i2b2 Cell Messaging

Project Management (PM) Cell

Table of Contents

DOCUMENT MANAGEMENT

Revision Number	Date	Author	Description of change
1.6.1	07/22/10	Janice Donahoe	Created 1.6 version of document.
1.6.2	09/23/11	Mike Mendis	

1. INTRODUCTION

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

1.1 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.

1.2 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 its own <message_header>, <response_header> and <message_body> and it may optionally echo the request's <request_header> as shown, below, in Figure 1.
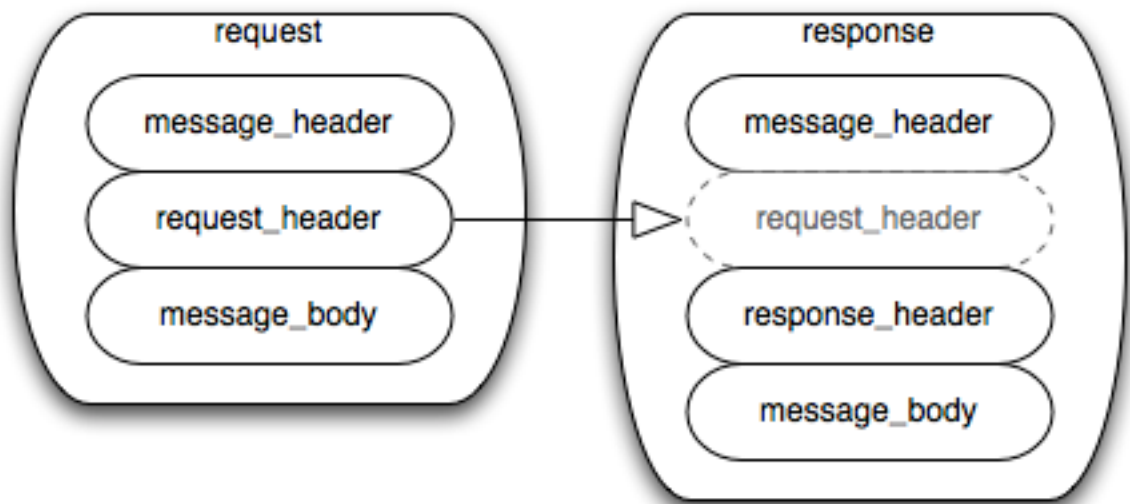
Figure 1: The basic structure of a request and response message. The request_header in the request can be echoed in the response.

1.2.1 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.

1.2.2 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.

1.2.3 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.

1.2.4 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.

1.3   i2b2 XML Schema Definitions

The i2b2 XML schema consists of three XSD files:

1.3.1   i2b2.xsd

This schema defines the type for the <message_header> and <message_body> tags described, in sections 1.2.1 and 1.2.4. This schema is included in the i2b2_request.xsd and the i2b2_response.xsd.

1.3.2   i2b2_request.xsd

This schema defines the type for the top-level <request> tag and the <request_header> tag described in section 0. It is used for validating i2b2 request messages.

1.3.3   i2b2_response.xsd

This schema defines the type for the top-level <response> tag and the <response_header> tag described in section 1.2.3. It is used for validating i2b2 response messages.

More details about the <request>, <response>, <message_header>, <request_header> and <response_header> tags can be found in a separate document describing the generic i2b2 message. The remainder of this document describes the contents of the <message_body> for the Project Management (PM) Cell.

2. THE PROJECT MANAGEMENT (PM) CELL MESSAGING DETAIL

The Project Management (PM) cell has two basic functions: to control user access to various services and to keep track of where these services are located. In addition, it stores various (possibly personalized) parameters associated with the services. User access is determined by a user's 'role', which is a variable that defines the actions that user may perform within the HIVE. The role indicates how much data to return and whether or not there is access to a particular service. The default role is 'user' and there are other roles, such as 'manager', 'administrator', and 'protected_access', each with its own set of privileges regarding what data can be viewed. An individual i2b2 user may have one or more roles. The HIVE itself is defined by the PM service. The name of the HIVE is defined in the 'domain' variable. A client which has the root location of the PM services as well as the method (REST/SOAP) for the access to the services can obtain the structure of the HIVE through these services.

When a person logs in to the i2b2 workbench, a login screen comes up that requires the username, password and target location to be entered. (The target location is also called the domain name or simply the domain.) A request message is then sent to the PM cell. Each cell contains a mapping of domain names to urls which provide the address for the service in the PM that will authenticate the user. If the domain is not recognized by a cell (does not exist in the predefined lookup table), or the user does not have access to that domain, then the person is not authenticated to use that specific cell. If the domain exists and the user has access, then authentication occurs. In effect, the mapping of domain names to urls provides the security to the authentication process when using each cell of the HIVE, because the domain name must be "registered" with each cell with the address of a HIVE's PM cell, and then referenced before proceeding. The PM cell returns the user's roles for a project to determine their permissions and privileges in each HIVE cell.

The message used by the PM cell for authentication and authorization is **getUserConfiguration**.

2.1 Use Case

The PM cell may be accessed either by an i2b2 client or by another i2b2 cell. The actor is either an end user logging into an i2b2 client, or an i2b2 cell needing to verify particular access. The getUserConfiguration request message is sent to the PM cell and a response message is returned.

When an end user logs in, the username, password and domain name are entered and are contained in the message header part of the request message sent to the PM cell. The PM cell uses these values to authenticate the user. The values above may be seen to be replicated in the message_body part of the response message, but are not used by the PM cell for authentication. The replicated values are used to offer the information to other client-sided tools using the object.

The password is substituted by an encrypted token for many interactions. Return messages will then not contain the password.

If a user does not have access to the domain or if the domain does not exist, then the request is considered invalid and the invalid user request is returned. If the user does have access to the domain, then the PM cell returns a list of projects the user has access to and the user has the opportunity to pick one in which to continue working. Then a more specific message may be generated to obtain data only relevant to that project.

When another cell accesses the PM server, the purpose is to see what roles exist for this particular user for this cell. The username, password and domain are still in the message header. The body of the getUserConfiguration message may include the name of the project being accessed. If the user has a role that allows them to access the cell, a valid response message is returned.

In summary, the PM server performs the following steps via the getUserConfiguration message:

- a. Authentication: Verifies that the user is valid and is associated with the domain provided. A blank project will return all the domains for the user.

- b. Authorization: Gets a list of cells and roles available for this client from the PM cell and return them in the response
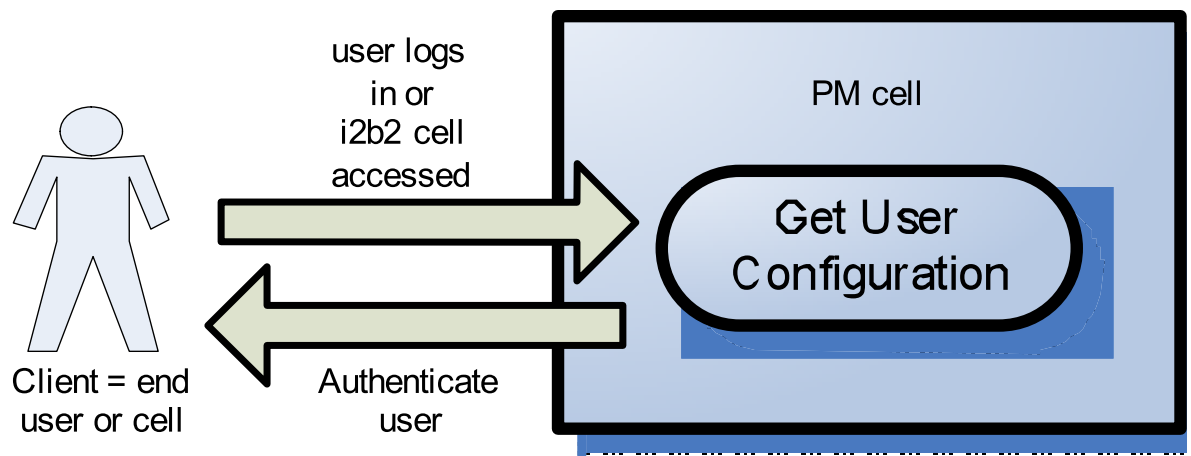
2.2 Messages

2.2.1 getServices - GetUserConfiguration Request Message

```
<message_body>
    <pm:get_user_configuration>
        <data_needed>USER</data_needed>
        <data_needed>CELL_DATAS</data_needed>
        <project>demo</project>
    </get_user_configuration>
</message_body>
```

A **getUserConfiguration message** is sent by a client application or another cell service to get a list of cells and/or roles associated with that client. The optional attribute 'project' can be used to specify for which project the PM cell should return data. If the attribute is blank, then all projects that the user has access to are returned. The optional parameter <data_needed> specifies an XPATH type method to return parts of the message. For example, USER to just receive the data enclosed by the <user> tags, CELL_DATAS to just receive the data enclosed by the <cell_datas> tags, and so forth for each top level tag. If the attribute is blank or is not included, all data is returned.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

```
<request>
    <request>
        <message_header></message_header>
        <request_header></request_header>
        <message_body></message_body>
    </request>
</request>
```

2.2.2 getServices – GetUserConfiguration Response Message

2.2.2.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

In the PM response message, there are optional 'param' XML elements in various locations of the XML. These can be used to describe additional parameters that may exist for different users, projects or global data. For example, if there is a need to prevent a user from getting access to particular data, the user param field could be used to hold a security alert. Or the project parameter could hold the name of a project sponsor or surrogate. The param tags are generic XML tags for adding more information to the messages.

```
<message_body>
    <configure>
        <environment>PRODUCTION</environment>
        <helpURL>http://i2b2.org/help</helpURL>
        <user>
            <full_name>Demo User</full_name>
            <user_name>demo</user_name>
            <password>demouser</password>
            <domain>HIVE</domain>
            <admin>true</admin>
            <project id="ASTH">
                <name>Asthma group</name>
                <key>X567j83</key>
                <wiki>htt://asthma.i2b2.org/wiki/</wiki>
                <role>USER</role>
                <role>MANAGER</role>
                <param name="IRB_Number">2008P00345</param>
            </project>
            <param name="user_key"1941jd2-02d21</param>
        </user>
        <cell_datas>
            <cell_data id="PM">
                <name>Project Management</name>
                <url>http://i2b2.org/PMService/</URL>
                <method>SOAP</method>
            </cell_data>
            <cell_data id="ONT">
                <name>ontology</name>
                <url>http://i2b2.org/OntologyServiceAsthma/</URL>
                <method>SOAP</method>
                <param name="OntMax">200</param>
                <param name="OntSyn">false</param>
            </cell_data>
            <cell_data id="PFT">
                <name>PFT</name>
```

```
            <url>http://i2b2.org/PFTService/</URL>
            <method>REST</method>
        </cell_data>
        <cell_data id="CRC">
            <name>PreviousQuery</name>
            <url>http://i2b2.org/QueryToolService/</URL>
            <method>SOAP</method>
        </cell_data>
    <cell_datas>
    <global_data>
        <param name-nickname>Demo guy</param>
    </global_data>
    </configure>
</message_body>
```

2.2.2.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Supplied password does not match user
password!</status>
    </result_status>
</response_header>
```

2.2.3 getServices - SetUser Request Message

```
<message_body>
    <pm:set_user>
        <user_name>local\mikem</user_name>
        <full_name>Mike Mendis</full_name>
        <email>mikem@i2b2.org</email>
        <admin>true</admin>
        <password>tiger</password>
    </pm:set_user>
</message_body>
```

A **setUser message** is sent by a client application or another cell service to add a user to the PM cell. If the user already exists, the setUser message will update the current record. If the user already exists and the password is not included, than the password will not change.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

```
<request>
    <request>
        <message_header></message_header>
        <request_header></request_header>
        <message_body></message_body>
    </request>
</request>
```

2.2.4 getServices – SetUser Response Message

2.2.4.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.4.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
```

```
        <status type="ERROR">User already exists.</status>
    </result_status>
</response_header>
```

2.2.5 getServices - DeleteUser Request Message

```
<message_body>
    <pm:delete_user>mikem</pm:delete_user>
</message_body>
```

A **deleteUser message** is sent by a client application or another cell service to delete a user to the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			

```
<request>
    <request>
        <message_header></message_header>
        <request_header></request_header>
        <message_body></message_body>
    </request>
</request>
```

2.2.6 getServices – DeleteUser Response Message

2.2.6.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
```

```
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.6.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.7 getServices - GetAllUser Request Message

```
<message_body>
    <pm:get_all_user></pm:get_all_user>
</message_body>
```

A **getAllUser message** is sent by a client application or another cell service to get a list of all user to the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			

2.2.8 getServices – GetAllUser Response Message

2.2.8.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <Users>
        <user>
            <full_name>Mike M</full_name>
            <user_name>mikem</user_name>
            <email>mikem@i2b2.org</email>
        </user>
    </users>
</message_body>
```

2.2.8.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Insufficient privileges.</status>
    </result_status>
</response_header>
```

2.2.9 getServices - GetUser Request Message

```
<message_body>
    <pm:get_user>demo</pm:get_user>
</message_body>
```

A **getUser message** is sent by a client application or another cell service to get a user in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self

ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.10 getServices – GetUser Response Message

2.2.10.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <Users>
        <user>
            <full_name>Mike M</full_name>
            <user_name>mikem</user_name>
            <email>mikem@i2b2.org</email>
            <password>1234567890123456</password>
        </user>
    </users>
</message_body>
```

2.2.10.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Insufficient privileges.</status>
    </result_status>
</response_header>
```

2.2.11 getServices - SetUserParam Request Message

```
<message_body>
    <pm:set_user_param>
        <user_name>mikem</user_name>
        <param datatype="i" name="hostid">i2b2.org</param>
    </pm:set_user_param>
```

```
    </message_body>
```

A **setUserParam message** is sent by a client application or another cell service to add a user parameter to the PM cell. The user needs to exist prior to calling this service. If the user and parameter name already exist, the record will be updated.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.12 getServices – SetUserParam Response Message

2.2.12.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.12.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.13 getServices - DeleteUserParam Request Message

```
<message_body>
    <pm:delete_user_param>34</pm:delete_user_param>
</message_body>
```

A **deleteUserParam message** is sent by a client application or another cell service to delete a user parameter from the PM cell.

The number in the value field is the id for that parameter.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.14 getServices – DeleteUserParam Response Message

2.2.14.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.14.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
```

```
            <status type="ERROR">User does not exists.</status>
        </result_status>
    </response_header>
```

2.2.15 getServices - GetUserParams Request Message

```
    <message_body>
        <pm:get_all_user_param>
            <user_name>mikem</user_name>
        </pm:get_all_user_param>
    </message_body>
```

A **getAllUserParam message** is sent by a client application or another cell service to get a list of all user parameters in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.16 getServices – GetAllUserParam Response Message

2.2.16.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
    <message_body>
        <Users>
            <user>
                <user_name>demo</user_name>
                <param datatype="T" is="8" name="IRB_Number">2008P00345</param>
            </user>
        </users>
```

```
</message_body>
```

2.2.16.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.17 getServices - GetUserParam Request Message

```
<message_body>
    <pm:get_user_param>8</pm:get_user_param>
</message_body>
```

A **getUserParam message** is sent by a client application or another cell service to get a user parameters value in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.18 getServices – GetUserParam Response Message

2.2.18.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <ns4:param datatype="int" id="8" name="aarrg">alive</ns4:param>
</message_body>
```

2.2.18.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.19 getServices - SetGlobal Request Message

```
<message_body>
    <pm:set_global>
    <project_path></project_path>
        <param datatype="i" name="aarrg">abc</param>
        <can_override>Y</can_override>
    </pm:set_global>
</message_body>
```

An **addGlobal message** is sent by a client application or another cell service to add a global parameter to the PM cell

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	
MANAGER		X	
USER			

2.2.20 getServices – SetGlobal Response Message

2.2.20.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.20.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Insufficient Privileges</status>
    </result_status>
</response_header>
```

2.2.21 getServices - DeleteGlobal Request Message

```
<message_body>
    <pm:delete_global>162</pm:delete_global>
</message_body>
```

A **deleteGlobal message** is sent by a client application or another cell service to delete a global parameter from the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	
MANAGER		X	
USER			

2.2.22 getServices – DeleteGlobal Response Message

2.2.22.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.22.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Insufficient Privileges.</status>
    </result_status>
</response_header>
```

2.2.23 getServices - GetAllGlobal Request Message

```
<message_body>
    <pm:get_all_global>/test</pm:get_all_global>
</message_body>
```

A **getAllglobal message** is sent by a client application or another cell service to get a list of all global parameters in the PM cell.

The value in the call is the project path

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

2.2.24 getServices – GetAllGlobal Response Message

2.2.24.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <params>
        <param datatype="i" id="11" name="aarrg">abc</param>
    </params>
</message_body>
```

2.2.24.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Does not exists.</status>
    </result_status>
</response_header>
```

2.2.25 getServices - GetGlobal Request Message

```
<message_body>
    <pm:get_global>11</pm:get_global>
</message_body>
```

A **getGlobal message** is sent by a client application or another cell service to get a global parameters value in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

2.2.26 getServices – GetGlobal Response Message

2.2.26.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <Global>
        <can_override>false</can_override>
        <project_path>/new</project_path>
        <param datatype="I" name="aarrg">abc</param>
    </Global>
</message_body>
```

2.2.26.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.27 getServices - SetRole Request Message

```
<message_body>
    <pm:set_role>
        <user_name>mem61</user_name>
        <role>DATA_DEID</role>
        <project_id>test_mart_3</project_id>
    </pm:set_role>
</message_body>
```

A **setRole message** is sent by a client application or another cell service to add a role to a project.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	
MANAGER		X	

USER			

2.2.28 getServices – SetRole Response Message

2.2.28.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.28.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.29 getServices - DeleteRole Request Message

```
<message_body>
    <pm:delete_role>
        <user_name>mem61</user_name>
        <role>DATA_DEID</role>
        <project_id>test_mart_3</project_id>
    </pm:delete_role>
</message_body>
```

A **deleteRole message** is sent by a client application or another cell service to delete a user in a project.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	
MANAGER		X	
USER			

2.2.30 getServices – DeleteRole Response Message

2.2.30.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.30.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.31 getServices - GetAllRole Request Message

```
<message_body>
    <pm:get_all_role>
        <project_id>test_mart_3</project_id>
    </get_all_role>
```

```
</message_body>
```

A **getAllRole message** is sent by a client application or another cell service to get a list of all users in a certain project.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	
MANAGER		X	
USER			

2.2.32 getServices – GetAllRole Response Message

2.2.32.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <roles>
        <role>
            <project_id>test_mart_3</project_id>
            <user_name>mikem</user_name>
            <role>DATA_AGG</role>
        </role>
        <role>
            <project_id>test_mart_3</project_id>
            <user_name>mikem</user_name>
            <role>DATA_OBFSC</role>
        </role>
    </roles>
</message_body>
```

2.2.32.2   RESPONSE MESSAGE FOR AN INVALID PROJECT REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Project does not exists.</status>
    </result_status>
</response_header>
```

2.2.33 getServices - GetRole Request Message

```
<message_body>
    <pm:get_role>
        <project_id>demoproject</project_id>
        <user_name>mikem</user_name>
    </pm:get_role>
 </message_body>
```

A **getRole message** is sent by a client application or another cell service to get a user in the specific project.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.34 getServices – GetRole Response Message

2.2.34.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <Roles>
        <role>
            <project_id>test_mart_3</project_id>
```

```
            <user_name>mikem</user_name>
            <role>DATA_AGG</role>
        </role>
        <role>
            <project_id>test_mart_3</project_id>
            <user_name>mikem</user_name>
            <role>DATA_OBFSC</role>
        </role>
    </Roles>
</message_body>
```

2.2.34.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.35 getServices - SetProjectUserParam Request Message

```
<message_body>
    <pm:set_project_user_param id="MYPRIJECT">
        <user_name>mikem</user_name>
        <param datatype="i" name="TEST">value</param>
    </pm:set_project_user_param>
</message_body>
```

A **setProjectUserParam message** is sent by a client application or another cell service to add a user parameter to a specific project. The user and project needs to exist prior to calling this service. If the id attribute is specified, the record will be updated.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.36 getServices – SetProjectUserParam Response Message

2.2.36.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.36.2   RESPONSE MESSAGE FOR AN INVALID USER

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.37 getServices - DeleteProjectUserParam Request Message

```
<message_body>
    <pm:delete_project_user_param>13</pm:delete_project_user_param>
</message_body>
```

A **deleteProjectUserParam message** is sent by a client application or another cell service to delete a user parameter from a specific project.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.38 getServices – DeleteProjctUserParam Response Message

2.2.38.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.38.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.39 getServices - GetAllProjectUserParam Request Message

```
<message_body>
<pm:get_all_project_user_param id="MYPRIJECT">
    <user_name>mikem</user_name>
</pm:get_all_project_user_param>
```

```
</message_body>
```

A **getAllProjectUserParam message** is sent by a client application or another cell service to get a list of all user parameters from a specific project.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.40 getServices – GetAllProjectUserParam Response Message

2.2.40.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <params>
        <param id="242" name="TEST">Wed Nov 25 13:13:34 EST 2009</param>
    </params>
</message_body>
```

2.2.40.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.41 getServices - GetProjectUserParam Request Message

```
<message_body>
    <pm:get_project_user_param>242</pm:get_project_user_param>
</message_body>
```

A **getProjectUserParam message** is sent by a client application or another cell service to get a user parameters value from a specific project.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body. The username who is the sender, needs to have the role of ADMIN in any project to execute this service or a MANAGER or USER for the specific project.

2.2.42 getServices – GetProjectUserParam Response Message

2.2.42.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <params>
        <param id="242" datatype="I" name="IRB">ADMIN</param>
    </params>
</message_body>
```

2.2.42.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">User does not exists.</status>
    </result_status>
</response_header>
```

2.2.43 getServices - SetProject Request Message

```
<message_body>
```

```
        <pm:set_project id="ONT">
            <name>My project</name>
        <key>123</key>
        <wiki>http://www.i2b2.org</wiki>
        <path>/project</path>
</pm:set_project>
</message_body>
```

A **setProject message** is sent by a client application or another cell service to add a user to the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

```
<request>
    <request>
        <message_header></message_header>
        <request_header></request_header>
        <message_body></message_body>
    </request>
</request>
```

2.2.44 getServices – SetProject Response Message

2.2.44.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.44.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Access Denied</status>
    </result_status>
</response_header>
```

2.2.45 getServices - DeleteProject Request Message

```
<message_body>
    <pm:delete_project id="ONT">
        <path>/project</path>
    </pm:delete_project>
</message_body>
```

A **deleteProject message** is sent by a client application or another cell service to delete a user to the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			

```
<request>
    <request>
        <message_header></message_header>
        <request_header></request_header>
        <message_body></message_body>
```

```
            </request>
        </request>
```

2.2.46 getServices – DeleteProject Response Message

2.2.46.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
        <message_header>
            <result_status>
                <status type="SUCCESS"/>
            </result_status>
        </message_header>
```

2.2.46.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
        <response_header>
            <result_status>
                <status type="ERROR">Porject does not exists.</status>
            </result_status>
        </response_header>
```

2.2.47 getServices - GetAllProject Request Message

```
        <message_body>
            <pm:get_all_project></get_all_project>
        </message_body>
```

A **getAllProjects message** is sent by a client application or another cell service to get a list of all projects from the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			

2.2.48 getServices – GetAllProject Response Message

2.2.48.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <projects>
        <project id="ASTH">
            <name>Asthma group</name>
            <path>/asthma</path>
            <key>X567j83</key>
            <wiki>http://asthma.i2b2.org/wiki/</wiki>
        </project>
    </projects>
</message_body>
```

2.2.48.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Insufficient privileges.</status>
    </result_status>
</response_header>
```

2.2.49 getServices - GetProject Request Message

```
<message_body>
    <pm:get_project id="ONT">
        <path>/project</path>
```

```
        </get_project>
    </message_body>
```

A **getProject message** is sent by a client application or another cell service to get a project in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.50 getServices – GetProject Response Message

2.2.50.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <project id="ASTH">
        <name>Asthma group</name>
        <key>X567j83</key>
        <wiki>http://asthma.i2b2.org/wiki/</wiki>
    </project>
</message_body>
```

2.2.50.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Invalid project.</status>
<   /result_status>
</response_header>
```

2.2.51 getServices - SetProjectParam Request Message

```
<message_body>
    <pm:set_project_param id="ASTH">
        <param datatype="I" name="IRB_Number">2008P00345</param>
    </set_project_param>
</message_body>
```

A **setProjectParam message** is sent by a client application or another cell service to add a project parameter to the PM cell. The project needs to exist prior to calling this service. If the id attribute is specified, the record will be updated.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.52 getServices – SetProjectParam Response Message

2.2.52.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.52.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Project does not exists.</status>
    </result_status>
</response_header>
```

2.2.53 getServices - DeleteProjectParam Request Message

```
<message_body>
    <pm:delete_project_param>24</delete_project_param>
</message_body>
```

A **deleteProjectParam message** is sent by a client application or another cell service to delete a project parameter from the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.54 getServices – DeleteProjectParam Response Message

2.2.54.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.54.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Project does not exists.</status>
    </result_status>
</response_header>
```

2.2.55 getServices - GetAllProjectParams Request Message

```
<message_body>
    <pm:get_all_project_param>Asthma</pm:get_all_project_param>
</message_body>
```

A **getAllProjectParam message** is sent by a client application or another cell service to get a list of all project parameters in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER		X	X

2.2.56 getServices – GetAllProjectParam Response Message

2.2.56.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <params>
        <param id="301" name="222">333</param>
    </params>
</message_body>
```

2.2.56.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Project does not exists.</status>
    </result_status>
</response_header>
```

2.2.57 getServices - GetProjetParam Request Message

```
<message_body>
    <pm:get_project_param>301</pm:get_project_param>
</message_body>
```

A **getProjectParam message** is sent by a client application or another cell service to get a project parameters value in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER		X	X

2.2.58 getServices – GetProjectParam Response Message

2.2.58.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <params>
```

```
                    <param id="301" name="222">333</param>
               </params>
          </message_body>
```

2.2.58.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
     <result_status>
          <status type="ERROR">Project does not exists.</status>
     </result_status>
</response_header>
```

2.2.59 getServices - SetHive Request Message

```
<message_body>
     <pm:set_hive>
          <environment>PRODUCTION</environment>
          <helpURL>http://i2b2.org/help</helpURL>
          <Active>true</Active>
          <domain_name>i2b2dev</domain_name>
     </pm:set_hive>
</message_body>
```

A **setHive message** is sent by a client application or another cell service to add a hive to the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER			
USER			

```
<request>
    <request>
        <message_header></message_header>
        <request_header></request_header>
        <message_body></message_body>
    </request>
</request>
```

2.2.60 getServices – SetHive Response Message

2.2.60.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.60.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Access Denied</status>
    </result_status>
</response_header>
```

2.2.61 getServices - DeleteHive Request Message

```
<message_body>
    <pm:delete_hive>1234567890123456</pm:delete_hive>
</message_body>
```

A **deleteHive message** is sent by a client application or another cell service to delete a hive in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER			
USER			

```
<request>
    <request>
        <message_header></message_header>
        <request_header></request_header>
        <message_body></message_body>
    </request>
</request>
```

2.2.62 getServices – DeleteHive Response Message

2.2.62.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.62.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Hive does not exists.</status>
    </result_status>
</response_header>
```

2.2.63 getServices - GetAllHive Request Message

```
<message_body>
    <pm:get_all_hive></pm:get_all_hive>
</message_body>
```

A **getAllHive message** is sent by a client application or another cell service to get a list of all hives from the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
HASIRB			X
USER			X

2.2.64 getServices – GetAllHive Response Message

2.2.64.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <hives>
        <hive>
            <environment>DEVELOPMENT</environment>
            <helpURL>http://www.i2b2.org</helpURL>
            <domain_name>i2b2dev</domain_name>
            <domain_id>i2b2dev</domain_id>
            <active>true</active>
        </hive>
    </hives>
</message_body>
```

2.2.64.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Insufficient privileges.</status>
    </result_status>
</response_header>
```

2.2.65 getServices - GetHive Request Message

```
<message_body>
    <pm:get_hive>1234567890123456</get_hive>
</message_body>
```

A **getHive message** is sent by a client application or another cell service to get a hive in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.66 getServices – GetHive Response Message

2.2.66.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <hive>
```

```
            <environment>DEVELOPMENT</environment>
            <helpURL>http://www.i2b2.org</helpURL>
            <domain_name>i2b2dev</domain_name>
            <domain_id>i2b2dev</domain_id>
            <active>true</active>
        </hive>
    </message_body>
```

2.2.66.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Invalid project.</status>
    </result_status>
</response_header>
```

2.2.67 getServices - SetHiveParam Request Message

```
<message_body>
    <pm:set_hive_param>
        <domain_id>i2b2dev</domain_id>
        <param name="IRB_Number">2008P00345</param>
    </pm:set_hive_param>
</message_body>
```

A **setHiveParam message** is sent by a client application or another cell service to add a hive parameter to the PM cell. The project needs to exist prior to calling this service. If the hive and parameter name already exist, the record will be updated.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X

USER			X

2.2.68 getServices – SetHiveParam Response Message

2.2.68.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.68.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Project does not exists.</status>
    </result_status>
</response_header>
```

2.2.69 getServices - DeleteHiveParam Request Message

```
<message_body>
    <pm:delete_hive_param>301</delete_hive_param>
</message_body>
```

A **deleteHiveParam message** is sent by a client application or another cell service to delete a hive parameter from the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.70 getServices – DeleteHiveParam Response Message

2.2.70.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.70.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Param does not exists.</status>
    </result_status>
</response_header>
```

2.2.71 getServices - GetAllHiveParams Request Message

```
<message_body>
    <pm:get_all_hive_param>i2b2demo</pm:get_all_hive_param>
</message_body>
```

A **getAllHiveParam message** is sent by a client application or another cell service to get a list of all hive parameters in the PM cell for a specific hive.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.72 getServices – GetAllHiveParam Response Message

2.2.72.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <params>
        <param id="303" name="IRB_Number">2008P00345</param>
        <param datatype="int" id="22" name="aarrgHive2">abc</param>
        <param datatype="int" id="163" name="aarrg">DAHSJK</param>
    </params>
</message_body>
```

2.2.72.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Hive does not exists.</status>
    </result_status>
</response_header>
```

2.2.73 getServices - GetHiveParam Request Message

```
<message_body>
    <pm:get_hive_param>303</get_hive_param>
</message_body>
```

A **getHiveParam message** is sent by a client application or another cell service to get a hive parameters value in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.74 getServices – GetHiveParam Response Message

2.2.74.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <params>
        <param id="303" name="IRB_Number">2008P00345</param>
    </params>
</message_body>
```

2.2.74.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Hive does not exists.</status>
    </result_status>
</response_header>
```

2.2.75 getServices - SetCell Request Message

```
<message_body>
    <pm:set_cell id="ONT">
        <project_path>/</project_path>
        <name>abc</name>
        <url>http://www.i2b2.org</url>
        <status>A</status>
        <method>SOAP</method>
        <can_override>true</can_override>
    </pm:set_cell>
</message_body>
```

A **setCell message** is sent by a client application or another cell service to add a cell to the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

```
<request>
    <request>
        <message_header></message_header>
        <request_header></request_header>
        <message_body></message_body>
    </request>
</request>
```

2.2.76 getServices – SetCell Response Message

2.2.76.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
```

```
            <status type="SUCCESS"/>
        </result_status>
    </message_header>
```

2.2.76.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Access Denied</status>
    </result_status>
</response_header>
```

2.2.77 getServices - DeleteCell Request Message

```
<message_body>
    <pm:delete_cell id="ONT">
        <project_path>/</project_poath>
    </pm:delete_cell>
</message_body>
```

A **deleteCell message** is sent by a client application or another cell service to delete a cell to the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			

```
<request>
```

```
<request>
    <message_header></message_header>
    <request_header></request_header>
    <message_body></message_body>
</request>
</request>
```

2.2.78 getServices – DeleteCell Response Message

2.2.78.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.78.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Porject does not exists.</status>
    </result_status>
</response_header>
```

2.2.79 getServices - GetAllCell Request Message

```
<message_body>
    <pm:get_all_cell></pm:get_all_cell>
</message_body>
```

A **getAllCell message** is sent by a client application or another cell service to get a list of all cells from the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			

2.2.80 getServices – GetAllCell Response Message

2.2.80.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <cells>
        <cell_data id="PM">
            <name>Project Management</name>
            <url>http://i2b2.org/PMService/</url>
            <method>SOAP</method>
            <project_path>/</project_path>
            <can_override>true</can_override>
        </cell_data>
        <cell_data id="ONT">
            <name>ontology</name>
            <url>http://i2b2.org/OntologyServiceAsthma/</url>
            <project_path>/</project_path>
            <can_override>true</can_override>
            <method>SOAP</method>
        </cell_data>
    </cells>
</message_body>
```

2.2.80.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Insufficient privileges.</status>
    </result_status>
</response_header>
```

2.2.81 getServices - GetCell Request Message

```
<message_body>
    <pm:cell_data id="ONT">
        <project_path>/</project_path>
    </pm:cell_data>
</message_body>
```

A **getCell message** is sent by a client application or another cell service to get a cell in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.82 getServices – GetCell Response Message

2.2.82.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <cell_data id="PM">
        <name>Project Management</name>
        <url>http://i2b2.org/PMService/</url>
        <method>SOAP</method>
        <project_path>/</project_path>
        <can_override>true</can_override>
    </cell_data>
</message_body>
```

2.2.82.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Invalid Cell.</status>
    </result_status>
</response_header>
```

2.2.83 getServices - SetCellParam Request Message

```
<message_body>
    <pm:set_cell_param>
        <cell_data id="PM">
            <param name="OntMax">200</param>
            <param name="OntSyn">false</param>
        </cell_data>
    </set_cell_param>
</message_body>
```

A **setCellParam message** is sent by a client application or another cell service to add a cell parameter to the PM cell. The cell needs to exist prior to calling this service. If the cell and parameter name already exist, the record will be updated.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.84 getServices – SetCellParam Response Message

2.2.84.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.84.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Cell does not exists.</status>
    </result_status>
</response_header>
```

2.2.85 getServices - DeleteCellParam Request Message

```
<message_body>
    <pm:delete_cell_param>301</delete_cell_param>
</message_body>
```

A **deleteCellParam message** is sent by a client application or another cell service to delete a cell parameter from the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.86 getServices – DeleteCellParam Response Message

2.2.86.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.86.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Project does not exists.</status>
    </result_status>
</response_header>
```

2.2.87 getServices - GetAllCellParams Request Message

```
<message_body>
    <pm:get_all_cell_param>
        <cell_data id="PM">
            <project_path>/</project_path>
        </cell_data>
    </get_all_cell_param>
 </message_body>
```

A **getAllCellParam message** is sent by a client application or another cell service to get a list of all cell parameters in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.88 getServices – GetAllCellParam Response Message

2.2.88.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <params>
        <param datatype="N" name="IRB_Number">2008P00345</param>
    </params>
</message_body>
```

2.2.88.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Cell does not exists.</status>
    </result_status>
</response_header>
```

2.2.89 getServices - GetCellParam Request Message

```
<message_body>
    <pm:get_cell_param>301</get_cell_param>
</message_body>
```

A **getCellParam message** is sent by a client application or another cell service to get a cell parameters value in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.90 getServices – GetCellParam Response Message

2.2.90.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <param name="IRB_Number">2008P00345</param>
</message_body>
```

2.2.90.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Cell does not exists.</status>
    </result_status>
</response_header>
```

2.2.91 getServices - SetPassword Request Message

```
<message_body>
    <pm:set_password>newpass</set_password>
</message_body>
```

A **setPassword message** is sent by a client application or another cell service to change the password of the user in the message_body. If the original password in

the message_body is correct than the password will be changed.   Only a user can
change there own password. If a admin needs to reset someone password, it is best
to use the set_user web_service call.

The domain, username, and password are sent in the header of the message, so do
not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN			X
MANAGER			X
USER			X

2.2.92 getServices – SetPassword Response Message

2.2.92.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.92.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Invalid password entered.</status>
    </result_status>
</response_header>
```

2.2.93 getServices - GetCodeInfo Request Message

```
<message_body>
    <get_code_info>
        <match_str>DATA_PROT</match_str>
    </get_code_info>
</message_body>
```

A **get_code_info message** implies that the user is passing a name and wants the associated information about that name returned. The category attribute will tell the service what metadata table to search in and does not have to be included. If column_cd is not included, all tables the user is allowed to access will be searched.

The attributes provide information about the results to be returned. If the number of rows found is greater than max, then an error message will be returned in the i2b2 header. If max is left out then it is interpreted that there is no max.

2.2.94 getServices – GetCodeInfo Response Message

2.2.94.1   RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <concepts>
            <concept>
                <tablecode>ABC</tablecode>
                <columncode>concept_path<columncode>
                <name>Protected_access</name>
                <code>DATA_PROT</code>
                <lookupkey>ABC</lookupkey>
                <hiveMgmt>None</hiveMgmt>
            </concept>
        </concepts>
    </result_status>
</message_header>
```

2.2.94.2   RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Invalid code entered.</status>
```

```
        </result_status>
    </response_header>
```

2.2.95 getVersion Message

i2b2 versioning will require that version numbers for different components match, that is, versioning will not be forward or backward compatible. If version numbers from different components are not equal, the user will be required to upgrade their software.

1. The i2b2workbench.properties file contains a message version number. The version number is a hive variable for the message version contained in the i2b2 message_header.

2. When a user logs on, the PM client sends a null 'get_message_version' tag

```
<message_header></message_header> <!- - Optional - >
<message_body>
    <get_message_version></get_message_version>
</message_body>
```

3. The response message from the PM cell contains the message version in the message body tag <i2b2_message_version>.

```
<message_header></message_header> <!- - Optional - >
<message_body>
    <i2b2_message_version>1.1</i2b2_message_version>
</message_body>
```

4. The workbench client compares the PM message version number with the version number from the workbench properties file. The two version numbers must be equal. If they are not equal, the user cannot continue until upgrading. If the version numbers are equal, the application will continue normally.

2.2.96 getServices – GetApproval Response Message

2.2.96.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <approval id="ra_pediatrics">
         <name>Pediatrics/</name>
        <description>SOAP</ description >
        <activation_date>/</ activation_date>
        <expiration_date>true</can_override>
        <object_cd>APPROVAL</object_cd>
    </approval >
</message_body>
```

2.2.96.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Invalid Approval.</status>
    </result_status>
</response_header>
```

2.2.97 getServices - SetApproval Request Message

```
<message_body>
    <set_approval id="ra_mart_test">
            <name>RA Mart Test</name>
            <description>Test environment for RA Mart</description>
            <object_cd>APPROVAL</object_cd>
             <activation_date>2010-07-07T12:00:00.000-04:00</activation_date>
             <expiration_date>NaN-NaN-NaNT12:00:00.000-04:00</expiration_date>
    </set_approval>
</message_body>
```

A **setApproval message** is sent by a client application or another cell service to add a Approval to the PM. If the approval id already exist, the record will be updated.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	
MANAGER		X	
USER			

2.2.98 getServices – SetApproval Response Message

2.2.98.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.98.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Approval does not exists.</status>
    </result_status>
</response_header>
```

2.2.99 getServices - DeleteApproval Request Message

```
<message_body>
    <pm:delete_approval>301</delete_ approval >
</message_body>
```

A **deleteApproval message** is sent by a client application or another cell service to delete a Approval from the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	
MANAGER		X	
USER			

2.2.100   getServices – DeleteApproval Response Message

2.2.100.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_header>
    <result_status>
        <status type="SUCCESS"/>
    </result_status>
</message_header>
```

2.2.100.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Approval does not exists.</status>
    </result_status>
</response_header>
```

2.2.101   getServices - GetAllApproval Request Message

```
<message_body>
    <pm:get_all_approval>
    </get_all_approval>
```

```
</message_body>
```

A **getAllApproval message** is sent by a client application or another cell service to get a list of all approvals in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.102   getServices – GetAllApproval Response Message

2.2.102.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
   <approvals>
     <approval id="ra_pediatrics">
         <name>Pediatrics/</name>
        <description>SOAP</ description >
        <activation_date>/</ activation_date>
        <expiration_date>true</can_override>
        <object_cd>APPROVAL</object_cd>
     </approval >
   </approvals>
</message_body>
```

2.2.102.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
   <result_status>
      <status type="ERROR">Cell does not exists.</status>
```

```
            </result_status>
        </response_header>
```

2.2.103   getServices - GetApproval Request Message

```
<message_body>
    <pm:get_approval>301</get_approval>
</message_body>
```

A **getApproval message** is sent by a client application or another cell service to get a approval in the PM cell.

The domain, username, and password are sent in the header of the message, so do not need to be sent separately in the message_body.

Role	Access Full	Access Project	Access Self
ADMIN	X	X	X
MANAGER		X	X
USER			X

2.2.104   getServices – GetApproval Response Message

2.2.104.1 RESPONSE MESSAGE FOR A VALID USER REQUEST

```
<message_body>
    <approval id="ra_pediatrics">
        <name>Pediatrics/</name>
        <description>SOAP</ description >
        <activation_date>/</ activation_date>
        <expiration_date>true</can_override>
        <object_cd>APPROVAL</object_cd>
    </approval >
</message_body>
```

2.2.104.2 RESPONSE MESSAGE FOR AN INVALID USER REQUEST

```
<response_header>
    <result_status>
        <status type="ERROR">Approval does not exists.</status>
    </result_status>
</response_header>
```

3. GLOSSARY

3.1 Message Tag & Attribute Definitions for PM_USER.xsd

The Project Management XML schema consists of one XSD file that defines the **<message_body>** for the entire PM cell, it is called ***PM_USER.xsd***.

3.2 Header Tag And Attributes Required By PM Cell:

<security>	Container for authentication
username	A valid user logon
password	The password associated with the user
domain	The server location from which the PM cell will authenticate and obtain roles for the user. When a service is accessed, the cell looks up the domain from a list and associates it with a URL. This URL is used to obtain authentication info. If the domain does not exist in the lookup table, the person is not authenticated.

3.3 Request Tag And Attribute:

<getUserConfiguration>	Container for a user data request
project	The project name associated with the user; if null, all projects are returned

3.4 Response Tags And Attributes:

<environment>	the conditions under which the client is being run, either PRODUCTION, DEVELOPMENT, INACTIVE, TEST, STOPPED or ARCHIVED

<helpurl>	Points to the URL for the message board displayed by the hive in the above environment.

<user>	Container for user-specific information.
user_name (user_ID)	Short string user for the user key.
full_name	The first and last name of the user.
password	The password of the user.
project	(defined below)
param	a name/value pair that can be used to give more info about a user.
name	A name that describes how the param tag is used.
admin	Is the user a admin

<project>	The projects associated with the user.
Id (Project_ID)	A unique string to identify a project.
name	Human-readable name of the project.
key	The key used to manage identified data of the project.
wiki	A url for information about the project.
role	One tag for each role associated with the project.
param	A name/value pair that can be used to give more info about a project.
name	A name that describes how the param tag is used.

<cell_datas>	The cell_data tags contain information about the cells a user has access to.
Id (Cell_ID)	A unique identifier for the cell.
name	The name of the cell.
url	The location of the cell.
param	A field that can be used to give more info about a cell.

☞ *NOTE: <cell_datas> cannot be null.*

<global_data>	Data which can be used by any of the cells, but is not specific to any individual cell, such as a database connection. Global_data is for additional variables associated with a particular domain.
param	A field that can be used to give more info about a cell.