

**WEB RESOURCE PLATFORM.** K. Grimes,<sup>1</sup> J. Padams<sup>1</sup>, and G. Hollins<sup>1</sup>. <sup>1</sup>Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. (kevin.m.grimes@jpl.nasa.gov)

**Introduction:** The Web Resource Platform (WRP) is a suite of tools that enables data and application sharing across disparate domains. It has been designed to be flexible and extensible, allowing data products and applications to be removed from and added to the system in a distributed fashion. It consists of multiple disparate components, which interact with each other through clear and well-defined application programming interfaces (APIs). WRP leverages existing web-based technologies and software to expose, discover, operate upon, and visualize planetary data resources.

**Product Repository – Juneberry:** Juneberry, the product repository component of WRP, makes data products uniformly accessible via REST calls (that is, "webifies" them). At the highest level, Juneberry exposes a directory of data files that exist on a user's file system to other users on the network. However, in addition to simply viewing a user's shared directory, the data files themselves can be webified, allowing users to interact with components of them via well-defined REST calls. Plug-ins can be created for different data product types, and then individual components of those images can be accessed and manipulated using simple web requests. These components include VICAR labels, PDS labels, and other pieces of information, including the raw product data itself.

If a directory containing images has been webified using Juneberry, for example, these images can be accessed directly via REST calls:

[https://pds-imaging.jpl.nasa.gov/.../1506288646\\_1506388236/N1506292227\\_1.IMG/0/image\[\]?output=gif](https://pds-imaging.jpl.nasa.gov/.../1506288646_1506388236/N1506292227_1.IMG/0/image[]?output=gif)

Since the image is in the familiar .gif format, Juneberry is able to webify it. Once the image has been webified, it can be manipulated by the user via REST calls. Suppose, for example, a user desires to resize the above image from the original 512x512 to 512x256. This can be achieved with the following URL:

[https://.../1506288646\\_1506388236/N1506292227\\_1.IMG/0/resized/512x256\[\]?output=gif](https://.../1506288646_1506388236/N1506292227_1.IMG/0/resized/512x256[]?output=gif)

This URL requests that image be returned as a image 512 pixels long and 256 pixels wide.

Users may also access raw image data with Juneberry. Image data is typically returned to the user as an array of arrays. The data of the above image can be retrieved with a simple REST call:

[https://.../1506288646\\_1506388236/N1506292227\\_1.IMG/0/raster/data\[\]?output=json](https://.../1506288646_1506388236/N1506292227_1.IMG/0/raster/data[]?output=json)

If the user would like to restrict the output to a specific range of data, they can specify that range within the selector brackets in the URL. Suppose, for example, the user would like to only retrieve the ten columns of the first row of data. This can be achieved as follows:

[https://.../1506288646\\_1506388236/N1506292227\\_1.IMG/0/raster/data\[0:1,0:10,0:1\]?output=json](https://.../1506288646_1506388236/N1506292227_1.IMG/0/raster/data[0:1,0:10,0:1]?output=json)

Juneberry has been applied successfully to large data stores, including some of those belonging to the NASA Planetary Data System.

**Resource Discovery – Cotinga:** In order to allow users to efficiently and accurately search data stores exposed by Juneberry, the data stores may be indexed using the resource discovery component of WRP, Cotinga. Cotinga is an indexing tool built on top of Apache Lucene and differs from other indexing tools in that it is designed to index webified products. That is, instances of Cotinga are built from product metadata information that has been exposed by Juneberry instances.

Cotinga crawls data files and indexes them in much the same way that Juneberry webifies them: it looks at the labels of the different images and indexes the files based off them. This allows users to make faceted queries out-of-the-box, without having to define complicated schemas or use other third-party tools. All that Cotinga requires to function properly is metadata such as PDS labels or VICAR labels.

**Tools Service:** At the highest level, the Tools Service component of WRP is software installed on a machine where tools (applications) are also installed. The Tools Service exposes these tools so that they can be invoked by a remote client somewhere else on the same network. As a result of calling services via the Tools Service, data can be returned via HTTPS to the client. It is then the client's responsibility to do what they want with the data and handle any errors returned by the Tools Service. In addition to the interaction with the tools themselves, the Tools Service also provides useful metadata about the tools, including usage (expected inputs and outputs).

The Tools Service is most frequently used on systems that contain specific installations of software that are difficult or impossible to recreate. For example, at JPL the VICAR image processing suite requires that a large number of environmental variables be set before it can be run properly. Additionally, even if users are able to set the environmental variables properly, the user will still likely have to tweak the build to ensure that it builds properly on their system. A user who does not want to endure the hassle of installing the VICAR software suite on their own machine may instead access another user's VICAR installation using the Tools Service.

Suppose, for example, that a user desires to run the VICAR `/generate/` program, which is a simple VICAR program that creates a VICAR file. Assuming that VICAR has been installed on machine `/host/` and Tools Service is running on port `/port/`, the user can use the VICAR `/generate/` program with the following REST call:

`https://host:port/ts/vic.gen?out=file.vic`

This request will create a VICAR file "file.vic" in the Tools Service data directory.

In addition to VICAR programs, other programs can easily be configured to work with the Tools Service. The shell command "ls" is among the executables that the Tools Service supports out-of-the box. The Tools Service also supports Java methods, such as "concat" or "env." If users would like to add more commands, they can simply create configuration files for them. Configuration files are usually only a few lines long and provide all the information that the Tools Service needs to expose the application to other users in the network.

Although not necessary, the Tools Service is often combined with Juneberry and Cotinga to provide a unified user experience. This typically involves pointing Juneberry to the data directory that the Tools Service puts files in after it generates them. This way, users can simply navigate to the Juneberry URL and download the files that they create using the Tools Service. If the file that they generated using the Tools Service is webifiable, they can manipulate it from there, too. If Cotinga is enabled on the system in addition to Juneberry, the files generated by the Tools Service that are webifiable will be indexed for efficient and accurate discovery in the future.

#### **End-User Web Application - Reference Client:**

The end-user web application consists of the following widgets: Product Browsing Widget, Search Widget, Tool Execution Widget, and Visualization Widget.

Together they provide a GUI that allows users to easily interact with the different components of WRP.

The Product Browsing Widget provides the capability of browsing product repositories hosted on different machines. Users can navigate through data products in different repositories. While the users navigate through data products, the metadata and actual data content can be viewed at both a high level and at a low level.

The Search Widget provides the capability of searching data products on remote machines. Different search methods such as wildcard search, fuzzy search, and range search are all possible. Additionally, terms and fields can be combined to perform queries.

In the Tool Execution Widget, users are provided with the capability of executing remote tools hosted on different machines. Users can interact with the Tools Service via either a text mode or a graphics mode; that is, users can manually type in commands to execute tools in a terminal-like interface or they can execute tools by interacting with HTML graphics.

In the Visualization Widget, users are able to visualize selective data products either as-is, on a two-dimensional map, or on a three-dimensional globe. Multiple data products can be combined, overlaid, and displayed at the same time. When displaying multiple data products, the displaying order of data products can be adjusted.

**Conclusion:** The Webification Resource Platform provides an array of capabilities that allow users to easily interact with functionalities that are not implemented on their own system. The Product Repository component of WRP allows users to expose their data products via a REST API. Users can choose to view the product (either in whole or in part), manipulate the product, or view its raw data. In addition to exposing data products, WRP also allows users to easily and accurately search for data products with Cotinga, an indexing tool built on Apache Lucene. The Tools Service component of WRP allows users to interact with applications and methods on remote machines via well-defined REST URLs; when combined with Juneberry and Cotinga, the products generated by the Tools Service can also be exposed and searched. Finally, the End-User Web Application allows those not comfortable with REST calls to interact with the rest of WRP's components in a straightforward manner. As a whole, the Web Resource Platform is a resource that can benefit any system which deals with large quantities of data.