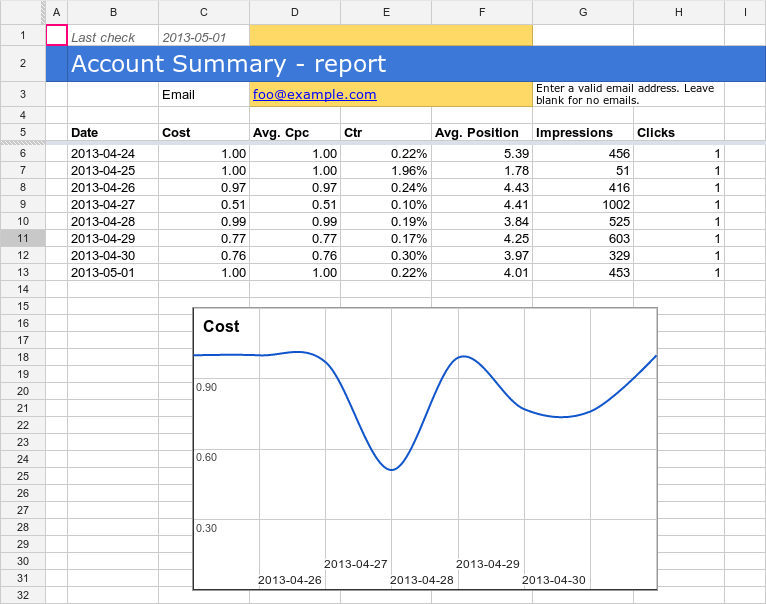
Account Summary Report



Account Summary Report is an at-a-glance report showing the performance of an entire AdWords account. Additionally, it sends an HTML-formatted daily email with current account statistics. An account manager could use a similar technique to email professionally-formatted reports directly to clients.



Scheduling

The script relies on previous day's statistics. Schedule it **Daily, 3am** or later to guarantee accuracy, since AdWords statistics may be [up to 3 hours delayed](https://support.google.com/adwords/answer/2544985).

How it works

The script fetches yesterday's statistics and updates the spreadsheet. The graphs are pre-configured to display the stats. If the email address is specified, the script sends out an HTML-formatted email.

**Tip:** if you set "Last check" date in the spreadsheet to a date in the past, the script will fill in the data between then and now. We recommend that you do not go back more than 90 days, as fetching statistics takes time.

Setup

* Setup a spreadsheet-based script with the source code below. Use Account Summary Report [template spreadsheet](http://goo.gl/yFkbH6)
* Create a new script with the source code below.
* Don't forget to update SPREADSHEET\_URL in code.
* Schedule the script **Daily, 5am**.

[Setting up a spreadsheet-based script](https://developers.google.com/adwords/scripts/docs/solutions/account-summary)

Source code

var SPREADSHEET\_URL = "[YOUR\_URL]";  
  
function main() {  
  var spreadsheet = SpreadsheetApp.openByUrl(SPREADSHEET\_URL);  
  spreadsheet.getRangeByName("account\_id\_report").setValue(AdWordsApp.currentAccount().getCustomerId());    
    
  var yesterday = getYesterday();  
  var last\_check = spreadsheet.getRangeByName("last\_check").getValue();  
  // Checks every day from last day checked to yesterday inclusive. If there isn't a last date checked,  
  // checks yesterday.  
  var date;  
  if (last\_check.length == 0) {  
    date = new Date(yesterday);  
  } else {  
    date = new Date(last\_check);   
    date.setDate(date.getDate() + 1);  
  }  
    
  var rows = [];  
    
  while (date.getTime() <= yesterday.getTime()) {  
    var row = getReportRowForDate(date);  
    rows.push([new Date(date), row['Cost'], row['AverageCpc'], row['Ctr'], row['AveragePosition'], row['Impressions'], row['Clicks']]);  
    date.setDate(date.getDate() + 1);  
  }  
    
  if (rows.length > 0) {  
    var access = new SpreadsheetAccess(SPREADSHEET\_URL, "Report");  
    var emptyRow = access.findEmptyRow(6, 2);  
    if (emptyRow < 0) {  
      access.addRows(rows.length);  
      emptyRow = access.findEmptyRow(6, 2);  
    }  
    access.writeRows(rows, emptyRow, 2);  
      
    var last\_check = spreadsheet.getRangeByName("last\_check").setValue(yesterday);  
    var email = spreadsheet.getRangeByName("email").getValue();  
    
    if (email) {  
      sendEmail(email);  
    }  
  }  
}  
  
function sendEmail(email) {  
  var day = getYesterday();  
  var yesterdayRow = getReportRowForDate(day);  
  day.setDate(day.getDate() - 1);  
  var twoDaysAgoRow = getReportRowForDate(day);  
  day.setDate(day.getDate() - 5);  
  var weekAgoRow = getReportRowForDate(day);  
    
  var html = [];  
  html.push(  
    "<html>",  
      "<body>",   
        "<table width=800 cellpadding=0 border=0 cellspacing=0>",   
          "<tr>",   
            "<td colspan=2 align=right>",   
              "<div style='font: italic normal 10pt Times New Roman, serif; margin: 0; color: #666; padding-right: 5px;'>Powered by AdWords Scripts</div>",   
            "</td>",   
          "</tr>",   
          "<tr bgcolor='#3c78d8'>",   
            "<td width=500>",   
              "<div style='font: normal 18pt verdana, sans-serif; padding: 3px 10px; color: white'>Account Summary report</div>",   
            "</td>",   
            "<td align=right>",   
              "<div style='font: normal 18pt verdana, sans-serif; padding: 3px 10px; color: white'>", AdWordsApp.currentAccount().getCustomerId(), "</h1>",   
            "</td>",   
            "</tr>",   
          "</table>",   
          "<table width=800 cellpadding=0 border=0 cellspacing=0>",   
            "<tr bgcolor='#ddd'>",   
              "<td></td>",   
              "<td style='font: 12pt verdana, sans-serif; padding: 5px 0px 5px 5px; background-color: #ddd; text-align: left'>Yesterday</td>",   
              "<td style='font: 12pt verdana, sans-serif; padding: 5px 0px 5px 5px; background-color: #ddd; text-align: left'>Two Days Ago</td>",   
              "<td style='font: 12pt verdana, sans-serif; padding: 5px 0px 5x 5px; background-color: #ddd; text-align: left'>A week ago</td>",   
            "</tr>",  
            emailRow('Cost', 'Cost', yesterdayRow, twoDaysAgoRow, weekAgoRow),  
            emailRow('Average Cpc', 'AverageCpc', yesterdayRow, twoDaysAgoRow, weekAgoRow),  
            emailRow('Ctr', 'Ctr', yesterdayRow, twoDaysAgoRow, weekAgoRow),  
            emailRow('Average Position', 'AveragePosition', yesterdayRow, twoDaysAgoRow, weekAgoRow),  
            emailRow('Impressions', 'Impressions', yesterdayRow, twoDaysAgoRow, weekAgoRow),  
            emailRow('Clicks', 'Clicks', yesterdayRow, twoDaysAgoRow, weekAgoRow),  
        "</table>",  
      "</body>",  
    "</html>");  
  MailApp.sendEmail(email, "AdWords Account " + AdWordsApp.currentAccount().getCustomerId() + " Summary Report", "", {htmlBody: html.join("\n")});  
}  
  
function emailRow(title, column, yesterdayRow, twoDaysAgoRow, weekAgoRow) {  
  return "<tr> \  
        <td style='padding: 5px 10px'>" + title + "</td> \  
        <td style='padding: 0px 10px'>" + yesterdayRow[column] + "</td> \  
        <td style='padding: 0px 10px'>" + twoDaysAgoRow[column] + formatChangeString(yesterdayRow[column], twoDaysAgoRow[column]) + "</td> \  
        <td style='padding: 0px 10px'>" + weekAgoRow[column] + formatChangeString(yesterdayRow[column], weekAgoRow[column]) + "</td> \  
      </tr>";  
}  
  
// returns noon in the timezone of the account  
function getYesterday() {  
  var now = new Date(Utilities.formatDate(new Date(),   
      AdWordsApp.currentAccount().getTimeZone(), "MMM dd,yyyy HH:mm:ss"));  
  var yesterday = new Date(now.getTime() - 24 \* 3600 \* 1000);  
  yesterday.setHours(12);  
  return yesterday;  
}  
  
function getReportRowForDate(date) {  
  var accountDate = new Date(Utilities.formatDate(date,  
      AdWordsApp.currentAccount().getTimeZone(), "MMM dd,yyyy HH:mm:ss"));  
  var dateString = Utilities.formatDate(accountDate, "PST", "yyyyMMdd");  
  return getReportRowForDuring(dateString + "," + dateString);  
}  
  
function getReportRowForDuring(during) {  
  var report = AdWordsApp.report(  
      "SELECT Cost, AverageCpc, Ctr, AveragePosition, Impressions, Clicks " +  
      "FROM ACCOUNT\_PERFORMANCE\_REPORT " +  
      "DURING " + during);  
  return report.rows().next();  
}  
  
function formatChangeString(newValue,  oldValue) {  
  var x = newValue.indexOf('%');  
  if (x != -1) {  
    newValue = newValue.substring(0, x);  
    var y = oldValue.indexOf('%');  
    oldValue = oldValue.substring(0, y);  
  }  
    
  var change = parseFloat(newValue - oldValue).toFixed(2);  
  var changeString = change;  
  if (x != -1) {  
    changeString = change + '%';  
  }  
    
  if (change >= 0) {  
    return "<span style='color: #38761d; font-size: 8pt'> (+" + changeString + ")</span>";   
  } else {  
    return "<span style='color: #cc0000; font-size: 8pt'> (" + changeString +  ")</span>";   
  }  
}  
  
function SpreadsheetAccess(spreadsheetUrl, sheetName) {  
  this.spreadsheet = SpreadsheetApp.openByUrl(spreadsheetUrl);  
  this.sheet = this.spreadsheet.getSheetByName(sheetName);  
    
  // what column should we be looking at to check whether the row is empty?  
  this.findEmptyRow = function(minRow, column) {  
    var values = this.sheet.getRange(minRow, column, this.sheet.getMaxRows(), 1).getValues();  
    for (var i = 0; i < values.length; i ++) {  
      if (!values[i][0]) {  
        return i + minRow;  
      }  
    }  
    return -1;  
  }  
  this.addRows = function(howMany) {  
    this.sheet.insertRowsAfter(this.sheet.getMaxRows(), howMany);  
  }  
  this.writeRows = function(rows, startRow, startColumn) {  
    this.sheet.getRange(startRow, startColumn, rows.length, rows[0].length).setValues(rows);  
  }  
}