i2b2 Cell Messaging

i2b2 Common Header

1 Table of Contents

2 Document Version History

This section lists only major document versions and not detailed changes. Versions are indicated by sequential integers.

Date	Version	Description	Author(s)
12/01/2006	1	Initial Version	Kristel Hackett
08/27/2007	1.1	Version 1.1	Lori Phillips

3 Introduction

This document gives an overview of the i2b2 common headers for both request and response XML messages between i2b2 cells in the i2b2 hive.

3.1 The i2b2 Hive

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 goal of i2b2 is to produce a comprehensive set of software tools to enable clinical investigators to collect and manage their project-related research data, including clinical and genomic data; that is, a software suite for the modern clinical research chart. Since different applications from different sources must be able to communicate with each other, a distributed computing model is needed, one that integrates multiple web-based applications in a standardized way.

The i2b2 hive and associated web services are the infrastructure used to create this integration. The hive is comprised of a collection of cells representing unique functional units. Cells in the hive have an array of roles, such as data storage, data analysis, ontology or identity management, natural language processing, and data conversion, derivation or de-identification. Each cell is a self-contained modular application that communicates with other cells via XML web services. A common i2b2 messaging protocol has been defined to enable the cells to interact with each other, sharing business logic, processes and data.

3.2 i2b2 Messaging Overview

All cells in the i2b2 hive communicate using standard, pre-defined i2b2 XML request and response messages.

A request message is sent from a client to a service and contains information, inside the top-level <request> tag, that allows the service to satisfy the request. The <request> tag contains a <message_header>, <request_header> and <message_body> as shown, below, in Figure 1.

The service sends back a response message, inside a top-level <response> tag, which informs the client about the status of the request and may also contain the actual results. The <response> tag contains it's own <message_header>, <response_header> and <message_body> and it may optionally echo the request's <request_header> as shown, below, in Figure 1.
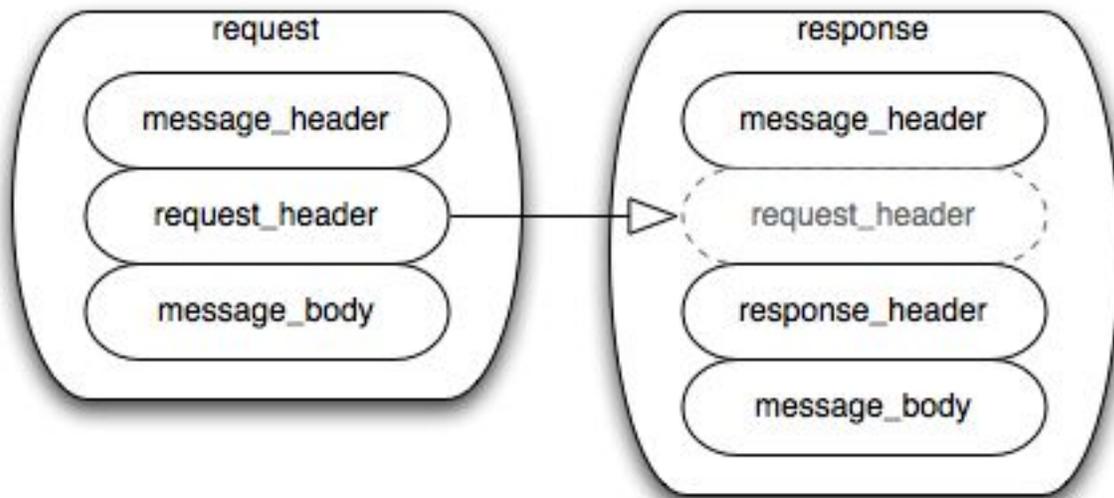
Figure 1: The basic structure of a request and response message. The request_header in the request can be echoed in the response.

3.2.1 Message Header

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. Both request and response messages contain this <message_header> tag which has control information such as sending application, receiving application and message type.

3.2.2 Request Header

The request must contain a <request_header> tag which includes information about how to process a request such as the amount of time it is willing to wait for a response. The <request_header> tag may optionally be echoed back in the response.

3.2.3 Response Header

The response must include a <response_header> tag which includes general information about the response such as status and error messages or where to look for the results if they are not included with the response.

3.2.4 Message Body

Both request and response messages contain a <message_body> tag which may contain any well-formed xml. Individual cells may define cell-specific XML that will be put inside <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.

3.3 i2b2 XML Schema Definitions

The i2b2 XML schema consists of three XSD files:

- **i2b2.xsd**
  This schema defines the type for the <message_header> and <message_body> tags. This schema is included in the i2b2_request.xsd and the i2b2_response.xsd.

- **i2b2_request.xsd**
  This schema defines the type for the top-level <request> tag and the <request_header> tag. It is used for validating i2b2 request messages.

- **i2b2_response.xsd**
  This schema defines the type for the top-level <response> tag and the <response_header> tag. It is used for validating i2b2 response messages.

The i2b2 <message_header>, <request_header> and <response_header> are described in this document. For descriptions of cell-specific content of the <message_body>, please refer to the documents of the specific i2b2 cells.

4 i2b2 Messaging Detail

4.1 Use Cases

All i2b2 cells must satisfy one specific use case which is to communicate with one another. The specific types of communication are defined by each cell. Any client of an i2b2 cell must communicate by sending a request to the cell and the cell responds by sending a response. Generic cell communication is defined in this document while Cell-specific communication is described in each cell's messaging document.
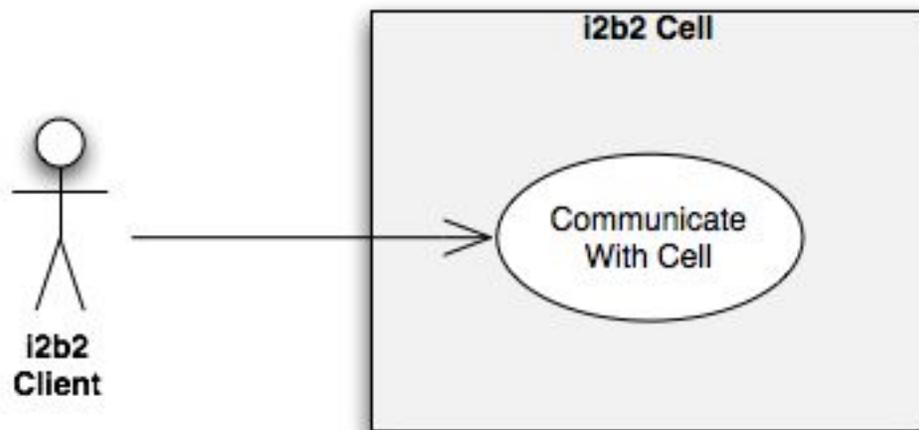

4.2 Services / Messages

Each cell in the i2b2 Hive communicates via web services using xml messages validated by the i2b2_request.xsd and i2b2_response.xsd schemas. The actual services provided by a cell depend on its use case. Cell-specific xml must be placed inside the <message_body> section of the i2b2 request or response. Each cell must describe its services and message sets in the document defining messaging for that cell.

The generic i2b2 request is satisfied by a <request> message which contains a <message_header>, <request_header> and <message_body>.

The generic i2b2 response is satisfied by a <response> message which contains a <message_header>, <response_header> and <message_body>. The response may optionally echo the <request_header> back to the client.

4.3 Sequence Diagrams

All i2b2 cells must satisfy the generic use case of communicating messages. The typical scenario for a communication between any client and any cell is illustrated below. The client sends a <request> message and the cell returns a <response> message. This basic behavior is the same regardless of the outcome or status of the request.
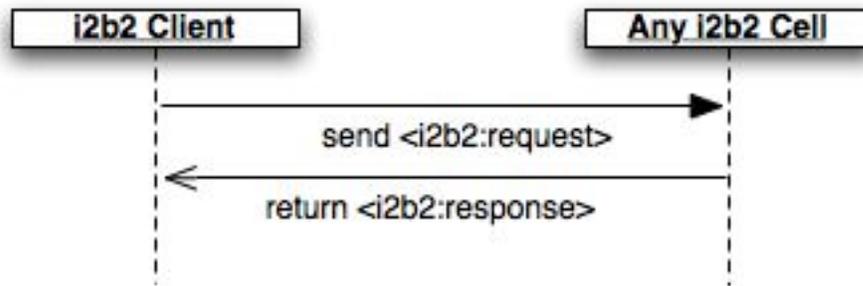
5 Example Messages

5.1 Request Example Message

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:request xmlns:ns4="http://www.i2b2.org/xsd/hive/plugin/"
xmlns:ns2="http://www.i2b2.org/xsd/hive/msg/1.1/">
    <message_header>
        <i2b2_version_compatible>1.1</i2b2_version_compatible>
        <hl7_version_compatible>2.4</hl7_version_compatible>
        <sending_application>
            <application_name>i2b2 Workbench</application_name>
            <application_version>1.0</application_version>
        </sending_application>
        <sending_facility>
            <facility_name>i2b2 Hive</facility_name>
        </sending_facility>
        <receiving_application>
            <application_name>Ontology Cell</application_name>
            <application_version>1.0</application_version>
        </receiving_application>
        <receiving_facility>
            <facility_name>i2b2 Hive</facility_name>
        </receiving_facility>
        <datetime_of_message>2007-07-30T14:55:32.596-
04:00</datetime_of_message>
        <security>
            <domain>Demo</domain>
            <username>demo</username>
            <password>demouser</password>
        </security>
        <message_type>
            <message_code>EQQ</message_code>
            <event_type>Q04</event_type>
            <message_structure>EQQ_Q04</message_structure>
        </message_type>
        <message_control_id>
            <session_id>1000</session_id>
            <message_num>YbN7LqqAfh4MFnBl2ItP</message_num>
            <instance_num>0</instance_num>
        </message_control_id>
        <processing_id>
            <processing_id>P</processing_id>
            <processing_mode>I</processing_mode>
        </processing_id>
        <accept_acknowledgement_type>AL</accept_acknowledgement_type>

<application_acknowledgement_type>AL</application_acknowledgement_type
>
        <country_code>US</country_code>
        <project_id role="manager">Demo</project_id>
```

```
    </message_header>

    </message_header>
    <request_header>
        <result_waittime_ms>120000</result_waittime_ms>
    </request_header>
    <message_body/>
</ns2:request>
```

5.2 Response Example Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ns3:response xmlns:ns3="http://www.i2b2.org/xsd/hive/msg/1.1">
    <message_header>
        <i2b2_version_compatible>1.1</i2b2_version_compatible>
        <hl7_version_compatible>2.4</hl7_version_compatible>
        <sending_application>
            <application_name>Ontology Cell</application_name>
            <application_version>1.0</application_version>
        </sending_application>
        <sending_facility>
            <facility_name>i2b2 Hive</facility_name>
        </sending_facility>
        <receiving_application>
            <application_name>i2b2 Workbench</application_name>
            <application_version>1.2</application_version>
        </receiving_application>
        <receiving_facility>
            <facility_name>i2b2 Hive</facility_name>
        </receiving_facility>
        <datetime_of_message>2007-07-30T15:09:36.355-
04:00</datetime_of_message>
        <security>
            <domain>Demo</domain>
            <username>demo</username>
            <password>demouser</password>
        </security>
        <message_type>
            <message_code>EQQ</message_code>
            <event_type>Q04</event_type>
            <message_structure>EQQ_Q04</message_structure>
        </message_type>
        <message_control_id>
            <session_id>1000</session_id>
            <message_num>YbN7LqqAfh4MFnBl2ItP</message_num>
            <instance_num>1</instance_num>
        </message_control_id>
        <processing_id>
            <processing_id>P</processing_id>
            <processing_mode>I</processing_mode>
        </processing_id>
```

```
            <accept_acknowledgement_type>AL</accept_acknowledgement_type>

<application_acknowledgement_type>AL</application_acknowledgement_type
>
            <country_code>US</country_code>
            <project_id>Demo</project_id>
        </message_header>
        <response_header>
            <info url="http://i2b2/getStatus">Status report</info>
            <result_status>
                <status type="PENDING">Processing your request</status>
                <polling_url interval_ms="100">
                     http://i2b2/polling
                 </polling_url>
                <conditions>
                    <condition type="WARNING" coding_system="i2b2">
                         Message here</condition>
                </conditions>
            </result_status>
        </response_header>
        <message_body/>
</i2b2:response>
```

6 Message Explanations

This section describes, in more detail, elements and attributes in the i2b2 request and response header.

6.1 Request Description

When an i2b2 client sends a XML request message, the i2b2 request message header must be included in the message along with the cell-specific message body. An example request message is shown above with empty cell-specific <message_body> tag.

In general, the request header specifies the necessary information for the receiving cell to process the request. Also, inside the <request_header> tag, the cell can define the maximum waiting time for a request. The receiving cell is guaranteed to send back a response after that waiting time is expired regardless of the process status of the request.

6.2 Response Description

Similar to the request messages, the i2b2 cells construct response XML messages by combing the i2b2 common response header and the cell-specific message body. As shown in the example response XML message above, the response header provides all the necessary data for the original requesting client to process the response message.

Moreover, the <response_header> tag allows the responding cell to indicate the process status of a request in the response message. For example, if an error has occurred, it can send back an error message within the <result status> tag.

6.2.1 Status Type "DONE"

In a <response_header> the status should be set to "DONE" only when the request has finished successfully and all results have been returned via previous responses and the current response.

```
<result_status>
    <status type="DONE">Successfully completed.</status>
    <polling_url interval_ms="50"/>
    <conditions/>
</result_status>
```

6.2.2 Status Type "PENDING"

The cell wrapper may set the status type to "PENDING" and set the polling_url to something that the client may use to poll for results. The cell wrapper does not have to send a pending message back to the client until the initial waiting period has expired. It will continue to send a pending message back on subsequent polls until the status is set to "DONE" or "ERROR".

```
    <result_status>
        <status type="PENDING">Processing your request.</status>
        <polling_url
interval_ms="50">http://i2b2/polling</polling_url>
        <conditions/>
    </result_status>
```

6.2.3 Status Type "ERROR"

If the cell wrapper encounters some error verifying, authenticating, or validating a message, then the cell wrapper will set the status type attribute to "ERROR" and should include a text message inside the status tag with a more detailed error message.

```
    <result_status>
        <status type="ERROR">Your request was not formatted
correctly.</status>
        <polling_url interval_ms="50"/>
        <conditions>
            <condition type="ERROR"
coding_system="i2b2">Missing...</condition>
        </conditions>
    </result_status>
```