

Alan L. Sullivan (3152)  
Todd M. Shaughnessy (6651)  
J. Elizabeth Haws (11667)  
Snell & Wilmer L.L.P.  
15 West South Temple, Suite 1200  
Beneficial Tower  
Salt Lake City, Utah 84101-1004  
Telephone: (801) 257-1900  
Facsimile: (801) 257-1800

Mark Lambert (Cal. Bar No. 197410)  
Mark Weinstein (Cal Bar No. 193043)  
Cooley Godward Kronish, LLP  
Five Palo Alto Square  
Palo Alto, California 94306-2109  
Telephone: (650)843-5003

Attorneys for Plaintiff

**IN THE UNITED STATES DISTRICT COURT**

**FOR THE DISTRICT OF UTAH, CENTRAL DIVISION**

PUBLIC ENGINES, INC., a Delaware  
Corporation,

Plaintiff,

vs.

REPORTSEE, INC., a Delaware  
Corporation,

Defendant.

**SUPPLEMENTAL DECLARATION OF  
STEVEN MEYERS**

Case No. 2:10-cv-317

Honorable Tena Campbell

I, Steven Meyers, hereby declare under penalty of perjury as follows:

1. I am an individual resident of the State of Utah, over the age of majority, and competent in every respect to make this declaration. The statements contained in this declaration

are based on my own personal knowledge, except for those statements explicitly made on information and belief. For statements made on information and belief, I am reliably informed and believe such matters to be true.

2. I am currently employed as the Director of Technical Operations of Public Engines, Inc. (“Public Engines”). In that capacity, I am responsible for the company’s data center operations and the management of all hardware systems and security related to the CrimeReports.com application and associated technologies.

3. I submit this declaration to supplement my declaration in this matter dated April 9, 2010, and to respond to certain issues raised by Defendant ReportSee, Inc. (“ReportSee”) in the Declaration of Mark Colin Drane and Defendants’ Memorandum Opposing Plaintiff’s Motion for Preliminary Injunction, both dated May 14, 2010.

4. In these papers, Mr. Drane and ReportSee claim that during 2009 and 2010 ReportSee accessed the data in CrimeReports.com through a “widget” located on the San Jose Police Department (“SJPD”) webpage. A widget is a “window” or “frame” on a webpage that displays the content of another, different webpage. Through the widget on the SJPD webpage, a user can access, search, and view all of the crime report data that appears on CrimeReports.com.

5. When a web browser accesses CrimeReports.com, our computers maintain an electronic log of that activity. That log shows when and how the browser accessed our computers and what it did while it was there. Public Engines has detailed logs showing ReportSee’s scraping of data from CrimeReports.com.

6. I understand that Mr. Drane and ReportSee are claiming that when ReportSee was scraping data from CrimeReports.com during 2009 and 2010, it was doing so through a CrimeReports.com widget, and specifically the widget on the SJPD webpage. As explained in detail below, those statements are inaccurate. Public Engines’ log files confirm that ReportSee’s scraper did not access CrimeReports.com via the widget on the SJPD webpage.

## SJPD Widget Code Was Not Requested

7. When a browser accesses the web server for CrimeReports.com via a widget, the log files associated with that access show this. Below is an example of what a log file looks like when a user accesses CrimeReports.com via the SJPD widget:

```
173.188.2.190 - - [23/Apr/2010:08:41:17 -0600] "GET
/widget/widget.html?utm_source=Crimereports.com&utm_medium=LargeWidget&utm_campaign=Widget&CRAddress=San%20Jose,%20CA,~Gilroy,%20CA,~Los%20Altos,%20CA,~Los%20Altos%20Hills,%20CA,~Los%20Gatos,%20CA,~Monte%20Serenio,%20CA,~Palo%20Alto,%20CA,~San%20Jose%20State%20University,%20CA,~Santa%20Clara,%20CA,~Saratoga,%20CA,~Sunnyvale,%20CA,&CRSearch=San%20Jose,%20CA&CRAgencyName=San%20Jose%20Police%20Department&Width= HTTP/1.1" 200 9504
"http://www.sjpd.org/crimestats/CrimeReports.html?CRSearch=San%20Jose,%20CA"
"Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9.1.9) Gecko/20100315
Firefox/3.5.9 GTB6 (.NET CLR 3.5.30729)"
```

8. The green portion above shows the actual page being requested by the user. It is a “GET” request, meaning that there will be no additional options passed in the body of the request. The page “/widget/widget.html” is the actual page being requested and following it are some options that are passed to the page.

9. The red portion above shows the “referring” page, or the webpage that is asking for the information from the CrimeReports.com database. In this case, the referring page is on the “sjpd.org” website. Because the widget is displayed as a window inside a page on the sjpd.org website, the referring page serves as a kind of wrapper around our widget.

10. The blue portion shows the “user agent” or type of web browser being used to access the page. In this case, Firefox 3.5.9, running on Windows Vista.

11. The “widget.html” page instructs the browser to request additional files from our web server. For example, images, style sheets, text, etc are retrieved in order to fill out the layout of the page. Additionally, this page requests JavaScript computer code that works together with the widget application.

12. Below is an example list of files that are retrieved from our web server during a typical session when a user is accessing the website through a widget:

```
GET /widget/widget.html?[snipped options]
GET /widget/css/calendar.css?ver=105
GET /widget/css/dialog.css?ver=105
GET /widget/css/global.css?ver=107
GET /widget/js/calendar/calendar-min.js?ver=105
GET /widget/js/connection/connection.js?ver=105
GET /widget/js/dom/dom-min.js?ver=105
GET /widget/js/event/event-min.js?ver=105
GET /widget/js/utility.js?ver=105
GET /widget/js/yahoo/yahoo-min.js?ver=105
GET /widget/img/CrimeReportsLogoBlue.png
GET /widget/img/btn_down_arrow.gif
GET /widget/img/btn_get-report.gif
GET /widget/js/calendar.js?ver=105
GET /widget/js/filter.js?ver=105
GET /widget/js/ga_area.js?ver=105
GET /widget/js/lightbox.js?ver=105
GET /widget/js/main_categories.js?ver=105
GET /widget/js/map.js?ver=105
GET /widget/js/prototype.js?ver=105
GET /widget/img/BlueFade.png
GET /widget/img/assault_inactive.png
GET /widget/img/btn_calendar_static.gif
GET /widget/img/btn_crime-types_closed_static.gif
GET /widget/img/burglary_inactive.png
GET /widget/img/calx.gif
GET /widget/img/filter-bar_bkd.gif
GET /widget/img/homicide_inactive.png
GET /widget/img/other_inactive.png
GET /widget/img/quality_of_life_inactive.png
GET /widget/img/robbery_inactive.png
GET /widget/img/theft_inactive.png
GET /widget/img/vehicle_inactive.png
POST /map/xmldata/isAjax/true/cb/1.8444955708616928
GET /widget/img/btn_calendar_roll.gif
POST /map/xmldata/isAjax/true/cb/33.665368467108436
GET /images/loading.gif
GET /map-icon/map-icon_map1_aqua.gif
GET /map-icon/map-icon_map1_green.gif
GET /map-icon/map-icon_map1_orange.gif
GET /map-icon/map-icon_map1_purple.gif
GET /map-icon/map-icon_list_aqua.gif
GET /map-icon/map-icon_list_green.gif
GET /map-icon/map-icon_list_orange.gif
GET /map-icon/map-icon_list_purple.gif
GET /map-icon/map-icon_list_red.gif
GET /map-icon/map-icon_map1_multiple.gif
GET /map-icon/map-icon_map1_red.gif
POST /map/xmldata/isAjax/true/cb/18.833380039803625
POST /map/xmldata/isAjax/true/cb/25.986896990096973
GET /widget/img/btn_calendar_roll.gif
```

```
GET /widget/img/theft_roll.png
OPTIONS /searchmap/xmldata/isAjax/true/cb/83.48764076417145
OPTIONS /searchmap/xmldata/isAjax/true/cb/28.824858414980103
GET /widget/js/calendar.js
GET /widget/js/calendar/calendar-min.js
GET /widget/js/connection/connection.js
GET /widget/js/dom/dom-min.js
GET /widget/js/event/event-min.js
GET /widget/js/filter.js
GET /widget/js/ga_area.js
GET /widget/js/lightbox.js
GET /widget/js/main_categories.js?ver=107
GET /widget/js/minimap.js?ver=107
GET /widget/js/prototype.js
GET /widget/js/utility.js
GET /widget/js/yahoo/yahoo-min.js
GET /widget/img/BlueFade2.png
GET /widget/img/CrimeReportsSmallPeople.png
```

13. The portions in blue are the requests for the JavaScript code that makes our widget application work. The JavaScript code contains a key that allows it to contact and access our Application Programming Interface, or API, which is defined as “an interface implemented by a software program which enables it to interact with other software.” (Wikipedia) In this case, our API allows the JavaScript portion of our application to interact with the server side of our application. Without that key, our server will refuse to provide data from our database.

14. Highlighted in red are the actual API requests made by the JavaScript code to the server side of our application. The API requests are “POST” requests, meaning that there is additional information encoded in the body of the request.

15. As can be seen from the examples above, if a user is accessing our web server through the SJPD widget (or any other widget), the log files show requests for /widget/widget.html and the associated JavaScript and other files that make up the application.

16. Along with my staff, I have reviewed all of the logs of web requests from ReportSee’s scraper from January 9, 2010 to March 31, 2010. We did not find a single request for /widget/widget.html, or any of the associated JavaScript and other files that work together to

make up the widget application. For example, the log of ReportSee's scraper for January 12, 2010, has the following requests:

```
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]1
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
GET /map/xmldata/isAjax/true/cb/88.17511697431006?[snipped]
(repeated a total of 3938 times)
```

17. Every single request from that IP address (174.129.243.60) was requesting direct access to Public Engines' API – in other words, direct access to Public Engines' database. Not one of them asked for widget code or the associated JavaScript files. This confirms that ReportSee did not access the CrimeReports.com web server via the SJPD widget, or any other widget.

#### **ReportSee's Scraper Did Not Use the Same API as the Widget**

18. We also know that ReportSee's scraper did not access the web server via the widget because it did not use the same API as the CrimeReports.com widget.

19. Prior to March 18, 2010, our server's API supported access through either "GET" or "POST" requests. The difference between a GET and POST is relatively straightforward and easy to recognize. Both are used to send data to the server, but a GET request includes the data in the URL, while a POST sends the data separately. The following example, in which some lines have been shortened for clarity (identified as [snipped]), shows the difference:

```
POST /map/xmldata/isAjax/true/cb/36.845178506844746 HTTP/1.1
Host: www.crimereports.com
User-Agent: Mozilla/5.0 (Macintosh; [snipped]) Gecko/20100401 Firefox/3.6.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
```

---

<sup>1</sup> "Snipped" means I have omitted the very long string of text that follows here.

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://www.crimereports.com/widget/widget.html?[snipped]
Content-Length: 1134
Cookie: PHPSESSID=2ptf2mt2f83fedt25n33gtep74; [snipped]
Pragma: no-cache
Cache-Control: no-cache
```

```
dayrange=7&custom_date_start=&custom_date_end=&startdate=2010-5-11&enddate=2010-5-18&fdate=Oct%201%2C%202007&three_day=Sat%20May%2015%202010%2015%3A53%3A45%20GMT-0600%20(MST)&seven_day=Tue%20May%2011%202010%2015%3A53%3A45%20GMT-0600%20(MST)&two_week=Tue%20May%2004%202010%2015%3A53%3A45%20GMT-0600%20(MST)&thirty_day=Sun%20Apr%2018%202010%2015%3A53%3A45%20GMT-0600%20(MST)&largeMap=&api_key=5h9a38ecriafiupriawRo6dluBiepro1ziakiazluspieq9uyia2iutouwrIesWi&&filter_categories[]=HOMICIDE&filter_categories[]=BREAKING%20%26%20ENTERING&filter_categories[]=ROBBERY&filter_categories[]=THEFT&filter_categories[]=THEFT%20OF%20VEHICLE&filter_categories[]=OTHER%20SEXUAL%20OFFENSE&filter_categories[]=SEXUAL%20ASSAULT&filter_categories[]=ASSAULT&filter_categories[]=ASSAULT%20W%2F%20DEADLY%20WEAPON&filter_categories[]=THEFT%20FROM%20VEHICLE&dayrange=7&custom_date_start=2010-5-11&custom_date_end=2010-5-18&mapradius=1&center_lat=37.3393857&center_lng=-121.8949555&address_lat=37.3393857&address_lng=-121.8949555&map_minlat=37.322666405074756&map_maxlat=37.356172359888916&map_maxlng=-121.86206817626953&map_minlng=-121.92781448364258&showS0=0
```

20. I have highlighted the body of the request in blue to differentiate it from the header of the request. The two are separated by a blank line. The body contains the specific information being requested from the server's API, including parameters such as date ranges, categories of crimes, and the location.

21. The requests we received from ReportSee's scraper looked like this:

```
GET
/map/xmldata/isAjax/true/cb/88.17511697431006?api_key=5h9a38ecriafiupriawRo6dluBiepro1ziakiazluspieq9uyia2iutouwrIesWi&address_lat=37.231&address_lng=-121.97&center_lat=37.231&center_lng=-121.97&custom_date_end=2010-1-11&custom_date_start=2010-1-8&dayrange=&enddate=2010-1-11&fdate=Aug+26%2C+2007&filter_categories%5B0%5D=BURGLARY&filter_categories%5B1%5D=HOMICIDE&filter_categories%5B2%5D=THEFT&filter_categories%5B3%5D=RAPE&filter_categories%5B4%5D=ROBBERY&filter_categories%5B5%5D=ASSAULT&filter_categories%5B6%5D=ASSAULT+W%2F+DEADLY+WEAPON&filter_categories%5B7%5D=VEHICLE+BURGLARY&filter_categories%5B8%5D=OTHER+SEXUAL+OFFENSE&filter_categories%5B9%5D=VEHICLE+THEFT&filter_categories%5B10%5D=ARSON&filter_categories%5B11%5D=KIDNAPPING&largeMap=&map_maxlat=37.251&map_maxlng=-121.93&map_minlat=37.211&map_minlng=-
```

```
122.01&mapradius=1&one_month=&seven_day=&showS0=0&startdate=2010-1-
8&three_day=&two_week= HTTP/1.12
Host: www.crimereports.com
Connection: close
Accept-encoding: gzip, deflate
User-agent: Zend_Http_Client
keepalive: 1
```

22. This particular request was made at 1:10am MST on January 12, 2010. It is typical of the requests made to our servers through March 18, 2010, when we changed our server API so that it would no longer allow GET requests. Highlighted in blue are the options passed in the query string as part of the GET request. However, legitimate API requests from the widget always use POST requests. Because ReportSee's scraper was using a GET request and not a POST request, it was not accessing our server through a widget.

23. When we changed our web server on March 19, 2010, to support only a POST request, ReportSee's scraper failed, again proving that it was not accessing the web server through a widget. In response to that change and in order to continue scraping, ReportSee changed its scraper after March 19 to use the POST API, which allowed it to continue scraping.

### **ReportSee's Requests Do Not Specify the Widget as Their Referrer**

24. We also know that ReportSee's scraper did not access the web server via the widget because the log files do not show the widget as the referrer.

25. As previously noted, web browsers include a "referrer" (actually spelled "Referer" in the HTTP headers) in their requests. A referrer tells the server which specific page directed the browser to that content. Below is an example of a log entry from a legitimate widget request to our API:

```
173.188.2.190 - - [23/Apr/2010:08:41:19 -0600] "POST
/map/xmldata/isAjax/true/cb/1.8444955708616928 HTTP/1.1" 200 727
"http://www.crimereports.com/widget/widget.html?utm_source=Crimereports.com&utm_m
edium=LargeWidget&utm_campaign=Widget&CRAddress=San%20Jose,%20CA,~~Gilroy,%20CA,~
Los%20Altos,%20CA,~Los%20Altos%20Hills,%20CA,~Los%20Gatos,%20CA,~Monte%20Sereno,%
20CA,~Palo%20Alto,%20CA,~San%20Jose%20State%20University,%20CA,~Santa%20Clara,%20
```

<sup>2</sup> It should be noted that from "GET" to "HTTP/1.1" is actually a single line of the request; it has been wrapped to fit within the margins of this document.



```
CA,~Saratoga,%20CA,~Sunnyvale,%20CA,&CRSearch=San%20Jose,%20CA&CRAgencyName=San%20Jose%20Police%20Department&Width=" "Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9.1.9) Gecko/20100315 Firefox/3.5.9 GTB6 (.NET CLR 3.5.30729)"
```

26. I have highlighted the referrer in red. The above example shows that the web browser was directed by “http://www.crimereports.com/widget/widget.html” to make the indicated POST request to our API.

27. Below is an example of a request from ReportSee’s scraper, which illustrates that it did not operate through widget requests to our API:

```
174.129.243.60 - - [09/Jan/2010:01:10:02 -0700] "GET /map/xmldata/isAjax/true/cb/88.17511697431006?api_key=5h9a38ecriafiupriawRo6dluBi epro1ziakiazluspieq9uyia2iutouwrIesWi&address_lat=37.231&address_lng=-121.97&center_lat=37.231&center_lng=-121.97&custom_date_end=2010-1-8&custom_date_start=2010-1-5&dayrange=&enddate=2010-1-8&fdate=Aug+26%2C+2007&filter_categories%5B0%5D=BURGLARY&filter_categories%5B1%5D=HOMICIDE&filter_categories%5B2%5D=THEFT&filter_categories%5B3%5D=RAPE&filter_categories%5B4%5D=ROBBERY&filter_categories%5B5%5D=ASSAULT&filter_categories%5B6%5D=ASSAULT+W%2F+DEADLY+WEAPON&filter_categories%5B7%5D=VEHICLE+BURGLARY&filter_categories%5B8%5D=OTHER+SEXUAL+OFFENSE&filter_categories%5B9%5D=VEHICLE+THEFT&filter_categories%5B10%5D=ARSON&filter_categories%5B11%5D=KIDNAPPING&largeMap=&map_maxlat=37.251&map_maxlng=-121.93&map_minlat=37.211&map_minlng=-122.01&mapradius=1&one_month=&seven_day=&showSO=0&startdate=2010-1-5&three_day=&two_week= HTTP/1.1" 200 1269 "-" "Zend_Http_Client"
```

28. I have highlighted the referrer this time with a yellow background for ease of reference, since in this case it is merely a single dash “-“. The dash indicates that the browser did not receive directions to access the API from any widget, nor did it receive directions to access the API from any webpage, such as CrimeReports.com. Instead, this shows that ReportSee’s scraper was directly accessing the server for CrimeReports.com, without being directed by any web page.

29. Additionally, when ReportSee changed its scraper on March 19, 2010, to use POST requests rather than GET requests, they also started including a referrer in their request:

```
184.73.50.166 - - [31/Mar/2010:00:27:17 -0600] "POST /map/xmldata/isAjax/true/cb/15.215975818729643 HTTP/1.1" 200 770 "http://www.crimereports.com/map" "Zend_Http_Client"
```

30. In this case, the referrer (again highlighted in red) is the main CrimeReports.com map page, not the widget. This is what a request would look like if it came through the main CrimeReports.com webpage. It is not what the request would look like if it came through a widget. For this additional reason, we know that ReportSee's scraper was not accessing our server via a widget.

### **Scraping vs. Web Page Indexing**

31. ReportSee compares its scraper to an indexer used by search engines such as Google. The two are very different. Wikipedia gives the following definition of a website indexer: "Search engine indexing collects, parses, and stores data to facilitate fast and accurate information retrieval."

32. When a user searches the internet using Google's search page, the Google index servers find pages relevant to the search terms entered by the user, and present a collection of links to those relevant pages. While Google does cache some (but not all) pages it indexes, it displays the cached pages in their original format. Most important, Google does not claim the data as its own. Rather, it makes very clear the origin of any pages it indexes. Additionally, it provides a link to the original page. It does not copy the content of the page and claim it as its own. Companies such as Google provide a service to websites by directing people to those websites. In fact, an entire industry has developed around "search engine marketing" or "search engine optimization" and many companies spend vast sums of money optimizing their website in order to appear higher in search engine results.

33. ReportSee's scraping of data from CrimeReports.com provides none of those benefits. The scraped data is incorporated into SpotCrime.com and presented to the user in a format very similar to the format used by CrimeReports.com – namely, pins on a map. The user is never told that the data was taken from CrimeReports.com. On the contrary, SpotCrime.com

claims that it owns the data that has been taken. The Terms of Use for SpotCrime.com, a copy of which are attached hereto as Exhibit A, state that “[a]ll words, pictures, content, graphs, charts, data and other matters presented or made available on the Website sourced from Us are copyrighted ‘Copyright 2007-2008 ReportSee, Inc.’”

34. A user encountering scraped data is not directed to view the data in its original form on CrimeReports.com. In short, unlike a search engine that provides a service to CrimeReports.com by making it easy for users to find our website, ReportSee is simply misappropriating the data and presenting it as if it owned that data.

#### **“Scrapers” vs. “Spiders”**

35. Google uses a spider to perform the work of indexing website pages. A spider has notable differences from a scraper. First, a spider’s purpose is to find all relevant pages on a website. A spider will download HTML pages and look for hypertext links to other pages. It will then continue on to spider those other pages. Second, search engine spiders respect robots.txt files (defined by the Robot Exclusion Standard). These files contain instructions on which parts of a website are considered indexable, and which parts are not. Third, a search engine spider, such as Google’s, is not interested in just a single site. It is intended to index the entire internet.

36. ReportSee’s scraper meets none of these criteria. Their scraper is intended to scrape the data directly from our API, while bypassing our HTML web pages. It does not respect the Robot Exclusion Standard. It has been designed to specifically target our website’s data, copy that data, and then incorporate that data into SpotCrime.com’s database.

#### **ReportSee Understands the Difference Between Scraping and Indexing**

37. ReportSee is aware of the difference between Scraping and website indexing. Its own Terms of Use for SpotCrime.com specifically recognize the distinction:

Direct Database Access prohibited. You may not directly access

our database except via the standard browser/graphic user interface, and You may not use any robot, script, or other automated tool to access or use the Website, any Information, or any data or the database. You may [not] manually access or copy the database, Information or Website or any part of [sic] thereof, except solely and exclusively in accordance with your license granted above. The above shall not prohibit any act permitted by law, as by, for example, a bona fide search engine that is automatically indexing the Website for its search engine database (as long as the result of a query to such search engine returns a link back to the Website), and the search engine is not re-presenting the Information in a manner that is or would be competitive to the Website, or using the Information to present in a search query response to a user, links to competing websites.

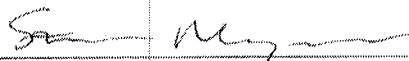
See Exhibit A.

38. ReportSee's Terms of Use prohibit the very conduct in which it has engaged. The Terms of Use prohibit scraping; they also prohibit directly accessing the database that underlies the SpotCrime.com website; they also prohibit accessing the data in any way other than through a standard web browser. ReportSee did all of these when it scraped data from CrimeReports.com.

39. ReportSee scraped crime report data from numerous law enforcement agencies. ReportSee scraped data from CrimeReports.com for all of the agencies or jurisdiction listed on Exhibit B, and perhaps others, during 2010.

I hereby affirm that the foregoing information is truthful under penalty of perjury of the laws of the United States.

DATED this 24 day of May, 2010.

  
\_\_\_\_\_  
Steven Meyers

## CERTIFICATE OF SERVICE

I certify that on the 24<sup>th</sup> day of May, 2010, a true and correct copy of the Supplemental Declaration of Steven Myers has been served on the following through ECF:

Joshua A. Glikin  
[glikin@bowie-jensen.com](mailto:glikin@bowie-jensen.com)

Jeffrey J. Hunt  
[jjh@pwlaw.com](mailto:jjh@pwlaw.com)

David C. Reymann  
[dreymann@parrbrown.com](mailto:dreymann@parrbrown.com)

Walter Diercks  
[wdiercks@rwdhc.com](mailto:wdiercks@rwdhc.com)

  
\_\_\_\_\_