are not visible to viewer. All such nodes are determined as outside the view frame.

Information on Previous Slide

Information on Next Slide

DRAKES BAY COMPANY

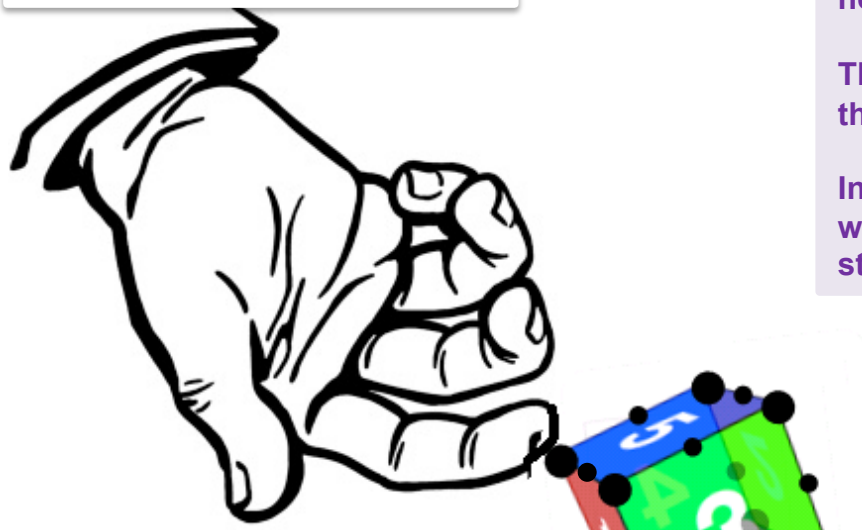
Invention Overview

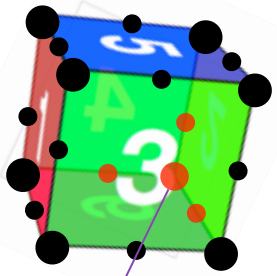
Step 3: Optimizing Animation

The method claimed removes the animation data of the parent node and connected nodes which are not visible on the screen.

This helps in optimization of the animation data associated with the nodes.

In the example below, animation data is removed from cube-face 6 which is not visible to the viewer. The same is described in the steps below.





Step 1.
3D object is characterized by nodes (both parent & child node) and animation data is applied therein.

Step 2:
View frame is analyzed to determine nodes that are outside the view frame such as face 6 on the bottom of the cube.

Step 3:
Animation data for determined nodes are removed.

Information on Previous Slide

CLAIM ELEMENTS

This section presents the shortlisted claim used for mapping. It also provides an overview on what information has been identified and to what extent. Below is the color-coding used for this purpose –

- **Green** text represents a sub-element, relevant information for which is directly and explicitly available in the product.
- Blue text represents additional important information related to a topic/subtopic added by the analyst team.

DRAKES BAY COMPANY

Elements and Sub Elements of Claim 1

Preamble:

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:

Clause 1:

determining if the node is outside a view frame;

Clause 2:

determining if any child node associated with the node is outside the view frame; and

Clause 3:

removing the animation data associated with the node if the node and any associated child node are outside the view frame.

PRODUCT OVERVIEW

This section provides an overview of the product mapped with the subject patent. This information is provided to aid understanding of the mapping by providing a glance at the product beforehand.

Purple text and boxes represent comments added by the analyst team to improve understanding.

DRAKES BAY COMPANY

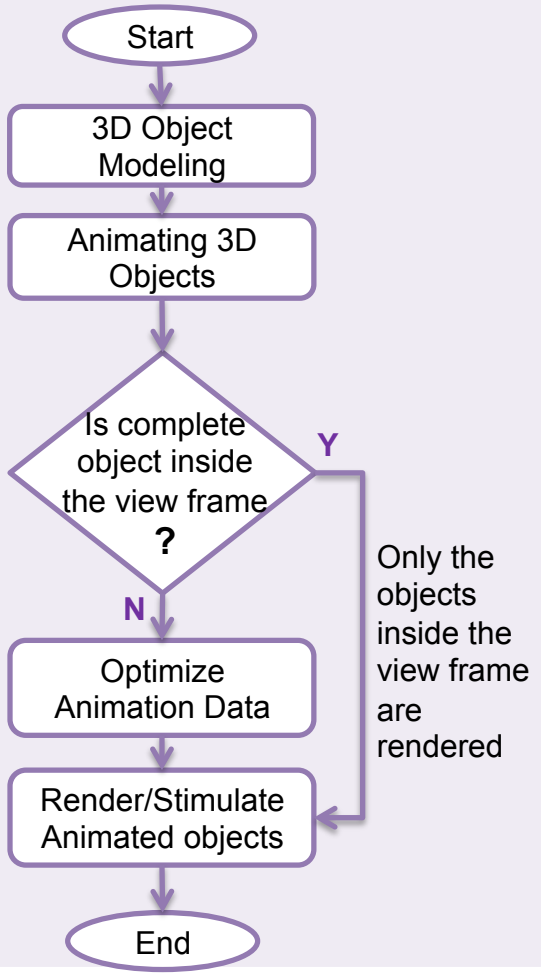
Company: Adobe
Product: Adobe Flash – Stage3D

Introduction

Stage3D is the new Flash API that Adobe recently released. It is dedicated to real-time 3D rendering. With Stage3D, you can take full advantage of the hardware accelerated capabilities of the user's computer GPU directly from Flash. Being able to use 3D acceleration in Flash opens up many possibilities for Flash games and Flash applications that were previously not possible.

Analyst Comment: Stage3D is an Adobe flash application programming interface (API) that uses hardware acceleration for 3D rendering instead of software mode that relies on the CPU for rendering. Stage3D is easier to use and offers faster rendering.

Diagram below depicts the workflow of the product. It is provided by analyst for illustrative purpose.



Source: www.adobe.com/devnet/flashplayer/articles/how-stage3d-works.html

DRAKES BAY COMPANY

Company: Adobe
Product: Adobe Flash - Stage3D

Market information:

The Adobe® Flash® Player runtime lets you effortlessly **reach more than 1 billion connected desktops across browsers and operating systems** with no additional software installation — 11 times more people than the best selling hardware console.

Use the Adobe AIR® runtime to package the same code into native apps for Windows, Mac, iPhone, iPad, and Android™ devices, reaching over two billion systems. Join a host of developers who have published more than 175,000 applications using AIR 3.8 or greater. **These applications were installed over 500 million times in the second half of 2013, with 300 million installations on Android and iOS alone!**

Source 1

With its 98%+ of market penetration, Flash Player is ubiquitous. The ubiquity of Flash Player and 3D hardware acceleration is a powerful combination that might transform the online gaming forever, similar to the rapid evolution of computer games after the introduction of 3D-accelerated hardware.

Source 2

Analyst Comment: Adobe Flash has over 98% market share and reaches more than 1 billion connected desktops worldwide.

Source 1: http://www.adobe.com/products/flashruntimes/statistics.html

Source 2: www.adobe.com/devnet/flashplayer/articles/how-stage3d-works.html

PRODUCT MAPPING

This section provides element-by-element mapping of the subject claim with the product identified. Below is the color-coding used for this purpose –

- **Green** text represents a sub-element for which relevant information is explicitly available.
- **Purple** text represents comments and notes added by the analyst team to improve understanding.
- **Blue box** represents the claim element that is being mapped with the product.

Note: To explicitly show presence of some claim elements, we have used references/ text which are not from the official website of Adobe.

Claim 1 (Preamble) (I)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:

If you've ever dreamed of writing your own console-style **3D game in Flash**, get ready to be blown away by the hardware accelerated power of Stage3D. *Source 1*

3D Animated content

Animated textures

There are many ways to code an animated texture and many situations where this is the perfect method to achieve an awesome effect. Imagine, for example, that you wanted to "project" a streaming video or cool-looking animation onto a surface in your game. This could be in a virtual movie theater, or for the screen of some futuristic video watch, smartphone or other special effect. *Source 2*

Computer System

With Stage3D, you can take full advantage of the **hardware accelerated capabilities of the user's computer GPU directly from Flash.** *Source 3*

Analyst Comment: Stage3D is a 3D animation Flash application used to create 3D animation content such as 3D games with animated textures. It runs on a computer system using a GPU (graphics processing unit) as a hardware accelerator.

Source 1: http://www.adobe.com/devnet/flashplayer/articles/stage3d-programmers-guide.html

Source 2: http://books.google.co.in/books?id=loJVExpIHcsC&pg=PT293&lpg=PT293&dq=Flash+11+stage+3D+movie&source=bl&ots=xSMPQpolVj&sig=kYT5hQIVvLeKYehQg1b2qY0iFYc&hl=en&sa=X&ei=w712VK7FAcThoATQ6YDgDw&ved=0CFMQ6AEwCA#v=onepage&q&f=false

Source 3: http://www.adobe.com/devnet/flashplayer/articles/how-stage3d-works.html

Claim 1 (Preamble) (II)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, **the 3D object having a hierarchy of parent nodes and children nodes**, each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:

*Description of **"the 3D object having a hierarchy of parent nodes and children nodes"** from the subject patent.*

> **Actors and props (collectively referred to as actors) are composed of a hierarchy of nodes**. The hierarchy begins with a root node and proceeds down to other nodes, **each node being associated with a discrete piece of 3D geometry (group of polygons) making up the 3D object. Each node is further identified by a node name. For example, a node representing an object's head might be named a "head" node.**
>
> **Each node has zero or more parent and child nodes, with the restriction that the linkages cannot form a loop. Thus, a trunk node may have a leg node as one of its children, and a head node as one of its parents.**

Analyst Comment: According to the subject patent, a 3D object (e.g. actors and props occurring in 3D content) is made up of various 3D geometries and each 3D geometry is associated with a unique node.
A node (e.g. trunk) may have other nodes connected to it called as parent and child nodes. For example, a parent node (e.g. head) and a child node (e.g. leg) are connected to the trunk node.

Source: http://patentimages.storage.googleapis.com/pdfs/US6538654.pdf (Column 5, Line 58)

Claim 1 (Preamble) (III)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, **the 3D object having a hierarchy of parent nodes and children nodes**, each node being associated with animation data, a method for optimizing the animation data associated with each node comprising:

Geometry forming parent and child nodes

Generally speaking, a 3D scene is defined as a group of 3D geometries (meshes). Each geometry is specified as a set of triangles, and each triangle, in turn, is comprised of a set of vertices. So, defining a 3D scene simply means to define a set of vertices, and eventually add some related rendering information—such as textures or vertex colors.

Analyst Comment: Similar to what described in subject patent (see slide 15), 3D objects in Adobe Flash are comprised of meshes (geometry) which are associated with the nodes.
Hence, here we have shown the presence of 3D geometry which indicates the existence of nodes in the product.

Source : http://www.adobe.com/devnet/flashplayer/articles/how-stage3d-works.html

Claim 1 (Preamble) (IV)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, **each node being associated with animation data**, a method for optimizing the animation data associated with each node comprising:

Animated textures

Animation data

There are many ways to code an animated texture and many situations where this is the perfect method to achieve an awesome effect. Imagine, for example, that you wanted to "project" a streaming video or cool-looking animation onto a surface in your game. This could be in a virtual movie theater, or for the screen of some futuristic video watch, smartphone or other special effect.

ɔurce 1

Generally speaking, **a 3D scene is defined as a group of 3D geometries (meshes).** Each geometry is specified as a set of triangles, and each triangle, in turn, is comprised of a set of vertices. So, defining a 3D scene simply means to define a set of vertices, and eventually add some related rendering information—such as textures or vertex colors.

Source 2

Analyst Comment: Stage3D can be used to put animation in various 3D scenes such as games. Since each 3D scene is made up of nodes (meshes), each node will have some animation data associated with it.

Source 1
http://books.google.co.in/books?id=loJVExpIHcsC&pg=PT293&lpg=PT293&dq=Flash+11+stage+3D+movie&source=bl&ots=xSMPQpoIVj&sig=kYT5hQIVvLeKYehQg1b2qY0iFYc&hl=en&sa=X&ei=w712VK7FAcThoATQ6YDgDw&ved=0CFMQ6AEwCA#v=onepage&q&f=false

Source 2: http://www.adobe.com/devnet/flashplayer/articles/how-stage3d-works.html

Claim 1 (Preamble) (V)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, each node being associated with animation data, **a method for optimizing the animation data associated with each node comprising:**

Description from the subject patent.

> **the animation and texture data of the 3D animation content is optimized to reduce the size of the animation data to be streamed over the Internet.** The system and method according to this second aspect of the invention gathers statistical information about the nodes and textures being utilized in each frame of the animated content. In doing so, **the system and method determines if a particular node and any of its children nodes are outside the view frame. If they are, then the animation data associated with the node is removed.**

Analyst Comment: According to the subject patent, optimization of animation data includes removing the data associated with a node which is outside the field of view.

Source: https://patentimages.storage.googleapis.com/pdfs/US6538654.pdf (Column 3, Line 17)

Claim 1 (Preamble) (VI)

In a computer system for creating animation data for a 3D object appearing in a 3D animated content, the 3D object having a hierarchy of parent nodes and children nodes, each node being associated with animation data, **a method for optimizing the animation data associated with each node comprising:**

Optimizing the animation data

Backface culling

Backface culling is the process in which meshes are rendered with only one side of each polygon being visible. This helps to speed up rendering as in, for example, a cube with opaque sides you would never be able to see the inside walls. When backface culling is turned on, which is the default, these inside faces (the backfaces) are never rendered. Generally, this works great but there are times when you want both sides of every polygon to be drawn.

Analyst Comment: According to the product description, optimization of animation data includes rendering only the data which is visible i.e. removing the data associated with a node (mesh) which is outside the field of view.

Source:
http://books.google.co.in/books?id=loJVExplHcsC&pg=PT293&lpg=PT293&dq=Flash+11+stage+3D+movie&source=bl&ots=xSMPQpolVj&sig=kYT5hQIVvLeKYehQg1b2qY0iFYc&hl=en&sa=X&ei=w712VK7FAcThoATQ6YDgDw&ved=0CFMQ6AEwCA#v=onepage&q&f=false

Claim 1 (Clause 1) (l)

determining if the node is outside a **view frame**;

*Description of "**view frame**" from the subject patent.*

> The animation inaccuracy of the child node does not affect the quality of the 3D movie if the child node is **not visible (i.e. outside the view frame).**

Analyst Comment: According to the subject patent, the concept of a view frame refers to the field of view in which the node is visible and is not obscured by the other nodes.

Source: http://patentimages.storage.googleapis.com/pdfs/US6538654.pdf (Column 65, Line 52)

determining if the node is outside a view frame;

Backface culling

Backface culling is the process in which meshes are rendered with only one side of each polygon being visible. This helps to speed up rendering as in, for example, a cube with opaque sides you would never be able to see the inside walls. When backface culling is turned on, which is the default, these inside faces (the backfaces) are never rendered. Generally, this works great but there are times when you want both sides of every polygon to be drawn.

Analyst Comment: The software culls the parts (nodes) of the objects outside the view frame. To do this, it is necessary to determine if the nodes are outside the view frame.

Source:
http://books.google.co.in/books?id=loJVExplHcsC&pg=PT293&lpg=PT293&dq=Flash+11+stage+3D+movie&source=bl&ots=xSMPQpolVj&sig=kYT5hQIVvLeKYehQg1b2qY0iFYc&hl=en&sa=X&ei=w712VK7FAcThoATQ6YDgDw&ved=0CFMQ6AEwCA#v=onepage&q&f=false

determining if any child node associated with the node is outside the view frame; and

Description from the subject patent.

Thus, removal of a parent node's animation data negatively affects the accuracy of the animation of a child node. **The animation inaccuracy of the child node does not affect the quality of the 3D movie if the child node is not visible (i.e. outside the view frame). However, if the child node is within the view frame, the inaccurate animation of the child node deteriorates the overall quality of the 3D movie.**

Analyst Comment: According to the subject patent, the animation of a child node affects the quality of the complete 3D movie. Hence it becomes important to determine if the child node lies outside the view frame.

Source: https://patentimages.storage.googleapis.com/pdfs/US6538654.pdf (Column 15, Line 60)

determining if any child node associated with the node is outside the view frame; and

Backface culling

Backface culling is the process in which meshes are rendered with only one side of each polygon being visible. This helps to speed up rendering as in, for example, a cube with opaque sides you would never be able to see the inside walls. When backface culling is turned on, which is the default, these inside faces (the backfaces) are never rendered. Generally, this works great but there are times when you want both sides of every polygon to be drawn.

Analyst Comment: In Stage3D, backface culling takes place for the nodes present outside the view frame. Determining if child nodes exist outside the view frame is a necessary part of this process.

Source:
http://books.google.co.in/books?id=loJVExplHcsC&pg=PT293&lpg=PT293&dq=Flash+11+stage+3D+movie&source=bl&ots=xSMPQpolVj&sig=kYT5hQIVvLeKYehQg1b2qY0iFYc&hl=en&sa=X&ei=w712VK7FAcThoATQ6YDgDw&ved=0CFMQ6AEwCA#v=onepage&q&f=false

removing the animation data associated with the node if the node and any associated child node are outside the view frame.

Backface culling

Backface culling is the process in which meshes are rendered with only one side of each polygon being visible. This helps to speed up rendering as in, for example, a cube with opaque sides you would never be able to see the inside walls. When backface culling is turned on, which is the default, these inside faces (the backfaces) are never rendered. Generally, this works great but there are times when you want both sides of every polygon to be drawn.

Removing the animation data

Analyst Comment: Inside faces (nodes or child nodes) of the 3D objects are removed if they are determined to be outside the view frame. Removing the nodes means the animation data is removed and therefore never rendered.

Source:
http://books.google.co.in/books?id=loJVExplHcsC&pg=PT293&lpg=PT293&dq=Flash+11+stage+3D+movie&source=bl&ots=xSMPQpolVj&sig=kYT5hQIVvLeKYehQg1b2qY0iFYc&hl=en&sa=X&ei=w712VK7FAcThoATQ6YDgDw&ved=0CFMQ6AEwCA#v=onepage&q&f=false

For additional information, please contact:

- Joseph W. Jennings
  - 1-415-927-2716
  - jjennings@drakesbaycompany.com

- Marisa Bracoloni
  - 1-415-927-2716
  - mbracoloni@drakesbaycompany.com