

# Proposal for Quality Assurance of Services in the VO

**VERSION:** 0.4

**DATE:** 2015-06-15

**FROM:** Renaud Savalle [renaud.savalle@obspm.fr](mailto:renaud.savalle@obspm.fr) and Pierre Le Sidaner [pierre.lesidaner@obspm.fr](mailto:pierre.lesidaner@obspm.fr)

**TO:** ops@ivoa.net

---

## Contents

1	Introduction	1
2	Procedure	1
2.1	Initial email	2
2.2	Subsequent emails	2
2.3	Final action	2
3	Implementation	2
3.1	Email Alias	2
3.2	Database	3
3.3	QA Status	3
4.	Conclusion	5
5.	References	5

## 1 Introduction

We have developed a DAL service validator and a program to perform daily validation of all the services in the VO [1]. We have identified a number of services which are *active* in the registry but do not perform correctly. A first class of services we want to consider here, are the services that do not reply for at least 40 days. We propose to a procedure to iterate with the curator, and eventually mark those services as *inactive* in the registry, thus cleaning it of such entries.

## 2 Procedure

## 2.1 Initial email

The procedure that we propose is as follow: if a service which is active in the registry has been found not responding for 40 days, we will send an email to the service curator. This email is composed automatically, but must be validated and sent by a human. In order not to flood the curators with many emails we will group by the services curated by a certain person. The content of this email is basically as follows:

*“You are the curator for the following services which have been found not responding for at least 40 days according to our validation service.”* Follows a list of services with a link to the validation program to check the result. *“Could you check the services registration, and if the services are obsolete, please change their statuses to inactive in the authoritative registry (ie where the services are declared).”*

According to OAI-PMH [2] the harvesting process allows to know the origin registry for each service. In addition, we keep a list of resources harvested regularly from all the publishing registries and are able to tell the curator in which registry its service is declared.

## 2.2 Subsequent emails

Following the initial email, we should keep checking the service every day and at during the same time we wait for an answer from the curator. If the curator does not reply, then we keep sending an email every week for up to 10 weeks. NB: at the moment due to lack of resources, this is done only every 6 months.

## 2.3 Final action

If we have sent 10 emails in 10 weeks and we still have no reply from the curator, we shall ask the IVOA executive to consider marking the service inactive in the registry where it is declared.

# 3 Implementation

## 3.1 Email Alias

In order to manage the process, we have defined a new email alias. For now it is: [registry.validation@obspm.fr](mailto:registry.validation@obspm.fr) and may later move to a more formal alias hosted at [ivoa.net](http://ivoa.net). This alias will include a core of people who care about the management of services in the registry.

## 3.2 Database

We maintain a PostgreSQL database to keep the history of the services. This database has an authenticated web front-end to allow exploring and managing this history. The database includes a main table *services* containing for each service:

1. IVO id and URL (serving as Primary Key)
2. Service Type (CS, SIA, SSA...)
3. Validation status for the spec
4. Validation status for the VOTable
5. Number of days the status has remained the same
6. Contact name and email of the curator
7. Number, dates and contents of emails sent
8. QA Status: status of the service relating to the quality assurance
9. QA History: a field describing the problems found with the services
10. Date when QA Status changed
11. VAO OAI Status

The enforcement of certain statuses in the DB is be made using SQL constraints.

The database also contains the following tables:

- *emails* to keep track of messages sent
- *registries* mirrors the RoR contents
- *resources* keeps a copy of the resources from the VOPDC Full Searchable Registry. Used to join with the *services* table to get the resources parameters

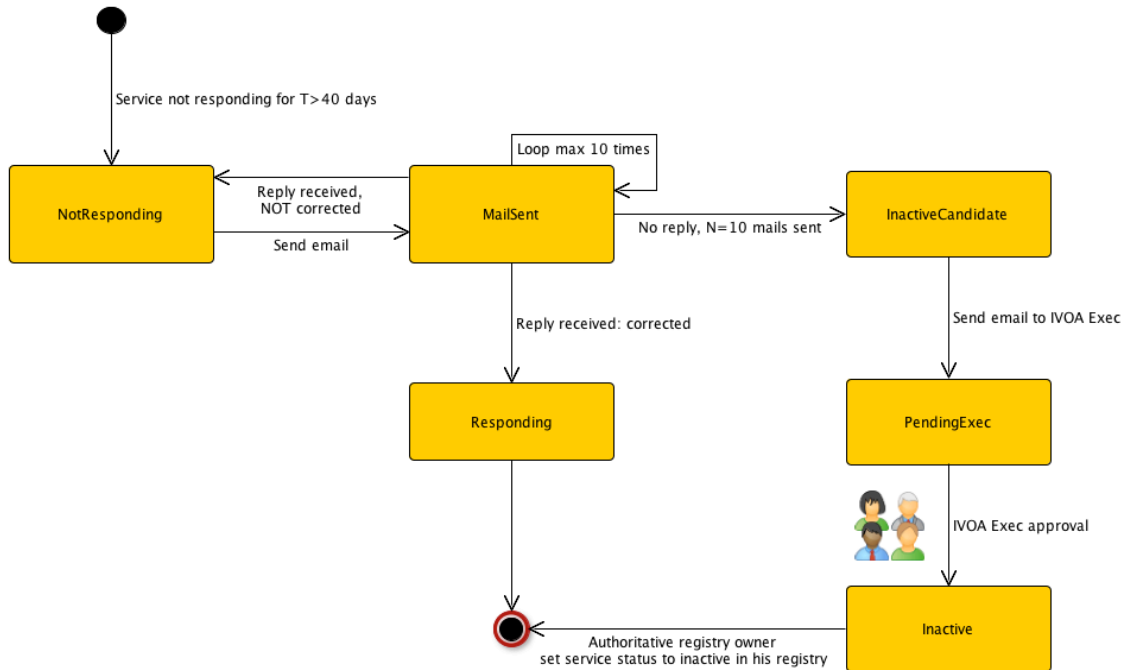
## 3.3 VAO OAI Status

When we consider services from the EURO-VO registry, we also check the status of the corresponding resources in the VAO registry using OAI-PMH. The OAI status is queried using a web service. Typical values are “active” or “idDoesNotExist”.

Since 2014 we consolidate the DB using VOParis full searchable registry which contains only active services.

## 3.4 QA Status

The QA status follows the following rules described in the following UML state diagram



State Diagram for the QA Status  
V1.1

## 3.5 Web Application

### 3.5.1 Availability

The web application is available at <http://voparis-validation.obspm.fr/adm>

### 3.5.2 Authentication

The web application requires the use of a login/password, each member of the alias defined in section 3.1 can request an access.

### 3.5.3 Main features

The web application has several features:

### 3.5.3.1 Registry Exploration

- `reg-update` Update the list of publishing registries from the RoR into the registries table
- `reg-harvest` Harvest the publishing registries from the registries table and fills the resources table

### 3.5.3.2 Services Validation

- **psql-import** - Import a SQLite v3 DB into the PSQL database.
- **psql-display** - Explore the PSQL database.
- `psql-details` - View/edit the details of a services entry in the PSQL database.
- `vao-find-home-registry` - Find the home registry for a IVOA resource using VAO web service
- `psql-update-home-registry` - Update the home registry column in the PSQL database using VAO web service
- `psql-update-oai-status` - Update the OAI status column in the PSQL database using VAO OAI web service
- `psql-edit-email` - Prepare (create or update) an email message
- `psql-display-emails` - Display queued and sent email messages, allows to send them

### 3.5.4 Typical Process

1. Use `psql-import` to update the PSQL DB with the list of non responding services from a SQLite3 DB generated daily
2. Use `psql-display` to explore the DB check the services, prepare emails
3. Use `psql-display-emails` to review the email queues and send the emails

## 4 Conclusion

Thanks to the procedure described above we are able to clean the registry of services not responding, and display the progression of those services in the registries.

## 5 References

[1] R. Savalle, P. Le Sidaner, A. Shih, J. Normand, G. Coquatre Zielgen: Validation and quality assurance for IVOA services. ADASS XXI (2011)

[2] <http://www.openarchives.org/>

## 6 Appendix: History of this document

V0.4 2015-06-17 minor corrections, for circulation to [ops@ivoa.net](mailto:ops@ivoa.net) during Interop Sesto June 2015

V0.3 2015-06-15: corrections and additions, for circulation to [registry.validation@obspm.fr](mailto:registry.validation@obspm.fr) during Interop Sesto June 2015

V0.2: replaced Tom Donaldson by Theresa Dower [dower@stsci.edu](mailto:dower@stsci.edu) in the TO: list, added parameter, constraints in the database part, added part on Web Application

V0.2 2012-01-03 initial version