i2b2 Cell Messaging

Data Repository (CRC) Cell (1.1)

1 Table of Contents

2 Document Version History

Date	Version	Description	Author(s)
12/01/2006	0.9	First Version	Kristel Hackett
05/10/2007	1.0	Revision	Vivian Gainer
09/05/2007	1.1	Revision 1.1 changes	Rajesh Kuttan

3 Introduction

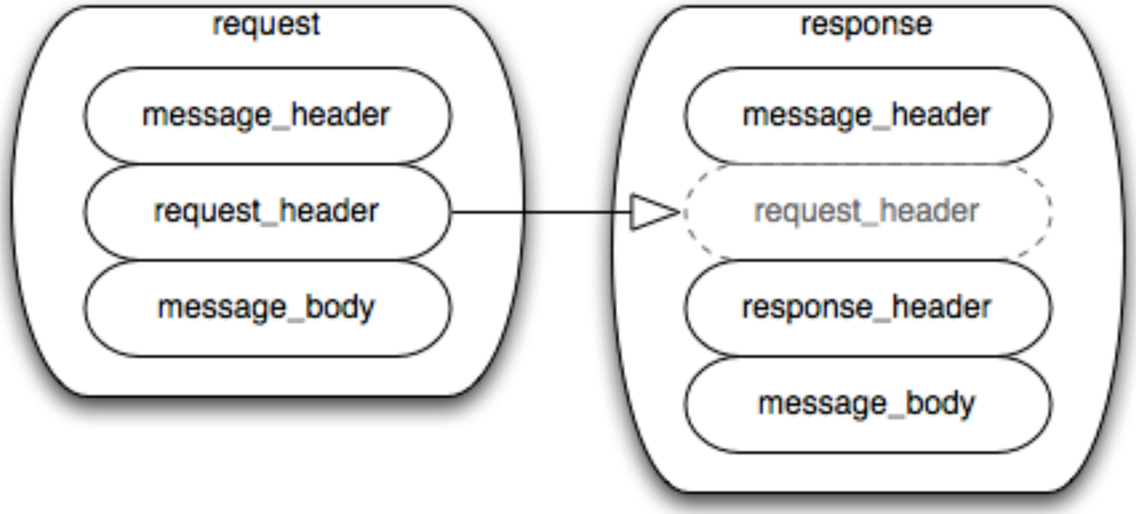
This document gives an overview of i2b2 cell messaging as well as a more detailed description of message formats specific to the Data Repository (CRC) Cell.

3.1 The i2b2 Hive

The Informatics for Integrating Biology and the Bedside (i2b2) is one of the sponsored initiatives of the NIH Roadmap National Centers for Biomedical Computing (http://www.bisti.nih.gov/ncbc/). One of the goals of i2b2 is to provide clinical investigators broadly with the software tools necessary to collect and manage project-related clinical research data in the genomics age as a cohesive entity – a software suite to construct and manage the modern clinical research chart. The i2b2 hive is a set of cells or modules that have a common messaging protocol that allow the cells to interact using web services and XML messages.

3.2 i2b2 Messaging Overview

All cells in the i2b2 hive must communicate using standard i2b2 XML messages. This message specifies certain properties that are common to cells and essential to the administration tasks associated with sending, receiving and processing messages. All requests are sent using a <request> tag and responses are returned using a <response> tag. The same <message_header> tag is used for both. The <request_header> is used for requests but may optionally be echoed back in the response. The response must include a <response_header>. The XSD specification of the i2b2 message permits individual cells to add cell-specific XML in the <message_body> tag. This cell-specific XML need not extend the i2b2 message schema since the i2b2 schema will allow insertion of tags from any namespace into the <message_body> tag. The following table illustrates the basic top-level elements contained within the request and response messages.



The i2b2 XML schema consists of three XSD files:

- **i2b2.xsd**
  This schema is not used directly to create i2b2 messages, but is included in the i2b2_request.xsd and the i2b2_response.xsd. It defines the <message_header> tag.

- **i2b2_request.xsd**
  This schema is used for validating i2b2 request messages. It defines the <i2b2:request> tag, which includes the <message_header> tag.

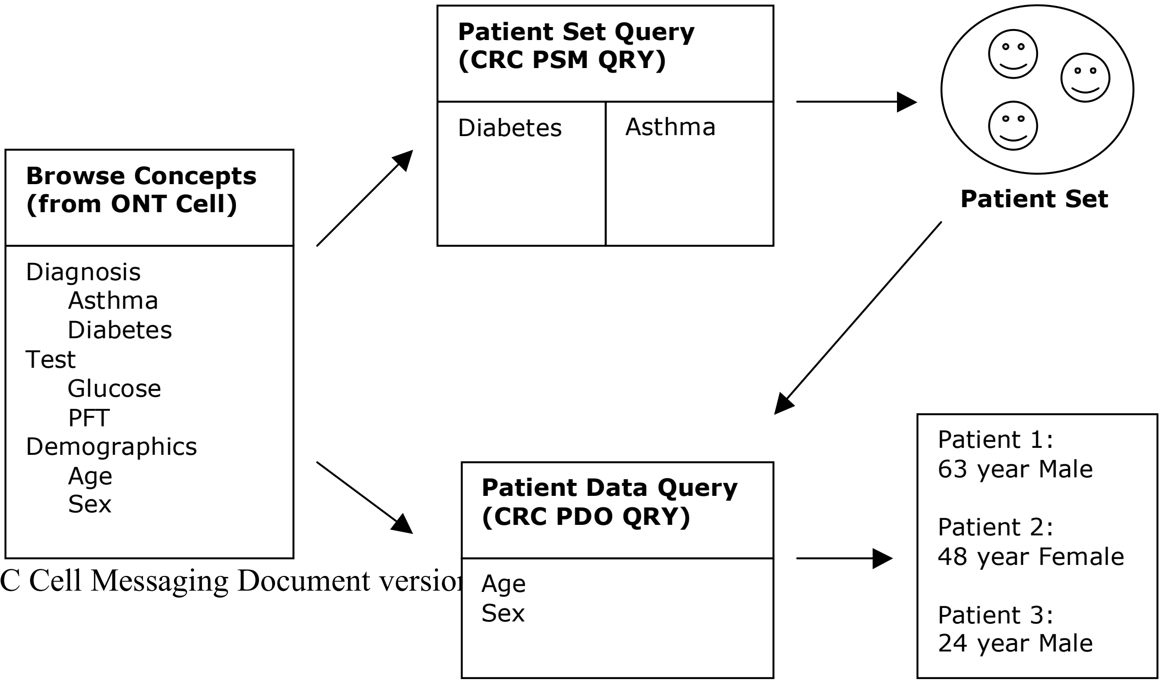
- **i2b2_response.xsd**
  This schema is used for validating i2b2 response messages. It defines the <i2b2:response> tag, which includes the <message_header> tag.

4 Data Repository (CRC) Cell Messaging Detail

The Data Repository Cell is one of the core cells in the i2b2 hive. Since much of the data in the repository is clinical in nature, it has also come to be known as the Clinical Research Chart (CRC) and the terms “data repository” and “CRC” are used interchangeably. The data repository is a warehouse of patient phenotypic and genotypic data that interacts with other cells to provide information for users. Communication with the CRC Cell, like all cells in the i2b2 hive, is handled via standardized XML web services. These XML messages conform to the i2b2 messaging standard described above, which allows cell-specific XML within the <message_body> tag. The rest of this document describes CRC-specific web services and the XML formats that encode them and illustrates how these XML messages are used to accomplish a set of interactions that correspond to typical CRC use cases.

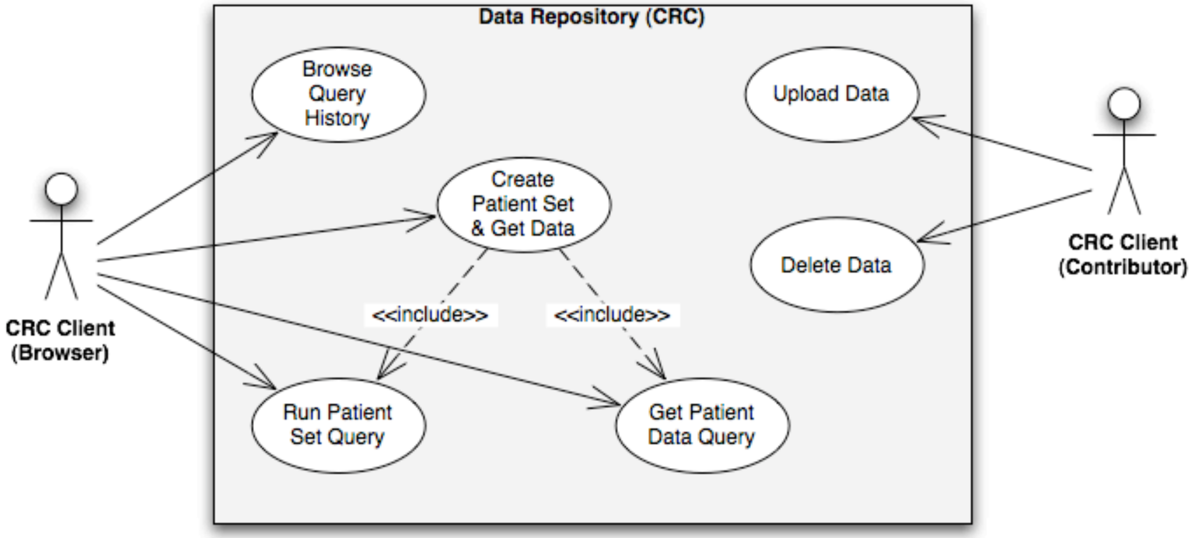
A typical CRC Client may want to define a patient set and then request patient data on that set. Both of these tasks require the user to first interact with another cell called the Ontology Management (ONT) Cell in order to choose concept codes to define the CRC request. Although the specific interactions with the ONT Cell are not described in this document, the following diagram shows the basic flow of information.

This diagram describes how a user may get the ages and genders for all patients who have either diabetes or asthma. The user starts by selecting the diagnoses ‘Diabetes’ and ‘Asthma’ from the ONT cell; these define the ‘Patient Set Query’, which creates a patient set in the data repository. Then the user selects the demographic concepts ‘Age’ and ‘Sex’ from the ONT cell to define the ‘Patient Data Query’. The patient data query returns the age and gender for all patients in the data set, those with diabetes or asthma.
.



4.1 Use Case

The CRC Cell is a repository of clinical data and has a set of services that respond to requests for patient data. A request might be issued by a client cell which is used by a researcher conducting a clinical trial in order to help gather a cohort. There are two types of clients or users, the 'browsing client' and the 'contributing client'. The contributing client adds content to the CRC by uploading patient data or deleting data by removing previous uploads. The browsing client has four possible interactions with the repository cell. The user may create queries that define patient sets, browse previous queries, rerun existing patient set queries and get specific patient data from a patient set.



4.2 Services / Messages

The CRC Cell provides services that support the interactions necessary for each of the four use cases described in the previous section. The services expect different message request types for each specific behavior or request.

4.2.1 Browse Query History

- Get a List of Saved Query Definitions – provides a list of all prior queries created by a client/user.
- Get a List of Saved Query Results – provides query results for given query run/instance.
- Get an XML Definition of a Defined Query – returns definition of the query

4.2.2 Run Patient Set Query

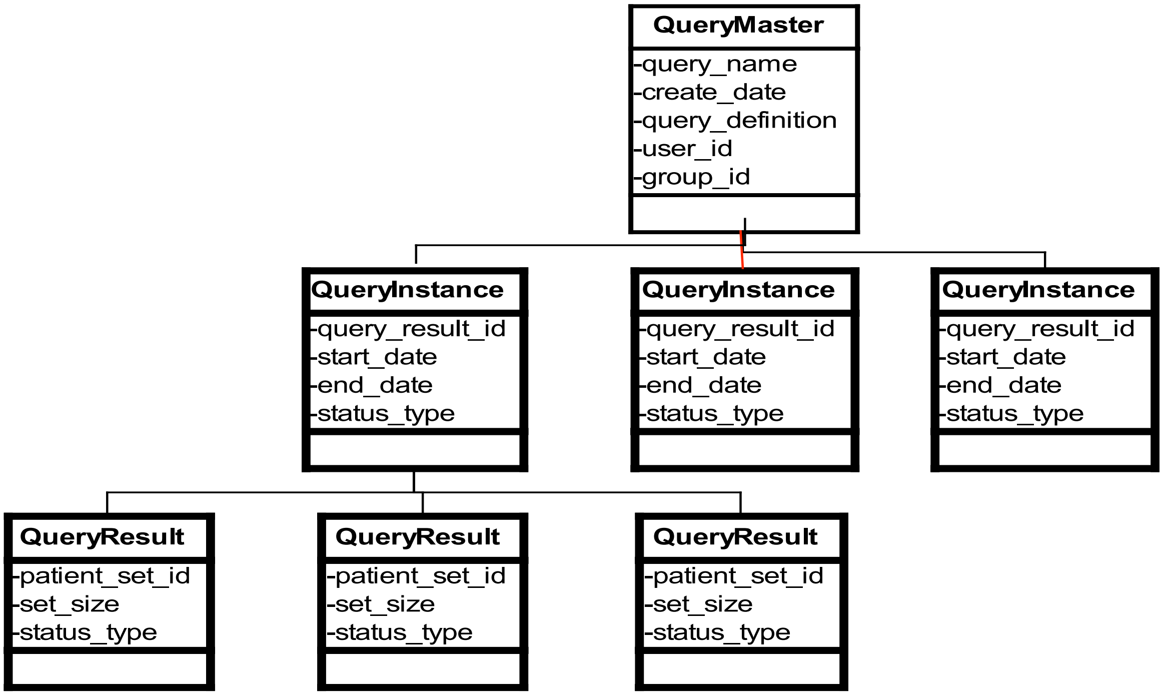
- Run Patient Set Query – runs the query and returns the result and its status

4.2.3 Get Patient Data Query

- Get Patient Data From a Patient Set - returns patient data object for the given patient set

4.3 Patient Set Query Service:

4.3.1 Conceptual Model:



QueryMaster holds master information about the query like the query name, query definition, user id, group_id. The query_definition element in QueryMaster holds the xml representation of the query constraint; its details are described below.

The run information of the query is recorded in QueryInstance and is created when ever the query is executed. There can be multiple QueryInstances for one QueryMaster and one QueryInstance can have multiple QueryResult's. QueryResult typically holds result information of the query, like id of patient set, the patient set size and result status.

4.3.2 Query Definition Details :

Example:

```
<query_definition>
      <query_name>Asthma/HTN</query_name>
      <query_timing>ANY</query_timing>
      <panel>
      <panel_number>1</panel_number>
      <invert>0</invert>
      <panel_date_from>1999-07-02Z</panel_date_from>
         <total_item_occurrences>1</total_item_occurrences>
         <item>
            <item_name>Diagnoses</item_name>
            <item_table>i2b2</item_table>
            <item_key>\i2b2\Diagnoses</item_key>
            <class>ENC</class>
            <constrain_by_date/>
         </item>
         <item>
            <hlevel>3</hlevel>
            <item_name>Acute Rheumatic fever</item_name>
            <item_table>TestRPDR</item_table>
            <item_key>\RPDR\Diagnoses\Circulatory system</item_key>
            <tooltip>Diagnoses \ Circulatory system \ </tooltip>
            <class>ENC</class>
            <constrain_by_date/>
         </item>
      </panel>
  </query_definition>
```

Element Name	Description	Required
query_name	Name of the query, must be unique	Yes
query_timing	Query timing is used only in setfinder query, The value could be 'ANY' or 'SAME'. If the value is 'ANY' then CRC selects patients irrespective of their visit/encounter_id and if the value is 'SAME', then CRC selects the patients who have the same visit/encounter_id across the item filter.	No
Specificity_scale	Test true negative cases. Not recommended to set this value.	No
Query_date_from	Apply the observation fact's start date condition at the query level.	No
Query_date_to	Apply the observation fact's end date condition at the query level.	No
Panel	Panel is a concept to group item within them. The set of observation facts for each item filter are unioned at the panel level. Panel has the attribute, "name" which is the key field for	Yes

	the panel and it is unique.	
panel_number	Panel number, just the serial number starting with 1.	Yes
panel_date_from	Apply the observation fact’s start date condition at the panel level.	No
Panel_date_to	Apply the observation fact’s end date condition at the panel level.	No
Invert	The invert value could be “1” or “0”. If this value is “1”, then query applies “NOT” condition for whole panel.	No
total_item_occurrences	Select the events only if the total number of occurrence is greater or equal to this value.	No
Item	Item contains the filter and query building information, like the item key, dimension table column name, data type, etc.	Yes
Hlevel	Hierarchy level, not required for this implementation.	No
Item_name	Name of the item, this is not required element and mostly for UI purposes.	No
Item_table	Dimension table name	No
Item_key	Item key representing the unique path of concepts available in metadata schema or the ontology cell. The format of item_key is [\\Dimension\concept path].	Yes
Tooltip	This is not required element and is mostly for UI purposes.	No
Class	This is not used, but just added to the specification. This is could be used to classify the data, for example whether we need fact’s image data, text data, etc.	No

Constrain_by_value	To constrain the observation value of a concept.
	`<constraint_by_value>`
	`   <value_type>N</value_type>`
	`   <value_operator>GE</value_operator>`
	`   <value_unit_of_measure>`
	`   </value_unit_of_measure>`
	`   <value_constraint>100</value_constraint>`
	`</constraint_by_value>`
	If the observation fact value type is numeric (“N”), the “nval_num” field in the observation_fact table is used in the query. If the fact value type is character(“C”), then the “tval_char” field is used in the query.
		No

constrain_by_date	Apply start and end date constraint at item level.	No
Constrain_by_modifier	Apply modifier constraint at item level.	No

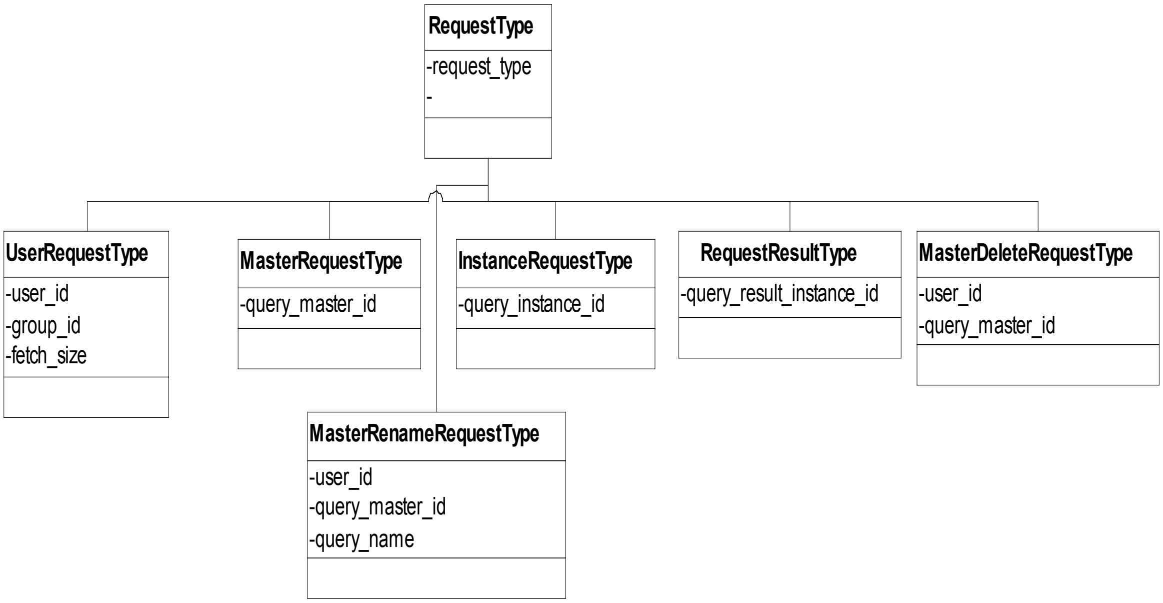
4.3.2.1 Request and Response message structure :

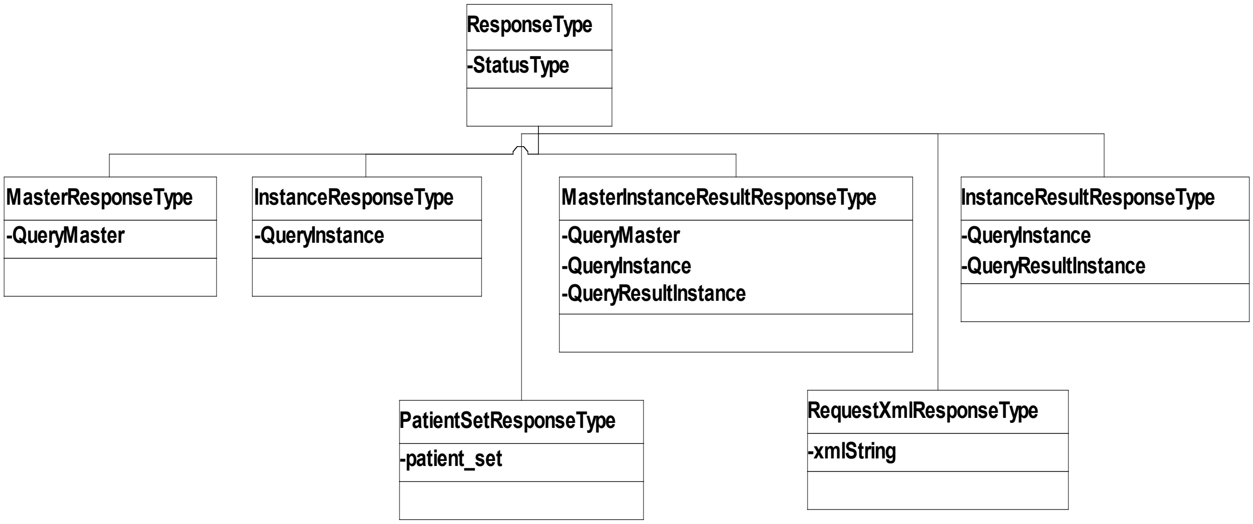
The patient set request/response message structure is divided into three parts: 1. PSMHeader 2. Request and 3. Response. For a request message, the <psmheader> and <request> parts are required; while a response message requires only the <response> part.

```
<i2b2:message_body>
     <crc:psmheader>
        <request_type></request_type>
     </crc:psmheader>

     <crc:request>
           …

     </crc:request>

     <crc:response>
           …

     </crc:response>
</i2b2:message_body>
```

Element name	Description
<psmheader>	Header has a <request_type> element, which will carry operation name. Each operation name have a specific request/response combination. Following are list of supported operation names: CRC_QRY_getRequestXml_fromQueryMasterId CRC_QRY_getQueryMasterList_fromUserId CRC_QRY_runQueryInstance_fromQueryDefinition CRC_QRY_getQueryMasterList_fromGroupId CRC_QRY_getQueryResultInstanceList_fromQueryInstanceId CRC_QRY_getQueryInstanceList_fromQueryMasterId CRC_QRY_deleteQueryMaster CRC_QRY_renameQueryMaster CRC_QRY_runQueryInstance_fromQueryMasterId
<request>	The <request> is modeled as an object using a polymorphic approach. All operation specific request objects inherit a base RequestType object containing a request_type attribute as shown in section 4.3.3.
<response>	The <response> is also modeled as an object using a polymorphic approach. All operation specific response objects inherit a base ResponseType object containing a StatusType attribute as shown in section 4.3.3.

4.3.3 Request and Response Object Model





The following chart shows the different request and response types for each service type listed above. The RequestType column describes what input is expected and the ResponseType column describes what output is expected.

Operation	RequestType						ResponseType					
	User	Master (Query)	Instance (Query Run)	QueryDefinition	PatientSet	ObservationFact	Master (Query)	Instance (Query Run)	Result	RequestXml	MasterInstanceResult	PatientData
Get a List of Saved Query Definitions	x						x					
Get a List of Saved Query Runs		x						x				
Get a List of Saved Query Results			x						x			
Get XML Definition of a Defined Query		x								x		
Run (New) Patient Set Query				x							x	
Run (Existing) Patient Set Query		x									x	
Get Patient Data From a Patient Set					x							x
Get Patient Data From Observation Fact						x						x

4.3.4 Use Case Scenario:

4.3.4.1 Execute a query and get its results.

The server saves the query information under the given user id and group id. The query is executed. Upon completion, a patient set is created with a list of patients who satisfy the query conditions.

The server will read the value <result_waittime_ms> from the <request_header> and if the query did not complete before the wait time specified in the request, it will send a response to the client with “PENDING” status. In this case, the client may later send a query instance request to see if the query is completed and get the query result information.

Request Type	Request	Response
CRC_QRY_getQueryMasterList_fromUserId	user_requestType	master_responseType

Example :

```
<request_header>
       <result_waittime_ms>90000</result_waittime_ms>
</request_header>

<message_body>
    <crc:psmheader>
        <request_type>
              CRC_QRY_runQueryInstance_fromQueryDefinition
        </request_type>
    </crc:psmheader>

    <crc:request xsi:type="crc:query_definition_requestType">
        <query_definition>
            <query_name/>
            <query_description/>
            <query_timing>SAME</query_timing>
            <specificity_scale>0</specificity_scale>
            <query_date_from>2000-12-30T00:00:00</query_date_from>
            <query_date_to>2000-12-30T00:00:00</query_date_to>
            <panel>
                <panel_number>0</panel_number>
                <panel_date_from>2000-12-30T00:00:00</panel_date_from>
                <panel_date_to>2000-12-30T00:00:00</panel_date_to>
                <invert>0</invert>
                <total_item_occurrences>0</total_item_occurrences>
                <item>
                    <hlevel>0</hlevel>
                    <item_name/>
                    <item_table/>
                    <item_key/>
                    <item_icon/>
```

```
                        <tooltip/>
                        <class/>
                    </item>
                </panel>
            </query_definition>
        </crc:request>

        <crc:response xsi:type="crc:master_instance_result_responseType">
            <query_master>
                <query_master_id>0</query_master_id>
                <name/>
                <user_id/>
                <group_id/>
                <create_date>2000-12-30T00:00:00</create_date>
                <request_xml/>
            </query_master>
            <query_instance>
                <query_instance_id>0</query_instance_id>
                <query_master_id>0</query_master_id>
                <user_id/>
                <group_id/>
                <batch_mode/>
                <start_date>2000-12-30T00:00:00</start_date>
                <end_date>2000-12-30T00:00:00</end_date>
                <query_status_type>
                    <status_type_id>6</status_type_id>
                    <name>COMPLETED</name>
                    <description/>
                </query_status_type>
            </query_instance>
            <query_result_instance>
                <result_instance_id>0</result_instance_id>
                <query_instance_id>0</query_instance_id>
                <query_result_type>
                    <result_type_id>1</result_type_id>
                    <name>PATIENTSET</name>
                    <description/>
                </query_result_type>
                <set_size>0</set_size>
                <start_date>2000-12-30T00:00:00</start_date>
                <end_date>2000-12-30T00:00:00</end_date>
                <query_status_type>
                    <status_type_id>3</status_type_id>
                    <name>FINISHED</name>
                    <description/>
                </query_status_type>
            </query_result_instance>
        </crc:response>
    </message_body>
```

4.3.4.2 Scenario : Check if the query is completed and get its results

This request provides a query_instance_id and returns associated result information if the query is in completed state.

Request Type	Request	Response
CRC_QRY_getQueryResultInstanceList_fromQueryInstanceId	instance_requestType	result_responseType

Example:

```
<message_body>
   <psmheader>
        <user login="demo">demo</user>
        <patient_set_limit>0</patient_set_limit>
        <estimated_time>0</estimated_time>
        <request_type>
              CRC_QRY_getQueryResultInstanceList_fromQueryInstanceId
        </request_type>
  </psmheader>

  <request xsi:type="ns4:instance_requestType"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <query_instance_id>6280</query_instance_id>
  </request>

  <response xsi:type="ns5:result_responseType"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
        <query_result_instance>
            <result_instance_id>6280</result_instance_id>
            <query_instance_id>6280</query_instance_id>
            <query_result_type>
                <result_type_id>1</result_type_id>
                <name>PATIENTSET</name>
            </query_result_type>
            <set_size>2000</set_size>
            <start_date>2007-09-06T10:42:14.000-04:00</start_date>
            <end_date>2007-09-06T10:42:15.000-04:00</end_date>
            <query_status_type>
                <status_type_id>3</status_type_id>
                <name>FINISHED</name>
            </query_status_type>
        </query_result_instance>
    </response>
  </message_body>
```

4.3.4.3 Scenario : Get a list of queries by user id.

This request fetches a list of query master information for the given user id. The client can also specify how many query master items to return from the server using the <fetch_size> element. The server returns query master items in descending order of query creation time.

Request Type	Request	Response
CRC_QRY_getQueryMasterList_fromUserId	user_requestType	master_responseType

Example :

```
<message_body>
     <psmheader>
           <request_type>
                 CRC_QRY_getQueryMasterList_fromUserId
           </request_type>
     </psmheader>

     <request xsi:type="ns3:user_requestType"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
           <user_id>user1</user_id>
           <fetch_size>100</fetch_size>
     </request>

     <response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:type="ns5:master_responseType">
           <status>
               <condition type="DONE">DONE</condition>
           </status>
           <query_master>
               <query_master_id>6302</query_master_id>
               <name>  1 y-Femal-Rheum@10:17:55</name>
               <user_id>demo</user_id>
               <group_id>Asthma</group_id>
               <create_date>2007-09-06T22:17:57.000-04:00</create_date>
           </query_master>
           <query_master>
               <query_master_id>6301</query_master_id>
               <name>  10 ye-Female@10:42:41</name>
               <user_id>demo</user_id>
               <group_id>Asthma</group_id>
               <create_date>2007-09-06T10:42:42.000-04:00</create_date>
           </query_master>
    </response>
</message_body>
```

4.3.4.4 Scenario: Get query definition from master id

This request will return <query_definition> information for the given query master id.

Request Type	Request	Response
CRC_QRY_getRequestXml_fromQueryMasterId	master_requestType	request_xml_responseType

Example :

```
 <message_body>
   <psmheader>
       <request_type>
             CRC_QRY_getRequestXml_fromQueryMasterId
       </request_type>
   </psmheader>

   <request xsi:type="ns4:master_requestType"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <query_master_id>6300</query_master_id>
    </request>

    <response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:type="request_xml_responseType ">
       <status>
           <condition type="DONE">DONE</condition>
       </status>
       <request_xml><![CDATA[
           <query_definition>
               <query_name/>
               <query_description/>
               <query_timing>ANY</query_timing>
               <specificity_scale>0</specificity_scale>
               <query_date_from>2000-12-30T00:00:00</query_date_from>
               <query_date_to>2000-12-30T00:00:00</query_date_to>
               <panel>
                   <panel_number>0</panel_number>
                   <panel_date_from>
                         2000-12-30T00:00:00
                   </panel_date_from>
                   <panel_date_to>2000-12-30T00:00:00</panel_date_to>
                   <invert>0</invert>
                   <total_item_occurrences>0</total_item_occurrences>
                   <item>
                       <hlevel>0</hlevel>
                       <item_name/>
                       <item_table/>
                       <item_key/>
                       <item_icon/>
                       <tooltip/>
                       <class/>
                   </item>
               </panel>
           </query_definition>]]>
         </request_xml>
     </response>
</message_body>
```

4.3.4.5 Scenario : Rename a query

Use this request to change the name of the query. If the same user already has the query with the specified name, then the server will return error in the <status> tag.

Request Type	Request	Response
CRC_QRY_renameQueryMaster	master_rename_requestType	master_responseType

Example:

```
 <message_body>
    <psmheader>
       <user login="demo">demo</user>
       <patient_set_limit>0</patient_set_limit>
       <estimated_time>0</estimated_time>
       <request_type>CRC_QRY_renameQueryMaster</request_type>
   </psmheader>

   <request xsi:type="ns4:master_rename_requestType"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <user_id>demo</user_id>
       <query_master_id>5997</query_master_id>
       <query_name>Demographics@03:21:10 -n[07-20-2007 ]</query_name>
   </request>

   <response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:type="ns5:master_responseType">
       <status>
           <condition type="DONE">DONE</condition>
       </status>
       <query_master>
           <query_master_id>5997</query_master_id>
           <name>Demographics@03:21:10 -n[07-20-2007 ]</name>
           <user_id>demo</user_id>
       </query_master>
   </response>
</message_body>
```

4.3.4.6 Scenario : Delete a query

Use this request to remove a query and its results. Delete will not permanently remove the query; it will just set the delete flag to true.

Request Type	Request	Response
CRC_QRY_deleteQueryMaster	master_delete_requestType	master_responseType

Example:

```
    <message_body>
       <psmheader>
            <user login="demo">demo</user>
            <patient_set_limit>0</patient_set_limit>
            <estimated_time>0</estimated_time>
            <request_type>CRC_QRY_deleteQueryMaster</request_type>
       </psmheader>

       <request xsi:type="ns4:master_delete_requestType"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <user_id>demo</user_id>
            <query_master_id>5997</query_master_id>
        </request>

      <response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="ns5:master_responseType">
            <status>
                <condition type="DONE">DONE</condition>
            </status>
            <query_master>
                <query_master_id>5997</query_master_id>
            </query_master>
       </response>
    </message_body>
```

4.4 Patient Data Object Query Service:

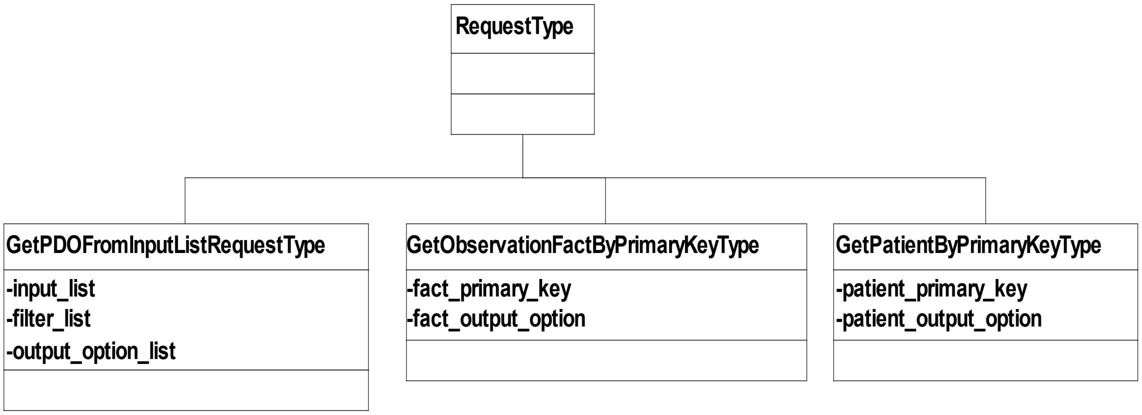
As the name suggests, these queries return Patient data objects (PDO) in the response message as specified by the request message's patient set and the filter criteria. The message structure divided into three parts: 1. PdoHeader, 2. Request and 3. Response. For the request message, the <pdoheader> and <request> parts are required, while for the response message, only the <response> part is required.

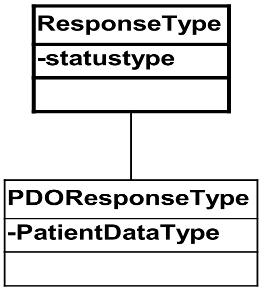
```
<i2b2:message_body>
   <crc:pdoheader>
       <request_type>GetPDOFromInputList_requestType</request_type>
        . . .
   </crc:pdoheader>

   <crc:request>
      . . .
   </crc:request>

   <crc:response>
      . . .
   </crc:response>
</i2b2:message_body>
```

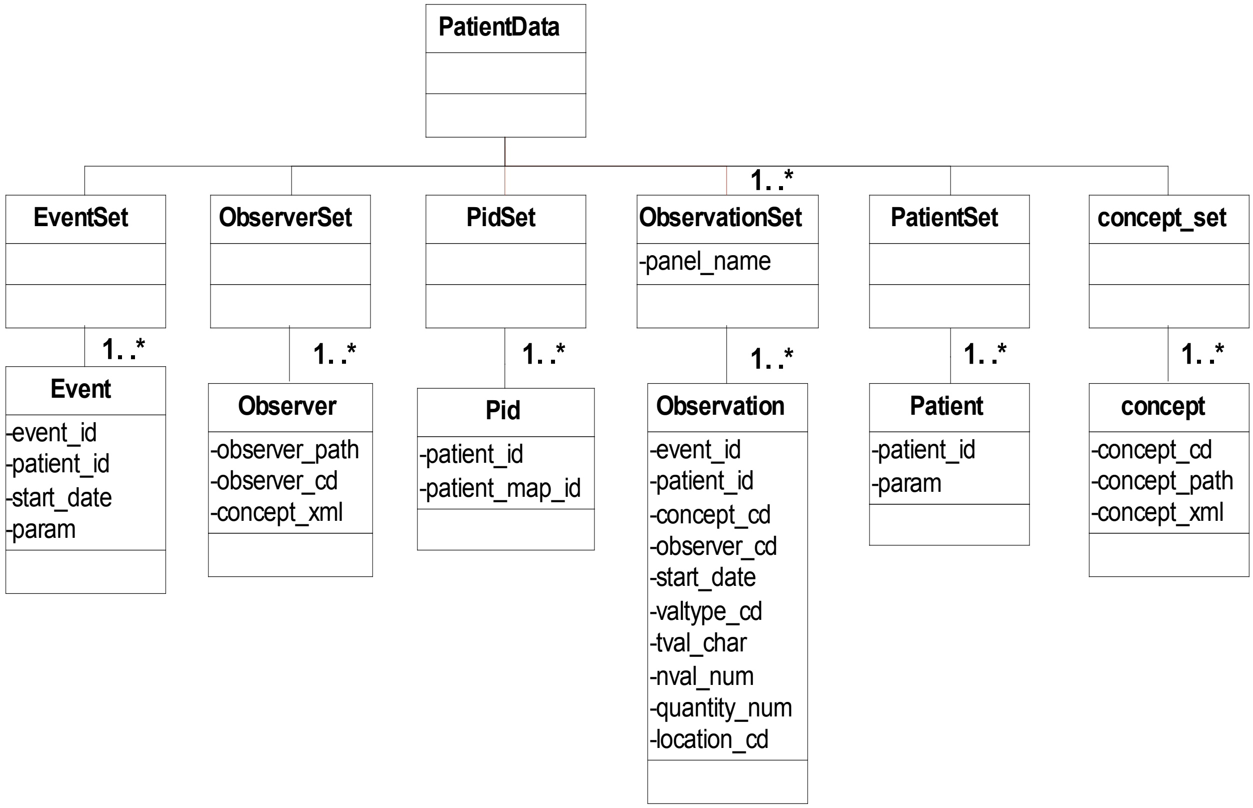
4.4.1 Request and Response message structure :

Element name	Description
<pdoheader>	Header contains a <request_type> element, which will carry operation name. Each operation name has a specific request/response combination. Following are list of supported operation names: getPDO_fromInputList get_observationfact_by_primary_key get_patient_by_primary_key get_event_by_primary_key get_concept_by_primary_key get_observer_by_primary_key
<request>	The <request> is modeled as an object using a polymorphic approach. All operation specific request objects inherit a base RequestType object as shown in section 4.4.2.
<response>	The <response> is also modeled as an object using a polymorphic approach. All operation specific response objects inherit a base PDOResponseType object containing a StatusType attribute and a PatientData object as shown in section 4.3.3.

4.4.2 Request and Response Object Model







4.4.3 Use Case Scenario

4.4.3.1 Scenario: Get patient data from a patient set id.

This request divided into three parts: an input_list, a filter_list and an output_option. The input_list accepts either the id of the patient set or a list of patient ids. The filter_list holds a list of panels. Panels in turn have item details which are used in constructing a PDO query. And finally the output_option specifies which set of patient data to return. Each of the patient data sections in output_option has attributes to specify the level of detail data expected in the response.

Request Type	Request	Response
`getPDO_fromInputList`	GetPDOFromInputList_requestType	master_responseType

Filter List Type :

Element Name	Description	Required
Query_name	Query name, used in setfinder query not used for PDO query	No
Query_Description	Query description, used in setfinder query not used for PDO query	No
Query_timing	Query timing is used only in setfinder query, The value could be 'ANY' or 'SAME'. If the value is 'ANY' then select patients irrespective of their visit/encounter_id and if the value is 'SAME', then selects the patients who have the same visit/encounter_id across the item filter.	No
Panel	Panel is a concept to group item within them and its entire item filters are 'UNION' . huh?? The attribute "name" is the key field for the panel and it is unique.	Yes
Invert	The invert value could be "1" or "0". If this value is "1", then query applies "NOT" condition for whole panel.	No
panel_accuracy_scale/ total_item_occurrences	Select the events only if the total number of occurrence is greater or equal to this value.	No
start_date	Apply the observation fact's start date condition at the panel level.	No
end_date	Apply the observation fact's end data condition at the panel level.	No
Item	Item contains the filter and query building information, like the item key, dimension tables column name, data type, etc.	Yes
Item_name	Name of the item, this is not required element and mostly for UI purpose.	No
Item_key	Item key represent the unique path of concepts available in metadata schema or the ontology cell. The format of item_key is [\\Dimension\concept path].	No
Item_icon	This is not required element and mostly for UI purpose.	No
Item_tooltip	This is not required element and mostly for UI purpose.	No
Class	This is not used, but just added to the specification. This is could be used to classify the data, for example whether we need fact's image data, text data, etc.	No
Start_date	Apply the observation start date constraint for the item level	No
End_date	Apply the observation end date constraint for	No

	the item level.	
Dim_tablename	Name of the filter dimension table in the database schema. This information is used to construct dimension filter SQL. For example: select * from observation_fact where concept_cd in (select concept_cd from concept_dimension where concept_path like '\Some concept path%')	Yes
Dim_columnname	Column name of the filter dimension table	Yes
Dim_dimcode	This is the value of concept/dimension code.	Yes
Dim_columndatatype	Data type of the filter column	Yes
Dim_operator	The conditional operator which can be used for filtering. For example 'LIKE', 'LE', 'GE', 'EQ'.	Yes
facttablecolumn	This is observation fact's column name to apply to the dimension filter.	Yes

Value_constraint	To constrain the observation value of a concept.
	`<constraint_by_value>`
	`<value_type>value_type0</value_type>`
	`<value_operator>value_operator0</value_operator>`
	`<value_unit_of_measure>`
	`</value_unit_of_measure>`
	`<value_constraint></value_constraint>`
	`</constraint_by_value>`
	If the observation fact value type is numeric, "nval_num" field in the observation_fact table is used in the query. If the fact value type is "character", then "tval_char" is used in the query
		No

Output Option Type :

Element Name	Description
Patient_set	Return the set of Patient dimension data either for the given patient list or for the patient present in the observation set.
Concept_set	Return the set of concept section data of a patient data object
Observation_set	Return the observation set of the patient data object. There could be a multiple number of <observation_set> returned and the number of <observation_set> returned will be equal to number of panel defined in the filter list. Observation set has the attribute "panel_name" which corresponds to "name" attribute defined in the <panel>.
Event_set	Return the set containing event/visit dimension data occurring in the observation set
Observer_set_using_filter_list	Return the set containing observer/provider

	dimension data occurring in the observation set
Concept_set_using_filter_list	Return the set of concept dimension data occurring in the observation set

Example:

```
    <message_body>

        <crc:pdoheader>
            <request_type>getPDO_fromInputList</request_type>
        </crc:pdoheader>

        <crc:request xsi:type="ns2:GetPDOFromInputList_requestType"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

            <!- - input list - - >
            <input_list>
                  <patient_list max="300" min="0">
                   <patient_set_coll_id>100</patient_set_coll_id>
                   </patient_list>
             </input_list>

            <!- - filter  list - - >
            <filter_list>
               <panel name="panel1">
                  <panel_invert>0</panel_invert>
                  <panel_accuracy_scale></panel_accuracy_scale>
                  <panel_start_date><panel_start_date>
                  <panel_end_date></panel_end_date>
                  <item>
                        <item_name> </item_name>
                        <item_key> </item_key>
                        <item_icon> </item_icon>
                        <item_tooltip></item_tooltip>
                        <dim_tablename> </dim_tablename>
                        <dim_columnname> </dim_columnname>
                        <dim_dimcode></dim_dimcode>
                        <dim_columndatatype> </dim_columndatatype>
                        <dim_operator> </dim_operator>
                        <facttablecolumn></facttablecolumn>
                        <value_constraint>
                              <value_type></value_type>
                              <value_operator></value_operator>
                              <value_unitofmeasure></value_unitofmeasure>
                              <value></value>
                       </value_constraint>
                  </item>
                  . . .
              </panel>

               <panel name="panel2">
                  <panel_invert>0</panel_invert>
                  <panel_accuracy_scale></panel_accuracy_scale>
                  <panel_start_date><panel_start_date>
                  <panel_end_date></panel_end_date>
```

```
          <item>
                <item_name> </item_name>
                <item_key> </item_key>
                <item_icon> </item_icon>
                <item_tooltip></item_tooltip>
                <dim_tablename> </dim_tablename>
                <dim_columnname> </dim_columnname>
                <dim_dimcode></dim_dimcode>
                <dim_columndatatype> </dim_columndatatype>
                <dim_operator> </dim_operator>
                <facttablecolumn></facttablecolumn>
                <value_constraint>
                      <value_type></value_type>
                      <value_operator></value_operator>
                      <value_unitofmeasure></value_unitofmeasure>
                      <value></value>
               </value_constraint>
          </item>
      </panel>
      . . .
    </filter_list>


    <!-- output options -- >
    <output_option>
        <patient_set select="using_fact_list" onlykeys="true"/>
        <concept_set select="using_fact_list" onlykeys="true"/>
        <observation_set blob="false" onlykeys="false"/>
        <event_set select="using_fact_list" onlykeys="true"/>
        <observer_set_using_filter_list onlykeys="true"/>
        <! — To specify generalized dimension type - ->
        <dimension_set_using_filter_list dimensionname="dimension1"
                onlykeys="true"/>
    </output_option>
</crc:request>

<! - - response begin -- >
<response>
    <patient_data>
      <!- - patient set section begins →
       <patient_set>
          <patient>
                <patient_id>patient_id6</patient_id>
          </patient>
          . .
        <patient_set>

       <!- - concept set section begins →
        <concept_set>
           <concept>
             <concept_path>concept_path0</concept_path>
             <concept_cd>concept_cd0</concept_cd>
             <name_char>name_char0</name_char>
           </concept>
          . .
         </concept_set>
      <!- - observation set section begins →
      <observation_set panel_name=”panel1”>
          <observation>
             <event_id source="source3">event_id3</event_id>
             <patient_id>patient_id9</patient_id>
             <concept_cd name="name0">concept_cd3</concept_cd>
             <observer_cd soruce="soruce0">observer_cd3</observer_cd>
             <start_date>2006-05-04T18:13:51.0Z</start_date>
             <modifier_cd name="name1">modifier_cd0</modifier_cd>
             <valuetype_cd>valuetype_cd0</valuetype_cd>
             <tval_char>tval_char0</tval_char>
             <nval_num units="units0">3.141592653589</nval_num>
             <valueflag_cd name="name2">valueflag_cd0</valueflag_cd>
             <quantity_num>3.141592653589</quantity_num>
             <units_cd>units_cd0</units_cd>
             <end_date>2006-05-04T18:13:51.0Z</end_date>
             <location_cd name="name3">location_cd0</location_cd>
          </observation>
          <observation>
          ‥
          </observation>
    </observation_set>

    <observation_set panel_name=”panel2”>
          <observation>
           ..
          </observation>
          <observation>
           ..
          </observation>
          ..
     </observation_set>

     <!- - event set section begins -- >
     <event_set>
          <event>
            <event_id source="source0">event_id0</event_id>
            <patient_id>patient_id0</patient_id>
            <start_date>2006-05-04T18:13:51.0Z</start_date>
            <end_date>2006-05-04T18:13:51.0Z</end_date>
          </event>
          . .
     </event_set>

    <!- - observer/provider set section begins -- >
     <observer_set>
          <observer>
             <observer_path>observer_path0</observer_path>
             <observer_cd>observer_cd0</observer_cd>
             <name_char>name_char3</name_char>
          </observer>
          ..
          </observer_set>

  </crc:patient_data>
```

```
   </response>
    <!- - response end -- >
</message_body>
```

4.4.3.2 Scenario: Get Observation blob by primary key.

This request returns observation blob using observation primary key

Example:

```
 <message_body>
     <pdoheader>
         <request_type>get_observationfact_by_primary_key</request_type>
   </pdoheader>

   <ns5:request
       xsi:type="ns5:GetObservationFactByPrimaryKey_requestType"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <fact_primary_key>
           <event_id>2004005981</event_id>
           <patient_id>52003</patient_id>
           <concept_cd>LCS-I2B2:c1009c</concept_cd>
           <observer_id>03840261</observer_id>
           <start_date>1995-08-24T00:00:00.179-05:00</start_date>
       </fact_primary_key>
       <fact_output_option select="using_filter_list" onlykeys="false"/>
   </ns5:request>

   <response>
       <patient_data>
           <observation_set>
             <observation>
                <event_id source="source3">event_id3</event_id>
                <patient_id>patient_id9</patient_id>
                <concept_cd name="name0">concept_cd3</concept_cd>
                <observer_cd soruce="soruce0">observer_cd3</observer_cd>
                <start_date>2006-05-04T18:13:51.0Z</start_date>
                 <observation_blob><![CDATA[
                   patient notes]]>
                 </observation_blob>
             </observation >
        </observation_set>
       </patient_data>
     </response>
</message_body>
```

4.5 Message Explanations

This section defines message elements in the CRC namespace (http://www.i2b2.org/xsd/cell/crc/psm/1.1/ and http://www.i2b2.org/xsd/cell/crc/pdo/1.1/) Each element defined will have an implied prefix of crc: unless another namespace is explicitly stated. Elements from other namespaces, which are included within CRC elements, will be listed but not expanded or defined in this document. Refer to other cell documents to get specific details on those elements.

4.5.1 Header

The <header> is the first CRC element within an i2b2 <message_body>. This section defines the elements shown in the example header shown, below.

```
<crc:header>
      <user login=""></user>
      <data_source></data_source>
      <patient_set_limit>0</patient_set_limit>
      <estimated_time>0</estimated_time>
      <create_date>2002-12-23T00:00:00</create_date>
      <submit_date>2002-12-23T00:00:00</submit_date>
      <complete_date>2002-12-23T00:00:00</complete_date>
      <request_type>getPDO_fromObservationFact</request_type>
</crc:header>
```

header: container for generic information useful for any crc message

user: user information used for authentication and login

data_source: information about the source of the data

patient_set_limit: limit the size of the patient set returned in a query

estimated_time: the time estimated for the query to complete

create_date: the date that a query was created

submit_date: the date that a query was submitted to be executed or run

complete_date: the date that a query finished executing

request_type: a code that tells the service what type of request to expect, which tells it what kind of xml to expect in the rest of the message.

4.5.2 Request

4.5.2.1 xsi:type="crc:user_requestType"

```
<crc:request xsi:type="crc:user_requestType">
      <user_id>some_user_id</user_id>
      <group_id>some_group_id</group_id>
      <fetch_size>5000</fetch_size>
  </crc:request>
```

4.5.2.2 xsi:type="crc:master_requestType"

```
<crc:request xsi:type="crc:master_requestType">
      <query_master_id>0</query_master_id>
  </crc:request>
```

4.5.2.3 xsi:type="crc:instance_requestType"

```
<crc:request xsi:type="crc:instance_requestType">
      <query_instance_id>0</query_instance_id>
  </crc:request>
```

4.5.2.4 xsi:type="crc:query_definition_requestType"

```
<crc:request xsi:type="crc:query_definition_requestType">
      <query_definition>
          <query_name/>
          <query_description/>
          <query_timing>SAME</query_timing>
          <specificity_scale>0</specificity_scale>
          <query_date_from>2000-12-30T00:00:00</query_date_from>
          <query_date_to>2000-12-30T00:00:00</query_date_to>
          <panel>
              <panel_number>0</panel_number>
              <panel_date_from>2000-12-30T00:00:00</panel_date_from>
              <panel_date_to>2000-12-30T00:00:00</panel_date_to>
              <invert>0</invert>
              <total_item_occurrences>0</total_item_occurrences>
              <item>
                  <hlevel>0</hlevel>
                  <item_name/>
                  <item_table/>
                  <item_key/>
                  <item_icon/>
                  <tooltip/>
                  <class/>
              </item>
          </panel>
      </query_definition>
  </crc:request>
```

4.5.2.5 xsi:type="crc:getPDO_fromInputList "

```
      <crc:request xsi:type="crc:patient_set_requestType">
            <select_option_list>
                <observation_fact blob="true" before="2005-12-30T00:00:00"
after="2003-12-30T00:00:00"/>
                <patient_dimension fact_related="false"/>
                <provider_dimension/>
                <visit_dimension detail="false"/>
                <concept_dimension status="true"/>
            </select_option_list>
            <filter_list>
               <panel name="panel1">
                  <panel_invert>0</panel_invert>
                  <panel_accuracy_scale></panel_accuracy_scale>
                  <panel_start_date><panel_start_date>
                  <panel_end_date></panel_end_date>
                  <item>
                        <item_name> </item_name>
                        <item_key> </item_key>
                        <item_icon> </item_icon>
                        <item_tooltip></item_tooltip>
                        <dim_tablename> </dim_tablename>
                        <dim_columnname> </dim_columnname>
                        <dim_dimcode></dim_dimcode>
                        <dim_columndatatype> </dim_columndatatype>
                        <dim_operator> </dim_operator>
                        <facttablecolumn></facttablecolumn>
                        <value_constraint>
                              <value_type></value_type>
                              <value_operator></value_operator>
                              <value_unitofmeasure></value_unitofmeasure>
                              <value></value>
                       </value_constraint>
                  </item>
                  . . .
              </panel>
            </filter_list>
            <patient_list min="1" max="10">
                <patient_num index="1">50</patient_num>
                <patient_num index="2">24</patient_num>
                <patient_num index="3">78</patient_num>
                <!--
                    <entire_patient_set/>
                    <patient_set_coll_id>0</patient_set_coll_id>
                 -->
            </patient_list>
        </crc:request>
```

4.5.2.6 xsi:type="crc: GetObservationFactByPrimaryKey_requestType "

```
<crc:request xsi:type="crc:observation_fact_set_requestType">
      </event_id>
      </patient_id>
      </concept_cd/>
      </observer_id/>
      </start_date/>
      </modifier_cd/>
  </crc:request>
```

4.5.3 Response

4.5.3.1 xsi:type="crc:master_responseType"

```
<crc:response xsi:type="crc:master_responseType">
      <query_master>
          <query_master_id>0</query_master_id>
          <name/>
          <user_id/>
          <group_id/>
          <create_date>2000-12-30T00:00:00</create_date>
          <delete_date>2000-12-30T00:00:00</delete_date>
          <request_xml/>
          <generated_sql/>
      </query_master>
      <query_master>
          <query_master_id>1</query_master_id>
          <name/>
          <user_id/>
          <group_id/>
          <create_date>2000-12-30T00:00:00</create_date>
          <delete_date>2000-12-30T00:00:00</delete_date>
          <request_xml/>
          <generated_sql/>
      </query_master>
  </crc:response>
```

4.5.3.2 xsi:type="crc:instance_responseType"

```
<crc:response xsi:type="crc:instance_responseType">
      <query_instance>
          <query_instance_id>0</query_instance_id>
          <query_master_id>0</query_master_id>
          <user_id/>
          <group_id/>
          <batch_mode/>
          <start_date>2000-12-30T00:00:00</start_date>
          <end_date>2000-12-30T00:00:00</end_date>
          <query_status_type>
              <status_type_id>0</status_type_id>
              <name>finished</name>
              <description/>
          </query_status_type>
      </query_instance>
      <query_instance>
          <query_instance_id>1</query_instance_id>
          <query_master_id>0</query_master_id>
          <user_id/>
          <group_id/>
          <batch_mode/>
          <start_date>2000-12-30T00:00:00</start_date>
          <end_date>2000-12-30T00:00:00</end_date>
          <query_status_type>
              <status_type_id>0</status_type_id>
              <name>finished</name>
              <description/>
          </query_status_type>
      </query_instance>
  </crc:response>
```

4.5.3.3 xsi:type="crc:result_responseType"

```
<crc:response xsi:type="crc:result_responseType">
      <query_result_instance>
          <result_instance_id>0</result_instance_id>
          <query_instance_id>0</query_instance_id>
          <query_result_type>
              <result_type_id>0</result_type_id>
              <name>PATIENT_SET</name>
              <description/>
          </query_result_type>
          <set_size>0</set_size>
          <start_date>2000-12-30T00:00:00</start_date>
          <end_date>2000-12-30T00:00:00</end_date>
          <query_status_type>
              <status_type_id>0</status_type_id>
              <name>finished</name>
              <description/>
          </query_status_type>
      </query_result_instance>
      <query_result_instance>
          <result_instance_id>1</result_instance_id>
          <query_instance_id>0</query_instance_id>
          <query_result_type>
              <result_type_id>1</result_type_id>
              <name>ENCOUNTER_SET</name>
              <description/>
          </query_result_type>
          <set_size>0</set_size>
          <start_date>2000-12-30T00:00:00</start_date>
          <end_date>2000-12-30T00:00:00</end_date>
          <query_status_type>
              <status_type_id>0</status_type_id>
              <name>finished</name>
              <description/>
          </query_status_type>
      </query_result_instance>
  </crc:response>
```

4.5.3.4 xsi:type="crc:request_xml_responseType"

```
      <crc:response xsi:type="crc:request_xml_responseType">
            <xml_string><![CDATA[
                <query_definition>
                    <query_name/>
                    <query_description/>
                    <query_timing>ANY</query_timing>
                    <specificity_scale>0</specificity_scale>
                    <query_date_from>2000-12-30T00:00:00</query_date_from>
                    <query_date_to>2000-12-30T00:00:00</query_date_to>
                    <panel>
                        <panel_number>0</panel_number>
                        <panel_date_from>2000-12-
30T00:00:00</panel_date_from>
                        <panel_date_to>2000-12-30T00:00:00</panel_date_to>
                        <invert>0</invert>
                        <total_item_occurrences>0</total_item_occurrences>
                        <item>
                            <hlevel>0</hlevel>
                            <item_name/>
                            <item_table/>
                            <item_key/>
                            <item_icon/>
                            <tooltip/>
                            <class/>
                        </item>
                    </panel>
                </query_definition>
            ]]></xml_string>
        </crc:response>
```

4.5.3.5 xsi:type="crc:master_instance_result_responseType"

```
<crc:response xsi:type="crc:master_instance_result_responseType">
      <query_master>
          <query_master_id>0</query_master_id>
          <name/>
          <user_id/>
          <group_id/>
          <create_date>2000-12-30T00:00:00</create_date>
          <delete_date>2000-12-30T00:00:00</delete_date>
          <request_xml/>
          <generated_sql/>
      </query_master>
      <query_instance>
          <query_instance_id>0</query_instance_id>
          <query_master_id>0</query_master_id>
          <user_id/>
          <group_id/>
          <batch_mode/>
          <start_date>2000-12-30T00:00:00</start_date>
          <end_date>2000-12-30T00:00:00</end_date>
          <query_status_type>
              <status_type_id>0</status_type_id>
              <name>finished</name>
              <description/>
          </query_status_type>
      </query_instance>
      <query_result_instance>
          <result_instance_id>0</result_instance_id>
          <query_instance_id>0</query_instance_id>
          <query_result_type>
              <result_type_id>0</result_type_id>
              <name>PATIENT_SET</name>
              <description/>
          </query_result_type>
          <set_size>0</set_size>
          <start_date>2000-12-30T00:00:00</start_date>
          <end_date>2000-12-30T00:00:00</end_date>
          <query_status_type>
              <status_type_id>0</status_type_id>
              <name>finished</name>
              <description/>
          </query_status_type>
      </query_result_instance>
      <query_result_instance>
          <result_instance_id>1</result_instance_id>
          <query_instance_id>0</query_instance_id>
          <query_result_type>
              <result_type_id>1</result_type_id>
              <name>ENCOUNTER_SET</name>
              <description/>
          </query_result_type>
          <set_size>0</set_size>
          <start_date>2000-12-30T00:00:00</start_date>
          <end_date>2000-12-30T00:00:00</end_date>
          <query_status_type>
              <status_type_id>0</status_type_id>
              <name>finished</name>
              <description/>
          </query_status_type>
      </query_result_instance>
  </crc:response>
```

4.5.3.6 xsi:type="crc:instance_result_responseType"

```
<crc:response xsi:type="crc:instance_result_responseType">
      <query_instance>
          <query_instance_id>0</query_instance_id>
          <query_master_id>0</query_master_id>
          <user_id/>
          <group_id/>
          <batch_mode/>
          <start_date>2000-12-30T00:00:00</start_date>
          <end_date>2000-12-30T00:00:00</end_date>
          <query_status_type>
              <status_type_id>0</status_type_id>
              <name>finished</name>
              <description/>
          </query_status_type>
      </query_instance>
      <query_result_instance>
          <result_instance_id>0</result_instance_id>
          <query_instance_id>0</query_instance_id>
```

```
     <query_result_type>
       <result_type_id>0</result_type_id>
       <name>PATIENT_SET</name>
       <description/>
     </query_result_type>
     <set_size>0</set_size>
     <start_date>2000-12-30T00:00:00</start_date>
     <end_date>2000-12-30T00:00:00</end_date>
     <query_status_type>
       <status_type_id>0</status_type_id>
       <name>finished</name>
       <description/>
     </query_status_type>
   </query_result_instance>
   <query_result_instance>
     <result_instance_id>1</result_instance_id>
     <query_instance_id>0</query_instance_id>
     <query_result_type>
       <result_type_id>1</result_type_id>
       <name>ENCOUNTER_SET</name>
       <description/>
     </query_result_type>
     <set_size>0</set_size>
     <start_date>2000-12-30T00:00:00</start_date>
     <end_date>2000-12-30T00:00:00</end_date>
     <query_status_type>
       <status_type_id>0</status_type_id>
       <name>finished</name>
       <description/>
     </query_status_type>
   </query_result_instance>
 </crc:response>
```

4.5.3.7 xsi:type="crc:patient_data_responseType"

```
<crc:response xsi:type="crc:patient_data_responseType">
   <crc:patient_data>
      <!-- see PDO cell messaging document -->
   </crc:patient_data>
 </crc:response>
```

4.6 XML Schema Definitions

The CRC XML schema consists of the following XSD files:

- **CRC.xsd**
  This schema is not used directly to create CRC messages, but is included in other CRC_PDO_QRY.xsd and CRC_PSM_QRY.xsd.

- **CRC_PDO_QRY.xsd**
  This schema is used for validating CRC patient data queries and defines a <crc:header> and <crc:sql> tag.

- **CRC_PDO_QRY_request.xsd**
  This schema is not directly used but is included in CRC_PDO_QRY.xsd.

- **CRC_PDO_QRY_response.xsd**
  This schema is not directly used but is included in CRC_PDO_QRY.xsd.

- **CRC_PSM_OBJ.xsd**
  This schema defines the data objects that hold information about patient set queries.

- **CRC_PSM_QRY.xsd**
  This schema is used for validating CRC patient set queries and defines a <crc:header> and <crc:sql> tag.

- **CRC_PSM_QRY_request.xsd**
  This schema is not directly used but is included in CRC_PSM_QRY.xsd.

- **CRC_PSM_QRY_response.xsd**
  This schema is not directly used but is included in CRC_PSM_QRY.xsd.

- **CRC_PSM_QRY_query_definition.xsd**
  This schema validates the xml format that defines a patient set query.