VO-IRAF Integration

Project Definition Document

Project ID:	1.5
Revision:	1.0
Project Lead:	Mike Fitzpatrick (NOAO)
QA&T Engineer	Wes Young (NRAO)
Document Status:	Final
Reviewed	Live Review: 27 Apr 2011 (No Actions) Final Approval: 27 Apr 2011 (RP)

1 Project Description

One approach for the successful take-up of VO technologies by working astronomers is to integrate VO tools and protocols into the software environments already being used. IRAF has a large and established user base; a systematic integration of VO technologies into this analysis environment could potentially *VO-enable* a large number of tasks used everyday by astronomers, and easy access to remote data/services from the IRAF environment mean VO technologies will be seen as new capabilities of a familiar system rather than as an entirely new technology to be adopted.

The goal of this project is to incorporate VO technology, protocols and services into the IRAF environment in a natural and *seamless* manner. Key to this is the idea that the details of how VO works (jargon descriptions, web-service protocols, XML document formats, etc) should be hidden whenever possible from the astronomer (an exception being the low-level interfaces we expect to be used only by expert programmers). The result will be enhancements to existing applications and infrastructure in the IRAF system, newly developed analysis applications, and new capabilities in the scripting and programming environment provided by IRAF which allows astronomers to build their own applications from existing tools.

The project described here refers only to Year 1 efforts. Continued work in Year 2 is expected to incorporate elements of the Desktop Integration project and implementation of new service capabilities.

2 Project Interdependencies

This project is expected to support the following other projects:

- 1.1 VAO Portal Infrastructure and Integration
- 1.18 Desktop Integration

This project is expected to depend on the following other projects:

- 1.1 VAO Portal Infrastructure and Integration [Y1]
- 1.2 Scalable Cross Comparison [Y1]
- 1.3 Interoperable SED Access and Analysis [Y1?]
- 1.4 Interoperable Time Series Access [Y1?]

- 1.6 Registry Improvements [Y1]
- 1.7 VOClient/CLI Extensions [Y1]
- 1.10 VAO Security for Portals and VOSpaces [Y2+]
- 1.11 ADQL TAP Client [Y2+]
- 1.12 TAP ADQL Query Builder [Y2+]
- 1.18 Desktop Integration [Y2+]

This project is expected to depend on the following IVOA standards currently in development

- SIAv2
- TAP / ObsTAP
- PQL
- Registry Extensions
- Time-Series Access Protocols
- SED Access Protocols

3 High-level Requirements

Req.#	Requirement statement	Verif. Method*
	VOTable Support	
1.	Users shall be able to use a VOTable in places where tasks accept tabular data in other formats (ASCII files, FITS bintables, .tab files, etc) for input.	**D,A**
1.1	Users shall be able to access the <PARAM> and <INFO> elements of a VOTable as standard table header information.	**D**
1.2	Users shall be able to select specific rows and/or columns of a VOTable using the existing task functionality.	**D**
1.2.1	Users shall be able to identify a column in a VOTable by the 'id', 'name' or 'ucd' attribute of a <FIELD> or by column number.	**D**
1.3	Users shall be able to select a column in a VOTable for use in tasks that expect a list of values.	**D**
1.4	VOTable Interface code will support the IVOA Standard specification of the VOTable format at time of release.	**D**
	SAMP Interoperability	
2.	Users shall be able to interoperate with other SAMP (or WebSAMP) enabled applications.	**D,A**
2.1	Users shall be able to send messages to specific clients or broadcast to all available clients.	**D**
2.2	Users shall be able to execute IRAF tasks and set/retrieve information in/from the IRAF environment via SAMP messaging from clients that implement the required message types.	**D**

	Registry Query and Access	
3.	Users shall be able to query the VO Registry using the keyword-search services provided by the VAO Directory service.	**D,A**
3.1	Users shall be able to constrain the results of a search by supplying additional parameters to a query (e.g. bandpass, service type, etc).	**D,T**
3.2	Users shall have access all information in the query return record.	**D,T**
3.3	Users shall be able to refer to a VO Resource (in tasks requiring a Resource parameter) in multiple way:	
3.3.1	.... By the IVOA identifier	**D,T**
3.3.2	.... By the ShortName? value in the Registry record	**D,T**
3.3.3	.... By the "anonymous" result of a Registry query, or	**D,T**
3.3.4	.... By a "personal alias" for the Resource	**D,T**
3.4	Users shall have the option to cache query search results to avoid redundant service calls.	**D**
	Data Query and Access	
4.	Users shall be able to access all DAL services supported by the underlying VOClient interface (SCS/SIA/SSA in Y1).	**D,T**
4.1	Users shall be able to use the WCS footprint of a 2-D image as the basis of a data access query.	**D,T**
4.2	Users shall be able to query multiple resources at a single position on the sky, OR a single resource for multiple positions.	**D,T**
4.3	Users shall be able to specify a base ServiceURL to data-access tasks expecting a Resource parameter.	**D,T**
4.4	Users shall be able to save the results of a query in a tabular format other than VOTable.	**D,T**
4.5	Users shall be able to select which data returned by a DAL service query is to be downloaded to local disk.	**D,T**

*Verification Method Codes?: T = Testing, D = Demonstration, I = Inspection, A = Analysis.

Comments:

Req 1: IRAF tasks dealing with tablular data have a pre-existing concept of what a "table" is, the interface code to be built here is responsible for mapping a VOTable to this concept.

Req 1.4: Backward compatability with older specifications will be a goal achieved if this is within the availabe resources and/or the effort is required to support the VOTable formats returned by a majority of VO services.

Req 3: Additional search services may be supported in later work once they are defined.

Req 3.2: Specialized methods will allow users to easily access commonly-used fields such as the Identifier or ServiceURL.

Req 3.3: IVO identifiers are unique to a Resource but are not convenient for the science user who may prefer to refer to their favorite data service simply as "DR7". Likewise users may simply want to query ALL currently registered services of a certain type looking for data. This Req. leads to a better user experience by permitting details of the Registry or a Resource to be hidden when unnecessary.

Req 3.4: This is an optimization.

Req 4.1: Simple DAL queries are based on the idea of a central position and search size, parameters easily obtained by computing an image footprint.

Req 4.2: Queries of multiple resources at multiple positions can be handled by the user in e.g. a script to loop over the lists.

Req 4.3: This permits client access to local, developmental, or otherwise unregistered services.

Req 4.5: Maintains the separation of query and access in DAL services, allows users to only access data of interest, e.g. because of the bandpass, resolution, data product level, etc.

4 Roadmap

Information given here may be deprecated during the design phase.

This work is expected to result in the following types of products:

- library or toolkit
- an end-user, desktop tool

The work for this project is expected to be delivered by these estimated release dates:

- *Summer 2011*

This project will update the following existing products:

- *IRAF core system updates, external package software*