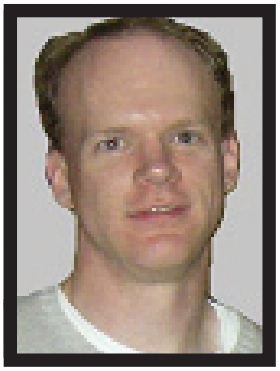
Create a Tag Cloud

2008

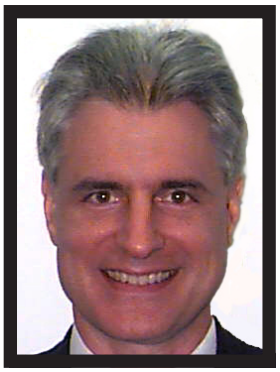
An Oracle "How To" Paper

Presented By:

Jeremy Blankenship



William Indest



ORACLE | SIEBEL

Introduction

Computers are useless. They can only give you answers.

Pablo Picasso

A **tag cloud** (or **weighted list** in visual design) is a visual depiction of user-generated tags used to describe the content of web sites. Tags are usually single words and are typically listed alphabetically, and the importance of a tag is shown with font size or color. Thus finding a tag alphabetically and by popularity is possible. In some cases the tags are hyperlinks that lead to a collection of items that are associated with the tag. Here is an example of a tag cloud from Wikipedia's entry on tag clouds:

In this paper we show you how to create a tag cloud using a field from Oracle's Customer Relationship Management application. We focus on the Public Sector application's PUB Offense business component and show how to create a tag cloud from the information stored in the field called Offense Type.

It's true that we don't know what we've got until we lose it, but it's also true that we don't know what we've been missing until it arrives

Anon Ymous

How It Works

Here's what the prototype looks like:

The html source that generated this image may be found in the Appendix.

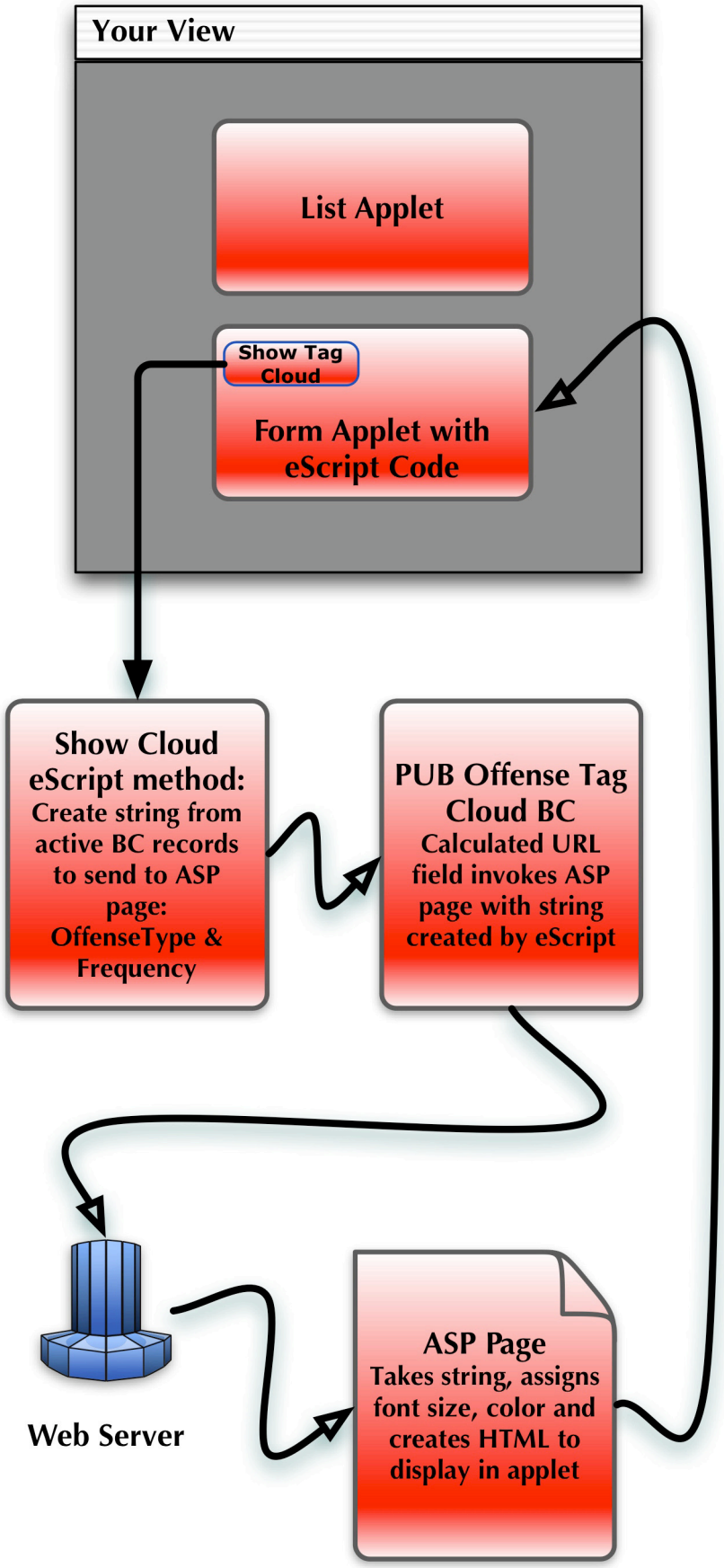
We created a new view with a list applet on top and a form applet on the bottom to display an HTML page. The Business Component is called JPS Offense Tag Cloud based on the class, CSSBCVExternalUrl. This BC doesn't do a lot. It has a calculated field for the URL, "<IFRAME SRC="" + GetProfileAttr("JPSTagURL") + "" height='700' width='800'></IFRAME>", that uses a profile attribute, **JPSTagURL**, that is created from the applet's eScript. The JPSTagURL is a string that invokes an ASP page. The ASP page is passed the list of Offense Types and frequencies calculated and normalized on a scale from 0 to 100. It has an Offense Id field that is used only for the link. The link is between PUB Offense and JPS Offense Tag Cloud; and, we added the JPS Offense Tag Cloud BC to the Business Object.

The new applet, JPS Offense Tag Cloud Applet should excite you; this is where all the action happens. The web page with the tag field, URL, on the Business Component is displayed here and the button triggers the generation of the Profile Attribute that the BC uses to display the URL.

Simplified

Click the button, invoke Show Cloud eScript, iterate over all active records creating a string consisting of Offense Type: Frequency (the ":" is a token for the ASP page to get the Offense Type as an HTML tag). The eScript sets the Profile Attribute and queries the Business Component which then makes a request via the URL calculated field which invokes the ASP page, passing the Offense Type: Frequency string as an argument to the ASP page. The ASP page chews on the string, assigns font sizes and color to the "Offense Type" tags based on frequency and creates an HTML page which is sent back to the form applet for display.

Here is an overview of the process:



An overview of the process flow.

Go For It

We created data to show the effectiveness of a tag cloud for this configuration effort. You will have to do the same with your own data. The prototype page was how we got a feel for what the data would look like on a scale from zero to one hundred with color changes.

Here is the eScript associated with the Tag Cloud applet:

eScript

```
function WebApplet_PreInvokeMethod (MethodName)
{
if (MethodName == "ShowMap")
{
   var sURL;
   var counter = 1;
   var arrOffenseType = new Array();
   var bcOffense = this.BusComp().ParentBusComp();
   with (bcOffense)
   {
     var isRecord = FirstRecord();

     var MaxOffenses = 1;

     if (isRecord)
     {
       sURL = "http://"+ Clib.getenv("COMPUTERNAME") +
".oracleads.com/tagcloud/showtagcloud.asp?";
       while (isRecord)
       {
         var sOffenseName = GetFieldValue("Offense Name");

         if (arrOffenseType[sOffenseName])
         {
            arrOffenseType[sOffenseName] += 1;
            if (ToInteger(arrOffenseType[sOffenseName]) >
MaxOffenses){
              MaxOffenses = ToInteger(arrOffenseType[sOffenseNam
e]);
            }
          } else  {
            arrOffenseType[sOffenseName] = 1;
          }

          counter += 1;

          isRecord = NextRecord();
        }
            FirstRecord();
      }
     }
     if (counter == 1)
     {
        sURL = "http://"+ Clib.getenv("COMPUTERNAME") +
".oracleads.com/tagcloud/blank.htm";
     } else {
        counter = 1;
        var OffenseType;
        for (OffenseType in arrOffenseType)
        {
           sURL += "OffenseType" + counter + "=" + OffenseType +
":" + ToInteger((arrOffenseType[OffenseType]/MaxOffenses) * 100)
+ "&";
           counter += 1;
```

```
            }
        }

        TheApplication().SetProfileAttr("JPSTagURL",sURL);
        this.BusComp().ExecuteQuery();
        return (CancelOperation);
        bcOffense = null;
        arrOffenseType = null;
      }

       return (ContinueOperation);
}
```

This is placed on the PreCanInvokeMethod of the applet:

```
function WebApplet_PreCanInvokeMethod (MethodName, &CanInvoke)
{
    if (MethodName == "ShowMap")
    {
        CanInvoke = "TRUE";
        return (CancelOperation);
    }
    return (ContinueOperation);
}
```

This code is executed when the view is visited:

```
function WebApplet_Load ()
{
    TheApplication().SetProfileAttr("JPSTagURL","http://"+ Clib.
getenv("COMPUTERNAME") + ".oracleads.com/tagcloud/blank.htm");
}
```

Here is the ASP source. This file, `showtagcloud.asp`, and the file, `blank.htm` file should be placed in the directory: `C:\Inetpub\wwwroot\tagcloud.`

ASP Source

```
<html>
<head>
    <title>Tag Cloud</title>


<%
  Dim offenseArray(100)
  Dim offensename


  for i=1 to 100
     offenseArray(i) = Request.QueryString("OffenseType" & i)
  next


%>
<style type="text/css">
a {text-decoration: none}
a {color:#a05a2c}
a {padding: 10px}
p {line-height: 200%}
body {background-color: #00222b}
/* body {font-family: Georgia} */
body {font-family: Arial}
</style>
```

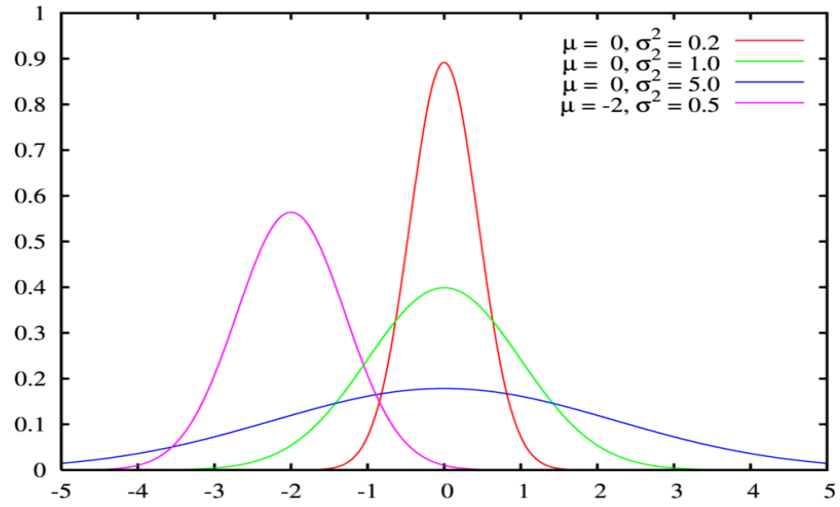
```
<script>
function processCloud(id,max) {
    var cloud = document.getElementById(id);

    if(!cloud) return;
        var min = 0.0;
        var range = max - min;
        var x = range/3.0;
        var small = x + min;   // one-third of the way; 33%
        var medium= small + x; // two-thirds of the way: 66%

    var tags = cloud.getElementsByTagName("a");
    for(var i=0;i<tags.length;i++) {
        var tag = tags[i];
        var title = tag.getAttribute('title');
        var f = title.substring(title.indexOf(":")+1);
        var fontSize = (125.0*(1.0+(1.5*f-max/2)/max))+"%";
        tag.style.fontSize = fontSize;
                if(f > small && f < medium){
            tag.style.color = "#d45500";
                } else if(f > medium) {
            tag.style.color = "#f60";
                }
    }
}
</script>
</head>

<body onload="processCloud('cloud', 100);">

    <div id='cloud'>
<%for i = 1 to 100
    if OffenseArray(i) <> "" then
        OffenseName = Left(OffenseArray(i),Instr(OffenseArray(i)
,":")-1) %>
        <a href='#' title='<%= OffenseArray(i) %>'><%=
OffenseName %></a>
    <% end if
next %>

</div>
</body>
</html>
```

There are lies, damned lies and statistics.

Mark Twain

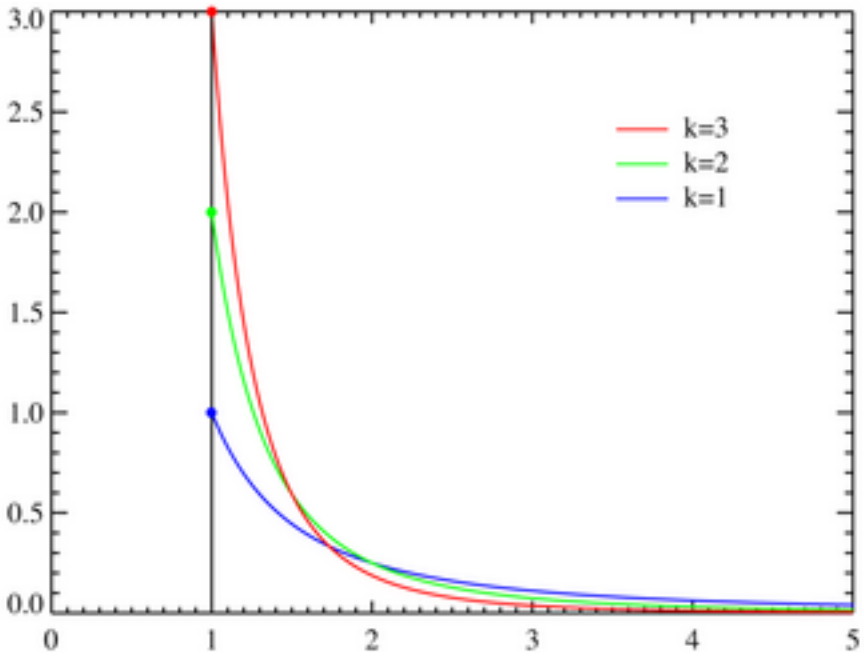
Danger: Statistics Ahead!

The spread of your data can be categorized as a distribution. Two fabulous distributions are the **normal distribution** or the "bell curve" and the **Pareto distribution**. A normal distribution with varying standard deviations is shown below.



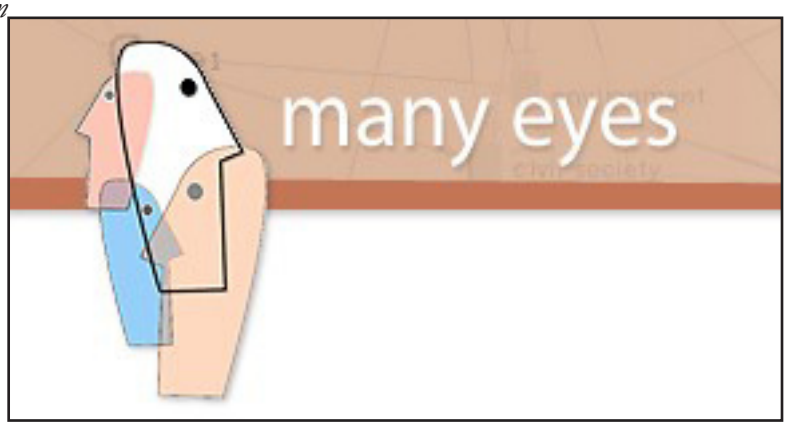
The **Pareto distribution**, named after the Italian economist Vilfredo Pareto, is a power law probability distribution that coincides with social, scientific, geophysical, actuarial, and many other types of observable phenomena. Outside the field of economics it is at times referred to as the **Bradford distribution**.

Pareto originally used this distribution to describe the allocation of wealth among individuals since it seemed to show rather well the way that a larger portion of the wealth of any society is owned by a smaller percentage of the people in that society. This idea is sometimes expressed more simply as the Pareto principle or the "80-20 rule" which says that 20% of the population owns 80% of the wealth. (This distribution is also related to the Long Tail musings of Wired's Chris Anderson.) The Pareto distribution is shown below; k is called the Pareto index. Don't worry about it.



To get a more evenly distributed range of font sizes in the tag cloud, it is necessary to "linearize" the original values. Understanding your underlying distribution helps you to linearize more effectively. You get a better result visually when you use a linearized representation for a tag cloud but you sacrifice some accuracy as you are turning a curved line into a straight line. There are many algorithms that are available on the web for linearizing a normal and a Pareto distribution. Here we assume the data was linear and simply transformed it to a zero to one hundred scale.

If you want to play with Tag Clouds some more may we suggest visiting *many-eyes.com*



There are several tricks that we used here. One is the creation of the array, arrOffenseType and using the value of the field, Offense Type, as an index.

Another trick is to loop over the array elements using a for loop: for (OffenseType in arrOffenseType). This is powerful stuff. Don't get too close.

Another trick is setting and getting the user specified Profile Attribute via:

```
TheApplication().SetProfileAttr("JPSTagURL",sURL);
```

This helps communicate what is going on in one part of the application with another part. While global variables are a no-no there are times when the constraints of an application dictate the usage of a global variable to get it done. This is a good example of doing the right thing in the wrong way.

While not a trick but a thoughtful design consideration is the creation of a generic ASP page for you to use independent of the business component for which you choose to create a tag cloud. Even though the array name is called OffenseArray if you pass the ASP a string with the pattern: field: frequency, you can reuse the ASP page for creating your tag clouds.

Here is html source that generated the image found in the "How It Works" section:

```
<html>
<head>
<style type="text/css">
a {text-decoration: none}
a {color:#a05a2c}
a {padding: 10px}
p {line-height: 200%}
body {background-color: #00222b}
/* body {font-family: Georgia} */
body {font-family: Arial}
</style>
<script>
function processCloud(id,max) {
       var cloud = document.getElementById(id);
       if(!cloud) return;
        var min = 0.0;
        var range = max - min;
        var x = range/3.0;
        var small = x + min;
        // one-third of the way; 33%
        var medium= small + x;
        //two-thirds of the way: 66%

       var tags = cloud.getElementsByTagName("a");
       for(var i=0;i<tags.length;i++) {
              var tag = tags[i];
              var title = tag.getAttribute('title');
              var f = title.substring(title.indexOf(":")+1);
              var fontSize = (125.0*(1.0+(1.5*f-max/2)/max))+"%";
              tag.style.fontSize = fontSize;
                 if(f > small && f < medium){
                     tag.style.color = "#d45500";
                 } else if(f > medium) {
                     tag.style.color = "#f60";
                 }
       }
}
</script>
</head>

<body onload="processCloud('cloud', 100);">

       <div id='cloud'>
<a href='#' title='Arson:10'>Arson</a>
<a href='#' title='Aggravated Assault:77'>Aggravated</a>
<a href='#' title='Simple Assault:83'>Simple</a>
<a href='#' title='Intimidation:12'>Intimidation</a>
<a href='#' title='Bribery:11'>Bribery</a>
<a href='#' title='Burglary/Breaking and Enteringg:82'>Burglary</a>
<a href='#' title='Counterfeiting/Forgery:90'>Counterfeiting</a>
<a href='#' title='Destruction/Damage/Vandalism of Property:80'>Property
Damage</a>
<a href='#' title='Drug/Narcotic Violations:100'>Drug/Narcotic</a>
<a href='#' title='Drug Equipment Violations:90'>Drug Equipment</a>
<a href='#' title='Embezzlement:44'>Embezzlement</a>
<a href='#' title='Extortion/Blackmail:20'>Extortion/Blackmail</a>
<a href='#' title='False Pretenses/Swindle/Confidence Game:30'>Swindle/
Confidence Game</a>
<a href='#' title='Credit Card/Automatic Teller Machine Fraud:80'>Credit
Card Fraud</a>
<a href='#' title='Impersonation:2'>Impersonation</a>
<a href='#' title='Welfare Fraud:12'>Welfare Fraud</a>
<a href='#' title='Wire Fraud:19'>Wire Fraud</a>
<a href='#' title='Betting/Wagering:27'>Betting/Wagering</a>
```

```
<a href='#' title='Operating/Promoting/Assisting Gambling:33'>Gambling</a>
<a href='#' title='Gambling Equipment Violations:56'>Gambling Equipment</a>
<a href='#' title='Sports Tampering:34'>Sports Tampering</a>
<a href='#' title='Murder and Nonnegligent Manslaughter:8'>Murder</a>
<a href='#' title='Negligent Manslaughter:19'>Negligent Manslaughter</a>
<a href='#' title='Justifiable Homicide:11'>Justifiable Homicide</a>
<a href='#' title='Kidnapping/Abduction:10'>Kidnapping/Abduction</a>
<a href='#' title='Pocket-picking:1'>Pocket-picking</a>
<a href='#' title='Purse-snatching:0'>Purse-snatching</a>
<a href='#' title='Shoplifting:70'>Shoplifting</a>
<a href='#' title='Theft from Building:10'>Theft from Building</a>
<a href='#' title='Theft from Coin-Operated Machine or Device:10'>Theft
from Machine</a>
<a href='#' title='Theft from Motor Vehicle:76'>Theft from Motor Vehicle</
a>
<a href='#' title='Theft of Motor Vehicle Parts or Accessories:100'>Theft
of Motor Vehicle Parts</a>
<a href='#' title='All Other Larceny:10'>All Other Larceny</a>
<a href='#' title='Motor Vehicle Theft:82'>Motor Vehicle Theft</a>
<a href='#' title='Pornography/Obscene Material:60'>Pornography</a>
<a href='#' title='Prostitution:21'>Prostitution</a>
<a href='#' title='Assisting or Promoting Prostitution:100'>Promoting
Prostitution</a>
<a href='#' title='Robbery:72'>Robbery</a>
<a href='#' title='Forcible Rape:14'>Forcible Rape</a>
<a href='#' title='Forcible Sodomy:1'>Forcible Sodomy</a>
<a href='#' title='Sexual Assault With an Object:43'>Sexual Assault With an
Object</a>
<a href='#' title='Forcible Fondling:11'>Forcible Fondling</a>
<a href='#' title='Incest:0'>Incest</a>
<a href='#' title='Statutory Rape:20'>Statutory Rape</a>
<a href='#' title='Stolen Property Offenses :93'>Stolen Property Offenses
</a>
<a href='#' title='Weapon Law Violations:10'>Weapon Law Violations</a>
<a href='#' title='Bad Checks:8'>Bad Checks</a>
<a href='#' title='Curfew/Loitering/Vagrancy Violations:39'>Curfew/
Loitering/Vagrancy Violations</a>
<a href='#' title='Disorderly Conduct:60'>Disorderly Conduct</a>
<a href='#' title='Driving Under the Influence:89'>DUI</a>
<a href='#' title='Drunkenness:10'>Drunkenness</a>
<a href='#' title='Family Offenses, Nonviolent:18'>Family Offenses,
Nonviolent</a>
<a href='#' title='Liquor Law Violations:1'>Liquor Law Violations</a>
<a href='#' title='Peeping Tom:1'>Peeping Tom</a>
<a href='#' title='Runaway:1'>Runaway</a>
<a href='#' title='Trespass of Real Property:22'>Trespass of Real
Property</a>
<a href='#' title='All Other Offenses:16'>Other</a>

</div>
</body>
</html>
```

EOF

Title: Creating Tag Clouds
Authors: Jeremy Blankenship and William Indest
Managers: Monica Morgan and James Perry

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com