RAPTOR
REAL-TIME ASSESSMENT AND PLANNING TOOL FOR OREGON
JAVASCRIPT TRANSITION PLAN

Version 1.0
As of May 6, 2015
## VERSION HISTORY

<table>
<thead>
<tr>
<th>Version #</th>
<th>Implemented By</th>
<th>Revision Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Daniel Stoelb</td>
<td>05/06/2015</td>
<td>Initial version of transition plan</td>
</tr>
</tbody>
</table>
# Table of Contents

1. BACKGROUND ........................................................................................................ 4

2. SYSTEM OVERVIEW .......................................................................................... 5

3. TRANSITION PURPOSE ....................................................................................... 6

4. PROJECT OVERVIEW ......................................................................................... 6

5. PROJECT TIMELINES/DELIVERABLES ............................................................... 6

   5.1 Month 1 – 2 (Current Functionality Research): .............................................. 6

   5.2 Month 2 – 3 (JavaScript Template Research): .............................................. 7

   5.3 Month 3 – 6 (JavaScript Template Coding): .................................................. 7

   5.4 Month 6 – 12 (JavaScript Final Coding): ...................................................... 7

   5.5 Month 12 (Deployment): ............................................................................. 7

   5.6 Month 13 – 24 (Documentation and Followup): ......................................... 8
1 BACKGROUND

The Real-Time Assessment and Planning Tool for Oregon (RAPTOR) is the Oregon Office of Emergency Management’s (OEM) situational awareness mapping tool.

In 2009, the Science and Technology Directorate (S&T) of the U.S. Department of Homeland Security (DHS) launched the Virtual USA (vUSA) initiative, which is helping to create a future where jurisdictions at all levels have the capabilities necessary to voluntarily share information with each other, as appropriate and authorized, regardless of the data format. The states of Alaska, Idaho, Montana, Oregon and Washington agreed to partner with the U.S. Department of Homeland Security (DHS) on a component of the Virtual USA Initiative, the Pacific Northwest (PNW) Pilot. The pilot advanced a technical and cultural shift in how the Nation shares information during an emergency.

Within Oregon, the State’s Office of Emergency Management (OEM), Department of Transportation (ODOT), the Department of Administrative Services (DAS) and Multnomah County’s Department of Emergency Management agreed to partner on the development of a GIS-enabled situational awareness prototype (i.e. VENOM – the Virtual Emergency Network of Multnomah). On behalf of their partners, Multnomah County took the lead on technical development of the prototype while OEM, ODOT, and DAS provided business requirements, access to vital information and geospatial datasets, and subject matter expertise. The prototype furnished a much needed virtual interoperability platform that allows collaboration and a coordinated response across the State’s Emergency Management community. The tool was the first step towards new emergency management capabilities, offering real-time situational information in combination with ‘traditional’ Geospatial Information Systems (GIS) layers to create a comprehensive picture of existing and potential situations. VENOM also served as a prototype for Oregon’s contribution to the Virtual USA (vUSA) Pacific NW Pilot.

As of October 2011, the State of Oregon’s OEM and DAS rebranded the tool as RAPTOR – the Real-time Assessment & Planning Tool for Oregon.

RAPTOR is the statewide version of the VENOM tool, built on the original concepts and stakeholder needs that were defined in VENOM. RAPTOR will allow OEM to spatially display interrelated information and aggregate information from various systems such as OR-IRIS, SAFE, Bridge, Ops Center, WebEOC, Computer Aided Dispatch (CAD), ODOT T-TIP and the National Weather Service into a geospatial platform. This allows for a real-time comprehensive situational picture. Examples of this include:

- Displaying the location of hospitals around a rapidly escalating traffic incident
- Showing the location of schools or childcare centers around a hazardous material spill
- Showing the infrastructure affected by a storm or natural disaster
- Providing predictive modeling tools to support ‘what-if’ scenario development and related planning activities
• Facilitated improved regional coordination and interoperability

RAPTOR’s potential value is that it instantly displays the relationships between events, allowing emergency operations centers (EOC’s) to be alerted to potential issues and providing real-time analysis for decision support. As this tool will complement other existing systems, it is intended to promote a multi-platform model of GIS information sharing amongst all levels of government via standardized feeds and data services such as GeoRSS, .xml and map services. This allows maximum information sharing with OEM regardless of which system agencies and localities are using for daily operations and in emergencies, therefore maximizing existing investments and minimizing potential costs.

2 SYSTEM OVERVIEW

The RAPTOR program is built on the ESRI Flex 3.0 API (Application Program Interface). As such, users need to have Adobe Flash plug-in version 10.0 or newer. This platform allows the application to run natively within any browser.

There are three different versions of RAPTOR, which are described below:

1. Public Version: the general public can view this version of RAPTOR at the following url: https://raptor.oregon.gov/raptorfx. Keep in mind that some of the layers are restricted and not available within this version of RAPTOR.

2. OpsCenter Version (RAPTOR/RAPTOR Training): entities and personnel that work in emergency management or are response personnel can access the OpsCenter version of RAPTOR through the single-sign-on portal at https://oregonem.com. This does require a user name and password for access. Through that same webpage, users can request an account. This version of RAPTOR includes the ability to view and populate incident information within the map (Oregon Planning tool). This same ability can be found in the RAPTOR Training site as well, with the caveat that any incident information populated in the training version of RAPTOR using the planning tools is for exercise/training purposes only.

3. PCII Version: entities and personnel that work for (or contracted by) state, local, or federal government can access the protected critical infrastructure information version of RAPTOR. This sign-on requires users to have first received PCII certification prior to registering for an account. PCII certification can be received via the following link at https://pciims.dhs.gov. After receiving certification, users can request an account for entry at https://raptor.oregon.gov. Users must supply a username and password for access. This version of RAPTOR includes Department of Homeland Security (DHS) data from the HSIP Gold program. This information contains sensitive data, such as utility lines and energy facilities. Additionally, users can populate and view incident information.
3 TRANSITION PURPOSE

In an effort to meet user needs and stay current with technology trends, the RAPTOR system must transition to the newer JavaScript API. This API allows for cross platform viewing of the RAPTOR application and works natively with all browsers and devices.

ESRI has indicated that their support for the FLEX API will be ending effective 2017 (specific language has stated that the product will be discontinued June 1, 2016), so any applications utilizing the FLEX API will need to be transitioned out of that API and into JavaScript.

It is the goal of this plan to provide benchmarks and deliverables for the transition of the program from the Flex API to the JavaScript API.

4 PROJECT OVERVIEW

The project is composed of moving from one coding language to another. As such, the code re-write must be done from scratch since no crosswalk exists between the two languages. Keep in mind that the language (API) is explicitly for the “wrapper” of the application, not the underlying data. No matter which language is used, the underlying data is the same.

This transition will encompass the following:
1. Analysis of what tools are available currently within RAPTOR (current functionality)
2. Analysis of what templates are currently available within the JavaScript API
3. Coding of JavaScript API version to match current data, functionality, enhance simplicity
4. Cutover to new JavaScript API version of RAPTOR

5 PROJECT TIMELINES/DELIVERABLES

Listed below are the timelines and assumed deliverables for the transition to the JavaScript API. Keep in mind that tasks are subject to change depending upon current functionality within the new JavaScript API.

5.1 MONTH 1 – 2 (CURRENT FUNCTIONALITY RESEARCH):

- Tasks:
  - Document current functionality/tools available within the FLEX version of RAPTOR.
  - Create inventory of datasets consumed within each configuration of RAPTOR.
- Deliverable:
  - Completed document describing current functionality and datasets currently utilized.
- Issues/Concerns:
5.2 MONTH 2 – 3 (JAVASCRIPT TEMPLATE RESEARCH):
- Tasks:
  - Research available JavaScript templates to assist with transition to JavaScript.
- Deliverable:
  - Completed document describing templates currently available.
- Issues/Concerns:
  - None

5.3 MONTH 3 – 6 (JAVASCRIPT TEMPLATE CODING):
- Tasks:
  - Attempt coding of multiple templates to determine ease of use, complete analysis of options available.
- Deliverable:
  - Completed document that determines ease of use and viability for each template available
- Issues/Concerns:
  - Can be time consuming to research individual templates

5.4 MONTH 6 – 12 (JAVASCRIPT FINAL CODING):
- Tasks:
  - Code decided upon template to match current functionality within RAPTOR.
- Deliverable:
  - Completed code, testing for JavaScript version of RAPTOR.
- Issues/Concerns:
  - Tools not currently available within the JavaScript API may cause delays – easier to modify pre-existing tools instead of coding new ones (writing from scratch).

5.5 MONTH 12 (DEPLOYMENT):
- Tasks:
  - Deploy to SVN, Web Servers, etc.
- Deliverable:
  - Cutover to JavaScript
- Issues/Concerns:
  - Completion date may move due to coding issues/tools not being available
5.6 MONTH 13 – 24 (DOCUMENTATION AND FOLLOWUP):

- Tasks:
  - Finalize documentation (user guide) and inventory for new system
- Deliverable:
  - Documented new system to include inventory and user guide
- Issues/Concerns:
  - None