Communique Responses

**Status**
This document is a request for a specification change for review.

**Summary**
The Communication OSID defines a means for asynchronous out of band communications between OSID Providers and OSID Consumers. OSID Consumers may supply simple responses to these messages. This document is a request to expand the Communication OSID to support a richer set of inputs.

**Table of Contents**

Current Specification.........................................................................................................................2
Problem..................................................................................................................................................2
Proposed Change...................................................................................................................................2
Impacts..................................................................................................................................................3
  Specification.......................................................................................................................................3
  OSID Consumers..............................................................................................................................3
  OSID Providers.................................................................................................................................3
Interoperability Considerations ..........................................................................................................3
Proposed Interfaces.............................................................................................................................3
  osid.communication.CommunicationSession......................................................................................3
  osid.communication.Communique......................................................................................................4
  osid.communication.ResponseForm....................................................................................................5
Example ..................................................................................................................................................6
Copyright Statement.............................................................................................................................9
1. **Current Specification**

Communiques are sent from a Communication OSID Provider to a user by way of Communication OSID Consumer. The consumer registers a CommunicationReceiver callback by which to receive these messages.

The Communique has a message, level, and an indication that a response is required. The response may require a selection from a set of ResponseOptions delivered via the Communique.

The Communication OSID can be used in conjunction with another OSID to send out of band messages and request user input.

2. **Problem**

There is no provision to supply data outside of a selection of fixed choices.

3. **Proposed Change**

Add a new response hook for supplying a ResponseForm. OsidForms typically have Metadata associated with each form input Metadata describes the syntax, constraints, and the purpose of the form input. The same Metadata can be used to describe a complex response input.

The inputs to OsidForms are generally specifically aligned with the OsidObject. An OsidObjectForm would have separate inputs for displayName and description, for example. However, the intent of the Communication OSID is to communicate with users through OSID Consumers. As such, specific agreements on what is being asked of the user need not be defined. Only the nature of the response is necessary to define. The proposed ResponseForm includes a set of Metadata and accepts input through abstract values.
4. Impacts

4.1. Specification
The set of changes is contained within the Communication OSID. The Communique and CommunicationSession will have additional methods to indicate and respond to these new ResponseForms.

4.2. OSID Consumers
A Communication OSID Consumer may not be prepared to handle a Communique requiring a complex response. This may also be true in the existing situation where it may not be ready to handle any response mechanism at all. While this change further complicates the functional agreement, existing OSID Consumers need to do nothing until it encounters an OSID Provider making use of this mechanism.

4.3. OSID Providers
Impact is limited to supplying a false value in Communique.respondViaForm() for compliance.

5. Interoperability Considerations
This adds a functional twist within the existing interfaces requiring an agreement between the OSID Consumer and OSID Provider.

Another consideration is the use of an abstract value in the ResponseForm. The object parameter type eases the burden on an OSID Consumer for supplying values. In the Configuration OSID, the Value specifies accessor methods for every primitive. The Value makes it clear to its consumer what to expect and binds the provider responsible for providing it. The ResponseForm flows in the other direction so there’s no benefit to the OSID Consumer. It becomes the responsibility of the OSID Provider the interpret the object based on its expectations described in Metadata.

6. Proposed Interfaces

6.1. osid.communication.CommunicationSession

<table>
<thead>
<tr>
<th>Interface</th>
<th>osid.communication.CommunicationSession</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implements</td>
<td>osid.OsidSession</td>
</tr>
<tr>
<td>Description</td>
<td>This session defines methods to process communications. Notifications are cancelled when this session is closed.</td>
</tr>
<tr>
<td>Method</td>
<td>registerForCommuniques</td>
</tr>
<tr>
<td>Description</td>
<td>Register for notifications of new communicues. CommunicationReceiver.newCommunique() is invoked when a new Communique is sent.</td>
</tr>
<tr>
<td>Errors</td>
<td>OPERATION FAILED</td>
</tr>
<tr>
<td></td>
<td>PERMISSION_DENIED</td>
</tr>
<tr>
<td>Compliance</td>
<td>mandatory</td>
</tr>
<tr>
<td></td>
<td>This method must be implemented.</td>
</tr>
</tbody>
</table>

Unable to complete request
An authorization failure occurred

Coppeto
### Method: acknowledge

**Description:** Sends a nonce response indicating that operations should continue or abort.

**Parameters:**
- `osid.id.Id` `communiqueId`: the Id of a communique

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILLEGAL_STATE</td>
<td><code>Communique.respondViaOption()</code> or <code>Communique.respondViaForm()</code> is true</td>
</tr>
<tr>
<td>INVALID_ARGUMENT</td>
<td><code>communiqueId</code> is not valid</td>
</tr>
<tr>
<td>NULL_ARGUMENT</td>
<td><code>communiqueId</code> is null</td>
</tr>
<tr>
<td>OPERATION_FAILED</td>
<td>unable to complete request</td>
</tr>
<tr>
<td>PERMISSION_DENIED</td>
<td>an authorization failure occurred</td>
</tr>
</tbody>
</table>

**Compliance:** mandatory

This method must be implemented.

### Method: respondViaOption

**Description:** Responds to a communique requiring a response option.

**Parameters:**
- `osid.id.Id` `responseOptionId`: the Id of a response option

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILLEGAL_STATE</td>
<td><code>Communique.respondViaOption()</code> is false</td>
</tr>
<tr>
<td>INVALID_ARGUMENT</td>
<td><code>responseOptionId</code> is not valid</td>
</tr>
<tr>
<td>NULL_ARGUMENT</td>
<td><code>responseOptionId</code> is null</td>
</tr>
<tr>
<td>OPERATION_FAILED</td>
<td>unable to complete request</td>
</tr>
<tr>
<td>PERMISSION_DENIED</td>
<td>an authorization failure occurred</td>
</tr>
</tbody>
</table>

**Compliance:** mandatory

This method must be implemented.

### Method: getResponseForm

**Description:** Gets the response form for submitting form-based responses.

**Parameters:**
- `osid.id.Id` `communiqueId`: the Id of a communique

**Return:**
- `osid.communication.ResponseForm`: a response form

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILLEGAL_STATE</td>
<td><code>Communique.respondViaForm()</code> is false</td>
</tr>
<tr>
<td>INVALID_ARGUMENT</td>
<td><code>communiqueId</code> is not valid</td>
</tr>
<tr>
<td>NULL_ARGUMENT</td>
<td><code>communiqueId</code> is null</td>
</tr>
<tr>
<td>OPERATION_FAILED</td>
<td>unable to complete request</td>
</tr>
<tr>
<td>PERMISSION_DENIED</td>
<td>an authorization failure occurred</td>
</tr>
</tbody>
</table>

**Compliance:** mandatory

This method must be implemented.

### Method: respondViaForm

**Description:** Responds to a communique requiring form input.

**Parameters:**
- `osid.communication.ResponseForm` `responseForm`: a response form

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILLEGAL_STATE</td>
<td><code>responseForm</code> already used in a create transaction</td>
</tr>
<tr>
<td>INVALID_ARGUMENT</td>
<td>the form contains an invalid value</td>
</tr>
<tr>
<td>NULL_ARGUMENT</td>
<td><code>responseForm</code> is null</td>
</tr>
<tr>
<td>OPERATION_FAILED</td>
<td>unable to complete request</td>
</tr>
<tr>
<td>PERMISSION_DENIED</td>
<td>an authorization failure occurred</td>
</tr>
<tr>
<td>UNSUPPORTED</td>
<td><code>responseForm</code> did not originate from <code>getResponseForm()</code></td>
</tr>
</tbody>
</table>

**Compliance:** mandatory

This method must be implemented.

### 6.2. osid.communication.Communique

**Interface:** `osid.communication.Communique`

**Implements:** `osid.OsidObject`

**Description:** A communique is a message to a user handled through an application. Communiques may require some form of acknowledgement ranging from a simple confirmation to data input.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Return</th>
<th>Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMessage</td>
<td>Gets the message.</td>
<td>osid.locale.DisplayText the message</td>
<td>mandatory</td>
</tr>
<tr>
<td>getLevel</td>
<td>Gets the message level for this communique.</td>
<td>osid.communication.CommuniqueLevel the message level</td>
<td>mandatory</td>
</tr>
<tr>
<td>isResponseRequired</td>
<td>Tests if the provider is blocking for a response. A response may take the form of a simple acknowledgement, a selection among a list of options, or a form input.</td>
<td>boolean true if the provider is blocking for a response false if no response is required</td>
<td>mandatory</td>
</tr>
<tr>
<td>respondViaOption</td>
<td>Tests if the provider is blocking on a choice selection. If respondViaOption() is true, then this method must return false. If isResponseRequired() is false and respondViaOption() is true, then a response is optional.</td>
<td>boolean true if the provider accepts a selection input, false otherwise</td>
<td>mandatory</td>
</tr>
<tr>
<td>respondViaForm</td>
<td>Tests if the provider is blocking on a form input. If respondViaOption() is true, then this method must return false. If isResponseRequired() is false and respondViaForm() is true, then a response is optional.</td>
<td>boolean true if the provider accepts a form input, false otherwise</td>
<td>mandatory</td>
</tr>
<tr>
<td>getCommuniqueRecord</td>
<td>Gets the communique record corresponding to the given Communique record Type. This method is used to retrieve an object implementing the requested record. The communiqueRecordType may be the Type returned in getRecordTypes() or any of its parents in a Type hierarchy where hasRecordType(communiqueRecordType) is true.</td>
<td>osid.communique.records.CommuniqueRecord the communique record</td>
<td>mandatory</td>
</tr>
</tbody>
</table>

**Parameters**
- osid.type.Type the type of the record to retrieve
- osid.communique.records.CommuniqueRecord the communique record

**Errors**
- NULL_ARGUMENT communiqueRecordType is null
- OPERATION_FAILED unable to complete request
- UNSUPPORTED hasRecordType(communiqueRecordType) is false

**6.3. osid.communication.ResponseForm**

**Interface** osid.communication.ResponseForm

**Implements** osid.OsidObjectForm

**Description** This is the form for submitting Responses.

**Method** getMetadata

**Description** Gets the metadata for all the fields. Some fields may be required and others optional.

**Return** osid.Metadata[] the metadata
### Compliance mandatory
This method must be implemented.

#### Method
**setValue**

**Description**
Sets the value for a given field.

**Parameters**
- `osid.id.Id id` - the Id of the metadata field
- `object value` - the value

**Errors**
- `INVALID_ARGUMENT` - id or value is invalid
- `NULL_ARGUMENT` - id or value is null
- `NO_ACCESS` - value cannot be modified

#### Method
**clearValue**

**Description**
Clears a value.

**Parameters**
- `osid.id.Id id` - the Id of the metadata field

**Errors**
- `NO_ACCESS` - Metadata.isRequired() or Metadata.isReadOnly() is true
- `NULL_ARGUMENT` - id is null

#### Method
**getResponseFormRecord**

**Description**
Gets the `ResponseFormRecord` corresponding to the given response record Type.

**Parameters**
- `osid.type.Type responseRecordType` - a response record type

**Return**
- `osid.communique.records.ResponseFormRecord` - the response form record

**Errors**
- `NULL_ARGUMENT` - responseRecordType is null
- `OPERATION_FAILED` - unable to complete request
- `UNSUPPORTED` - hasRecordType(responseRecordType) is false

#### Compliance mandatory
This method must be implemented.

### 7. Example

**Starting Communications from OSID Consumer**

```java
CommunicationsReceiver receiver = new MyCommunicationReceiver(this.communicationManager);
```

**A Communication Receiver Implementation**

```java
public class MyCommunicationReceiver
    implements org.osid.communication.CommunicationReceiver {

    private final org.osid.communication.CommunicationSession session;

    protected CommunicationReceiver(org.osid.communication.CommunicationManager mgr) throws org.osid.OsidException {
        this.session = mgr.getCommunicationSession(this);
        return;
    }
```
@OSID @Override
public void newCommunique(org.osid.communication.Communique com)
    throws org.osid.OsidException {
    if (com.respondViaOption()) {
        displayOptions(com);
    } else if (com.respondViaForm()) {
        displayForm(com);
    } else {
        displayMessage(com);
    }
    return;
}

protected void displayMessage(org.osid.communication.Communique com)
    throws org.osid.OsidException {
    switch (com.getLevel()) {
        case DEBUG:
            popupDebugMessage(com.getMessage());
            break;
        case INFORMATIONAL:
            popupInfoMessage(com.getMessage());
            break;
        default:
            popupAlert(com.getMessage());
    }
    this.session.acknowledge(com.getId());
    return;
}
protected void displayOptions(org.osid.communication.Communique com)
    throws org.osid.OsidException {

    SelectWidget w = new SelectWidget();
    w.setText(com.getMessage());
    for (org.osid.communication.ResponseOption option :
        com.getResponseOptions()) {
        w.addOption(option.getDescription());
    }

    int choice = w.popupAndGetSelection();
    this.session.respondViaOption(com.getResponseOptions()[choice]);
    return;
}

protected void displayForm(org.osid.communication.Communique com)
    throws org.osid.OsidException {

    FormWidget w = new FormWidget();
    w.setText(com.getMessage());
    org.osid.communication.ResponseForm form =
    this.session.getResponseForm(com.getId());
    for (org.osid.Metadata metadata : form.getMetadata()) {
        w.addFieldBasedOnMetadata(metadata);
    }

    w.popupAndCollectInput();
    Object[] values = w.getValues();
    for (int i ; i < values.length; i++) {
        form.setValue(form.getMetadata()[i].getId(), values[i]);
    }

    this.session.respondViaForm(form);
    return;
}

protected close() {
    try {
        this.session.close();
    } catch (Exception e) {} 
    return;
}
8. Copyright Statement

Copyright (C) Ingenescus (2014). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the authors, Ingenescus, or other organizations, except as required to translate it into languages other than English.

This document and the information contained herein is provided on an "AS IS" basis and Ingenescus and the authors DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.