```
var msgrControl;
var mgrsText;
var mouseFormat;
var mouseDisplayBtn;
var mgrsCtrlAction;
var pointCtrlAction;
var llpointCtrl;

 function addMGRSControl(){
      var toolbar2;

      mgrsCtrl = new OpenLayers.Control.DrawFeature(drawLayer,
OpenLayers.Handler.Point, {featureAdded: showAllCoords,
            handlerOptions: {layerOptions: {  styleMap: new OpenLayers.StyleMap({
            'default': new OpenLayers.Style({
                 fillColor: "#FFEC8B",
                 fillOpacity: 0.3,
                 stroke: true,
                 cursor: "crosshair",
                 strokeColor: "#000000",
                 strokeWidth: 2,
                 pointRadius: 15
              })
           })}}});

      mgrsCtrlAction = new GeoExt.Action({control: mgrsCtrl,

iconCls: 'getPoint',

tooltip: 'Get Coordinates at Point Click',

toggleGroup: 'aoi',
                                                                                id:
'mgrsCtrlBtn',
                                                                                map:
map});



     toolbar2 = Ext.getCmp("toolbar2");
     toolbar2.add(new Ext.Toolbar.Separator());
     toolbar2.add(mgrsCtrlAction);


      //toolbar.addControl(mgrsCtrl, {id: 'mgrsCtrlBtn', iconCls: 'geolocate',
tooltip: 'Get MGRS Coordinates at Point Click',  toggleGroup: 'map'});

 }

 function showAllCoords(){

           feat = drawLayer.features[0];

           var geom = feat.geometry;
           //console.log(distGeom);
           geom.transform( map.getProjectionObject(), new
OpenLayers.Projection("EPSG:4326"));
           //alert(geom.toShortString());


           var geom = feat.geometry;
           //console.log(distGeom);
```

```
            //alert(geom.toShortString());
            var mgrsText = LLtoMGRS(geom.y, geom.x, 5)
                            Ext.Ajax.request({ url:
gSchema+".grid_Functions.getCoord?srid="+srid+"&p_wkt="+geom+"&ddFormat="+mgrsText,
                                 success: showAllCoordSuccess,
                                 method: "POST",
                                 disableCaching: false});

 }

 function showAllCoordSuccess(response) {

          Ext.Msg.prompt("Point Coordinates", "Coordinates at click point (X,Y)",""
,"" ,  80,  response.responseText);
       }



  String.prototype.rpad = function(padString, length) {
        var str = this;
    while (str.length < length)
        str = str + padString;
    return str;
}


  function convertDMS (coordinate, type) {
    var coords = new Array();

    abscoordinate = Math.abs(coordinate)
    coordinatedegrees = Math.floor(abscoordinate);

    coordinateminutes = (abscoordinate - coordinatedegrees)/(1/60);
    tempcoordinateminutes = coordinateminutes;
    coordinateminutes = Math.floor(coordinateminutes);
    coordinateseconds = (tempcoordinateminutes - coordinateminutes)/(1/60);
    coordinateseconds =  Math.round(coordinateseconds*100);
    coordinateseconds /= 100;

    if( coordinatedegrees < 10 )
      coordinatedegrees = "0" + coordinatedegrees;

    if( coordinateminutes < 10 )
      coordinateminutes = "0" + coordinateminutes;

    if( coordinateseconds < 10 )
      coordinateseconds = "0" + coordinateseconds;

    coordinateseconds+="";

    coords[0] = coordinatedegrees;
    coords[1] = coordinateminutes;
    coords[2] = coordinateseconds.rpad("0",5);
    coords[3] = this.getHemi(coordinate, type);

    return coords.join("");
  }

  /**
 *    * Return the hemisphere abbreviation for this coordinate.
 *       */
 function getHemi (coordinate, type) {
    var coordinatehemi = "";
```

```
    if (type == 'LAT') {
      if (coordinate >= 0) {
        coordinatehemi = "N";
      }
      else {
        coordinatehemi = "S";
      }
    }
    else if (type == 'LON') {
      if (coordinate >= 0) {
        coordinatehemi = "E";
      } else {
        coordinatehemi = "W";
      }
    }

    return coordinatehemi;
  }

/* function setMouseLLDisplay(){

   //if (pointCtrl) pointCtrl.featureAdded =  showDMSCoords;

   if (mouseControl.displayProjection=="EPSG:4326"){
          if (mouseDisplayBtn.pressed) {
             mouseControl.formatOutput =  function(lonLat) {
           var markup = convertDMS(lonLat.lat, "LAT");
           markup += " " + convertDMS(lonLat.lon, "LON");
           return markup
          }

          }else{
             mouseControl.formatOutput =function (lonLat){
                var digits=parseInt(this.numDigits);var newHtml=this.prefix+
                        lonLat.lon.toFixed(digits)+
                        this.separator+
                        lonLat.lat.toFixed(digits)+
                        this.suffix;return newHtml;}
          }
   }
  // console.log("changing mouseDisplay");

 }
*/

function googleEarthLayer(layerName, fullname, maptype, kmz){
     alert ("Hello");

 }


function loadO3D() {
        Ext.DomHelper.append(document.body, [{ id: "mgrsDiv"}]);
        mgrsText = new Ext.Toolbar.TextItem({contentEl: 'mgrsDiv', cls:
'x-status-text-panel', tooltip: "MGRS Coordinates of the Mouse" });

        mgrsControl = new OpenLayers.Control.GridMousePosition({ element: mgrsDiv
});
        map.addControl (mgrsControl);
        addMGRSControl();
       // addDMSDisplay();
        Ext.getCmp("srFrame").hide();
        Ext.getCmp("trafficBtn").hide();
```

```
        Ext.getCmp("svBtn").hide();
        Ext.getCmp("mapcenter").bottomToolbar.items.itemAt(2).hide();
        Ext.getCmp("mapcenter").bottomToolbar.items.itemAt(3).hide();
        Ext.getCmp("mapcenter").bottomToolbar.items.itemAt(4).hide();
        Ext.getCmp("mapcenter").bottomToolbar.items.insert(20,mgrsText);

        Ext.getCmp("cm_viewport").on("add", function() {
        if (Ext.getCmp("mapcenter").topToolbar.items.getCount()==30)
Ext.getCmp("mapcenter").topToolbar.items.itemAt(21).hide() ;
   });

        Ext.getCmp("cm_viewport").on("add", function() {
        if (Ext.getCmp("mapcenter").topToolbar.items.getCount()==30) {
             Ext.getCmp("mapcenter").topToolbar.items.itemAt(21).hide() ;
             Ext.getCmp("mapcenter").topToolbar.items.itemAt(15).hide() ;
             Ext.getCmp("mapcenter").topToolbar.items.itemAt(16).hide() ;
             Ext.getCmp("mapcenter").topToolbar.items.itemAt(17).hide() ;
        }
   });


   Ext.getCmp("cm_viewport").on("add", function() {
      if (Ext.getCmp("mapcenter").topToolbar.items.getCount()==30)
Ext.getCmp("mapcenter").topToolbar.items.itemAt(22).hide() ;
   });

    Ext.getCmp("cm_viewport").on("add", function() {
        if (Ext.getCmp("mapcenter").topToolbar.items.getCount()==30)
Ext.getCmp("mapcenter").topToolbar.items.itemAt(19).hide() ;
     });
     Ext.getCmp("cm_viewport").on("add", function() {
        if (Ext.getCmp("mapcenter").topToolbar.items.getCount()==30)
Ext.getCmp("mapcenter").topToolbar.items.itemAt(20).hide() ;
     });
/*    Ext.getCmp("cm_viewport").on("add", function() {
      if (pointCtrl) pointCtrl.featureAdded =  showDMSCoords;
    });

  /* cp.on('statechange', setMouseLLDisplay, this, {
    single: true,
    delay: 1000
   }); */


   af = new Ext.Toolbar.Fill();
   toolbar2.add(af);



  var clearSession = new Ext.Toolbar.Button({
                    text: "Reset",
                    tooltip: 'Reset Session',
                    handler:  function(){
                              fullZoom();
                            toggleChecks(tree,false);
                            collapseTree();
                        }
                 });

   toolbar2.add(clearSession);

    /*Ext.getCmp("cm_viewport").on("add", function() {
        if (Ext.getCmp("layerTree_").topToolbar.items.getCount()==5)
```

```
Ext.getCmp("layerTree_").topToolbar.items.itemAt(4).hide() ;
          //console.log(Ext.getCmp("layerTree_").topToolbar.items.getCount());
     });*/
        Ext.getCmp("cm_viewport").doLayout();
}
```