OASIS ebXML Registry

Proposal: Iteration Support in Queries

Category: Enhancement to existing feature

Date: July 29, 2002

Version 0.3

Authors: Farrukh Najmi, Nikola Stojanovic

Table of Contents

Status of this Document

This note describes the initial proposal for the `Iteration Support in Queries` work item for OASIS ebXML Registry V3.0. It is expected that the Query sub-team of the OASIS ebXML Registry TC will improve upon this initial proposal and then submit it for consideration by ebXML Registry TC at large.

1 Abstract

This document proposes a minor enhancement to existing query capabilities feature of the OASIS ebXML Registry targeted for version 3.0. The `Iteration Support in Queries` enables the registry to handle queries that match very large results sets in a scalable manner. The feature enables the large result set to be broken-down into smaller chunks and retrieved in multiple queries.

2 Motivation

The following motivations drive this proposal:

1. Enable large-scale registries.
2. Gracefully handle queries that are not specific enough.

3 Assumptions

The following assumptions are made in this proposal:

1. Feature will be enabled in a backward compatible manner.

4 External Dependencies

This proposal depends upon the following external artifacts and event:

- No external dependencies

5 Use Cases

There are many scenarios where the iteration support within queries is necessary.

5.1 Query Is Too Broad

An user may submit a query that is not specific enough and matches a result set that is too large. This can be the case of a naïve user. It may also be the case of a malicious user engaged in a `Denial of Service (DOS)` attack.

5.2 Result Set Too Large

A registry may be of a very large scale and have very large result sets that match reasonable queries.

6 Changes To ebRS

```
[Note]The following section will replace section 8.1
     in ebRS. Note that there are some formatting
     changes (syntax, parameters, returns,
     exceptions) that are there for improved
     clarity. The format changes have been
     influenced by the recently released UDDI V3
     specs. Finally note the only change to XML
     Schema is the introduction of startIndex,
```

```
            maxResults to AdhocQueryRequest, and the
            introduction of startIndex and totalResultCount
            in AdhocQueryResponse.
```

6.1 Ad Hoc Query Request/Response

A client submits an ad hoc query to the QueryManager by sending an AdhocQueryRequest. The AdhocQueryRequest contains a subelement that defines a query in one of the supported Registry query mechanisms.

The QueryManager sends an AdhocQueryResponse either synchronously or asynchronously back to the client. The AdhocQueryResponse returns a collection of objects whose element type depends upon the responseOption attribute of the AdhocQueryRequest. These may be objects representing leaf classes in [ebRIM], references to objects in the registry as well as intermediate classes in [ebRIM] such as RegistryObject and RegistryEntry.

Any errors in the query request messages are indicated in the corresponding query response message.

☞☞

☞☞**Figure 1: Submit Ad Hoc Query Sequence Diagram**

For details on the schema for the business documents shown in this process refer to Appendix **Error! Reference source not found.**.

6.1.1 AdhocQueryRequest

The AdhocQueryRequest is used to submit queries to the registry.

6.1.1.1 Syntax:

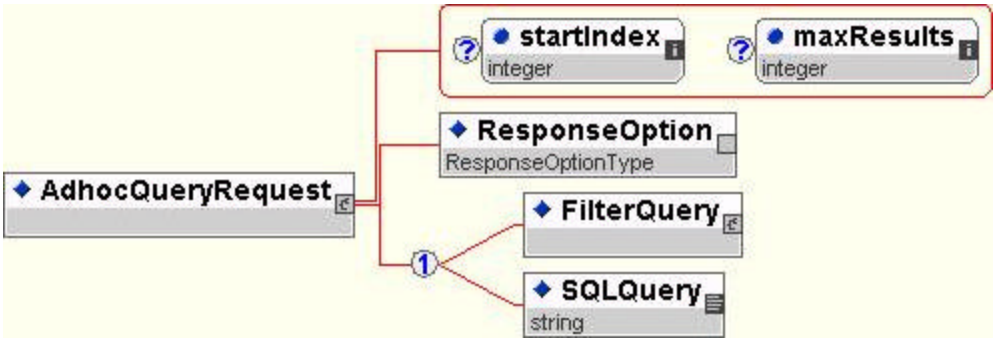

Figure 2: AdhocQueryRequest Syntax

6.1.1.2 Parameters:

- *FilterQuery:* This parameter specifies a registry Filter Query.
- *maxResults*: This optional parameter specifies a limit on the maximum number of results (that are instances of the specified return type), the client wishes the query to return. If unspecified, the registry should return either all the results, or in case the result set size exceeds an registry operator specific limit, the registry should return a sub-set of results that are within the bounds of the registry operator specific limit.
- *ResponseOption*: This required parameter allows the client to control the format and content of the AdhocQueryResponse to this request. See section 6.1.3 for details.
- *SQLQuery*: This parameter specifies a registry SQL Query.
- *startIndex*: This optional integer value is used to indicate which result set SHOULD be returned first results set when iterating over a large result set. The default value is 0, which returns result sets starting with index 0 (first result set).

6.1.1.3 Returns:

This request returns an AdhocQueryResponse upon success. See section 6.1.2 for details.

6.1.1.4 Exceptions:

In addition to the exceptions common to all requests, the following exceptions may be returned:

- InvalidQueryException: signifies that the query syntax was invalid. Client must fix the query syntax and re-submit the query.

6.1.2 AdhocQueryResponse

The AdhocQueryRequest sent by the registry as a response to AdhocQueryRequest.

6.1.2.1 Syntax:

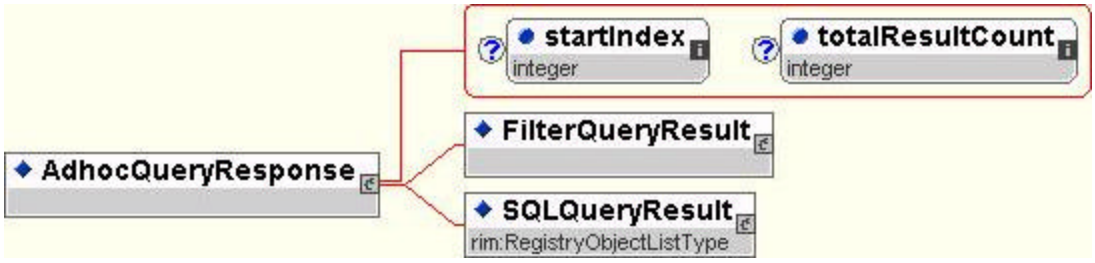

Figure 3: AdhocQueryResponse Syntax

6.1.2.2 Parameters:

- *FilterQueryResult:* This parameter specifies the result of a registry Filter Query.
- *SQLQueryResult*: This parameter specifies the result of a registry SQL Query.
- *startIndex*: This optional integer value is used to indicate the index for the first result in the result set returned by the query, within the complete result set matching the query within the registry. By defaut, this value is 0.
- *totalResultCount*: This optional parameter specifies the size of the complete result set matching the query within the registry. When this value is unspecified, the client should assume that value is the size of the result set contained within the result.

6.1.3 ReponseOption

A client specifies an ResponseOption structure within an AdhocQueryRequest to indicate the format of the results within the corresponding AdhocQueryResponse.

6.1.3.1 Syntax:

Figure 4: ResponseOption Syntax

6.1.3.2 Parameters:

- *returnComposedObjects:* This optional parameter specifies whether the RegistryObjects returned should include composed objects such as Classifications, ExternalIdentifiers etc. The default is to return all composed objects.
- *returnType:* This optional enumeration parameter specifies the type of RegistryObject to return within the response. Enumeration values for returnType are explained in section 6.1.3.3.

6.1.3.3 Enumeration returnType

Enumeration values for returnType are as follows:

?? ObjectRef - This option specifies that the AdhocQueryResponse may contain a collection of ObjectRef XML elements as defined in [ebRIM Schema]. Purpose of this option is to return just the identifiers of the registry objects.

?? RegistryObject - This option specifies that the AdhocQueryResponse may contain a collection of RegistryObject XML elements as defined in [ebRIM Schema]. In this case all attributes of the registry objects are returned (objectType, name, description, …) in addition to id attribute.

?? RegistryEntry - This option specifies that the AdhocQueryResponse may contain a collection of RegistryEntry or RegistryObject XML elements as defined in [ebRIM Schema], which correspond to RegistryEntry or RegistryObject attributes.

?? LeafClass - This option specifies that the AdhocQueryResponse may contain a collection of XML elements that correspond to leaf classes as defined in [ebRIM Schema].

?? LeafClassWithRepositoryItem - This option specifies that the AdhocQueryResponse may contain a collection of ExtrinsicObject XML elements as defined in [ebRIM Schema] accompanied with their repository items or RegistryEntry or RegistryObject and their attributes. Linking of ExtrinsicObject and its repository item is accomplished using the technique explained in Section **Error! Reference source not found.** -**Error! Reference source not found.**.

If “returnType” is higher then the RegistryObject option, then the highest option that satisfies the query is returned. This can be illustrated with a case when OrganizationQuery is asked to return LeafClassWithRepositoryItem. As this is not possible, QueryManager will assume LeafClass option instead. If OrganizationQuery is asked to retrieve a RegistryEntry as a return type then RegistryObject metadata will be returned.

6.1.4 Iterative Query Support

The AdhocQueryRequest and AdhocQueryResponse support the ability to iterate over a large result set matching a logical query by allowing multiple AdhocQueryRequest requests to be submitted such that each query requests a different sliding window within the result set. This feature enables the registry to handle queries that match a very large result set, in a scalable manner.

The iterative queries feature is not a true Cursor capability as found in databases. The registry is not required to maintain transactional consistency or state between iterations of a query. Thus it is possible for new objects to be added or existing objects to be removed from the complete result set in between iterations. As a consequence it is possible to have a result set element be skipped or duplicated between iterations.

Note that while it is not required, it may be possible for implementations to be smart and implement a transactionaly consistent iterative query feature. It is likely that a future version of this specification will require a transactionaly consistent iterative query capability.

6.1.4.1 Query Iteration Example

Consider the case where there are 1007 Organizations in a registry. The user wishes to submit a query that matches all 1007 Organizations. The user wishes to do the query iteratively such that Organizations are retrieved in chunks of 100. The following table illustrates the parameters of the AdhocQueryRequest and those of the AdhocQueryResponses for each iterative query in this example.

AdhocQueryRequest Parameters		AdhocQueryResponse Parameters		
startIndex	maxResults	startIndex	totalResultCount	# of Results
0	100	0	1007	100
100	100	100	1007	100
200	100	200	1007	100
300	100	300	1007	100
400	100	400	1007	100
500	100	500	1007	100
600	100	600	1007	100
700	100	700	1007	100
800	100	800	1007	100
900	100	900	1007	100
1000	100	1000	1007	7

6.2 Filter Query Support

FilterQuery is an XML syntax…

7 Notes

These notes are here to not lose the thought and will be merged into the proposal later.

-
-