Mobile Government
Gateway – Services Communication

Date: 15/04/2009
Version: 6.2
Department: PMO
01. **Document Reference**

*mGov-RED-Communication between Services and m-Gov gateway v6_2.doc*

02. **Document Type**

Word 2000 Document

03. **Security Classification**

Company Confidential

04. **Synopsis**

The aim of this document is to formalize and explain the methods involved for communication between the m-Gov Gateway and the applications (services such as the Court’s application, Licenses application, etc.) making use of the gateway.

05. **Document control**

<table>
<thead>
<tr>
<th>Author/s</th>
<th>Change controller</th>
<th>Distribution controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Christian Gatt</td>
<td>Christian Gatt</td>
<td></td>
</tr>
<tr>
<td>Software Developer</td>
<td>Software Developer</td>
<td></td>
</tr>
<tr>
<td>PMO</td>
<td>PMO</td>
<td></td>
</tr>
<tr>
<td>Michel Bugeja</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enterprise Architect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAU</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

06. **Authorisation**

Godwin Caruana,
Chief Enterprise Architect

Signature  Date:
## 07. Modification history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1</td>
<td>07/10/02</td>
<td>Draft</td>
</tr>
<tr>
<td>Version 2</td>
<td>14/10/02</td>
<td>Draft</td>
</tr>
<tr>
<td>Version 3</td>
<td>10/12/03</td>
<td>Ver 3</td>
</tr>
<tr>
<td>Version 4</td>
<td>12/01/03</td>
<td>Ver 4</td>
</tr>
<tr>
<td>Version 5</td>
<td>22/07/03</td>
<td>Ver 5</td>
</tr>
<tr>
<td>Version 6</td>
<td>15/04/04</td>
<td>Ver 6</td>
</tr>
<tr>
<td>Version 6.1</td>
<td>15/09/08</td>
<td>Ver 6.1 Correction of Schema</td>
</tr>
<tr>
<td>Version 6.2</td>
<td>15/04/09</td>
<td>Ver 6.2 MITA Transition</td>
</tr>
</tbody>
</table>
# Table of Contents

1.0 INTRODUCTION .......................................................................................................................... 5

2.0 FIRST TIME REGISTRATION OF SERVICES .............................................................................. 6

3.0 REGISTRATION METHODS .......................................................................................................... 7

   3.1 MANUAL .................................................................................................................................. 7
   3.2 AUTONOMOUS ......................................................................................................................... 7

4.0 SENDING OF MESSAGES AND ACKNOWLEDGEMENTS ............................................................. 8

   4.1 MANUAL SYSTEM .................................................................................................................... 9
   4.2 INTEGRATED SYSTEM ............................................................................................................. 10

5.0 BULK REGISTRATION/DE-REGISTRATION .............................................................................. 18

   5.1 INTEGRATING THE USE OF THE M-GOV BULK REGISTRATION SERVICE ....................... 19
   5.2 BULK REGISTRATION AND DE-REGISTRATION XML EXAMPLES ..................................... 20

6.0 APPENDICES ............................................................................................................................... 23

   6.1 APPENDIX 1 ............................................................................................................................ 23
   6.2 APPENDIX 2 ............................................................................................................................ 24
1.0 Introduction

As discussed with the project teams, the m-Government infrastructure is to provide a number of interfaces to allow the services to communicate with the gateway. These interfaces will allow the services to communicate with the m-Gov gateway either in a manual way that is through a number of web pages, or in an integrated way that is amalgamating use of the m-Gov functionality within the service’s applications so as to achieve the required goals.

If manual use is required by a service than sending of messages, registration/de-registration of mobile numbers and checking of message statuses are to be achieved via the use of a number of web pages. On the other hand acknowledgements from the m-Gov gateway and messages from citizens are to be forwarded to the services by the use of email.

If integrated use is preferred by the service than sending of messages, registration/de-registration of mobile numbers, receiving of acknowledgements and messages are to be achieved through the exchange of predefined XML files between the m-Gov gateway and the services.
2.0 First time registration of services

Initially any service requiring use of m-Gov services is to apply and register with the implemented Support System. Once the service is registered it is provided with the following information:

- Unique identification number dialed by mobile users to communicate with the service (issued by the Malta Communication Authority)
- Service's email address
- Username
- Password

Note: Additional usernames and passwords for the same service can be applied for in the same manner.

It is imperative that apart from giving the usual registration information, such as the user responsibilities for the service, password, etc, the service’s registration must indicate the interface mechanism with the m-Gov gateway (i.e. Manual or Integrated).
3.0 Registration Methods

One important aspect to keep in mind when registering a new mobile number is the “Data Protection Act” since the service sending the message could be sued for sending messages to recipients who are not registered for the service. Therefore one must integrate appropriate business processes to cater for checking and vetting of each and every mobile number before actual registration is computed.

Two methods of registration/de-registration are available for the m-Government gateway:

1. Manual registration/de-registration
2. Autonomous registration/de-registration

3.1 Manual

Manual registration and de-registration is to be provided through the implemented m-Gov web pages. This functionality will be available for administrators of the system.

3.2 Autonomous

Autonomous registration/de-registration will be available using the mechanism similar to sending messages. The only difference is that the types will be different and the posts will be done to a different ASP page over https. Refer to section 5 for detailed explanation of autonomous bulk registration/de-registrations.
4.0 Sending of messages and acknowledgements

The following figure displays the lifecycle of a message and the related acknowledgements a service will receive on submitting a message.

![Diagram showing the lifecycle of a message and related acknowledgements](attachment:diagram.png)

**Figure 1:** Messages and acknowledgements being transferred between the services, m-Gov infrastructures and operators

From figure 1 above one may notice that a message could be initiated from two main points (a) from the services and (b) the operator’s infrastructure. When a message is instantiated from a service it is packaged with other messages into an XML file (in accordance to the defined XSD) and posted to the m-Government gateway. On successfully receiving the XML file the gateway will reply back with an object state of 200 and a string containing _ok and the file size in bytes (see appendix 1 for code sample). If the XML file was successfully sent but could not be loaded an object state of 200 and a string with the error returned is passed. It is then up to the gateway to process the XML files sent and process them accordingly but this is done asynchronously from the first process mentioned above.

After receiving the XML file from the services the gateway will store the file and process it at a later stage. The steps involved after capturing the XML file are:

- Validate the XML file and check if it is well formed or valid (according to the XSD). If this check fails an object state of 200 and a string with the error _ERR(1) is returned.
- Check if the username and password are valid and that they have access to the particular service. If the username and password supplied are incorrect than the XML file is stored to disk but the messages are not processed.
• If the user name and password are valid then each individual message is processed. The first check done on each message is to check if the originator ID is equivalent to the ID of the service sending the XML file.

• If successful another check is done on the mobile number to see if it is registered to the service. If a subject is present in the message (within the Subject element of the XML file) then the mobile number must also be registered to the appropriate subject. If any one of these checks fails appropriate acknowledgements are posted to the service.

• Once valid, the message is passed to an internal queue where it will be posted to the appropriate mobile operator. When the m-Gov gateway successfully posts the message, an acknowledgement is forwarded to the services indicating that the message has been sent to the mobile providers.

• When the appropriate mobile operator receives the message it has to forward an acknowledgement to the m-Gov gateway stating that it successfully received the message. Another acknowledgement is sent to the gateway once the message is delivered to the recipient’s mobile. The acknowledgement indicating that the message arrived on the recipient’s mobile is then forwarded to the appropriate service.

Similarly, when a message originates from the mobile operators they are to send the messages to the m-Gov gateway, which in turn forwards them to the appropriate service as an XML file or as a number of emails.

4.1 Manual System

As already described, the manual systems will interact with the m-Gov gateway via a number of web pages offered. A user is to logon to the system using the assigned username and password. Once authenticated, the logged on user may administer his account, register/de-register mobile numbers, send messages and check the status of sent messages. A session is assigned to the user once logged on to the system so as to reduce the risk of malicious use of the system if the user does not logout when he is not using it.

The manual systems will receive messages and all information related to the message via email. It is therefore up to the service to check his email box for any new messages. The email's structure will be in a standard format and will include appropriately the following information:

- Transaction ID of message
- Origin Number of message
- Destination Number of message
- Type of message, i.e. an SMS or an acknowledgement
- The actual Message
- The mobile number of the recipient
- The state of the message, i.e. successful, invalid or any other error.
- The subject
4.2 Integrated System

All integrated systems, using the m-Gov gateway to send and receive messages, have to communicate using XML, that is they are to post messages via a defined XML schema to an ASP page offered by the m-Gov infrastructure. Also, each service is to publish a web page to accept XML files from the gateway. Each web page (both the m-Gov and service pages) must return an object state of 200 and a string containing _ok and the file size_ in bytes on successfully receiving an XML file (refer to appendix 1 for code sample). If for any reason the m-Gov gateway does not manage to establish a connection or be able to post the XML file then the XML file in question is attached to an email and posted to the service’s email address.

The XML file sent by the services to the gateway must have the following structure (is in conformance with the proposed XSD – refer to appendix 2):

```
<PostData>
  <Params>
    <UserID>  </UserID>
    <Password>  </Password>
    <Src>  </Src>
  </Params>
  <Root>
    <Transaction>
      <TransID>  </TransID>
      <Orig>  </Orig>
      <Dest>  </Dest>
      <TimeStamp>  </TimeStamp>
      <Type>  </Type>
      <Msg>  </Msg>
      <Mobile>  </Mobile>
      <Subject>  </Subject>
    </Transaction>
    <Transaction>
      <TransID>  </TransID>
      <Orig>  </Orig>
      <Dest>  </Dest>
      <TimeStamp>  </TimeStamp>
      <Type>  </Type>
      <Msg>  </Msg>
      <Mobile>  </Mobile>
      <Subject>  </Subject>
    </Transaction>
  </Root>
</PostData>
```
Similarly the XML file sent by the m-Gov gateway to the services will have the following structure (note that the XML file might have both acknowledgements and messages):

```
<UserID Element>
User ID assigned by support system

>Password Element
Password assigned by support system

>Src Element
The src is required to indicate if the service using the m-Gov services is automated or manual. For all integrated services the value should always be “A”.

>TransID Element
Unique hash value (SHA 256) computed by the service creating the message. This hash value is to be produced by hashing the concatenated string of the Origin, Destination, Mobile, Timestamp and message. This element should be made up of a string of 256 characters.
```
**Orig and Dest Elements**
The origin and destination elements are to indicate the origin and destination of a message or acknowledgement respectively. Each one of these values is to be unique and defined beforehand. The values are going to be a 6-digit value for example:

- m-Gov – 501500
- Vodafone – 501501
- Go Mobile – 501502
- Other MO – 501xxx
- Courts – 501600
- Licenses – 501601
- Etc – Etc

**TimeStamp Element**
The timestamp element indicates the time when the transaction is executed that is for example when the message is received on the mobile recipient, when the m-Gov infrastructure forwards the message, etc. The type of the timestamp element is a 19 character fixed string in the following format “dd/mm/yyyy hh:mm:ss”.

**Type Element**
The type element indicates if the transaction is a message or an acknowledgement:

The following acknowledgements are to be forwarded to the services:

- Acknowledgement 52 (ack52) instigated once the m-Gov infrastructure acknowledges a message received from a service.
- Acknowledgement 53 (ack53) instigated once the m-Gov infrastructure receives an acknowledgement that the message originally sent by the service has been received on the recipients mobile.

Following are the integers proposed for this element:

- 51 - indicates that the transaction is a new message
- 52 - indicates that the transaction is an ack52
- 53 - indicates that the transaction is an ack53
- 54 - indicates that the transaction is a registration
- 55 - indicates that the transaction is a de-registration

**Msg Element**
The msg element is to contain the actual message being passed. When the transaction is an acknowledgement this element is to be null. This element is to be of type string containing 160 characters.

**Mobile Element**
The mobile element is to contain the mobile number of the recipient in international format. This element is not required when acknowledgements are posted. The type of this element should be a string of 13 characters starting with 00356 (e.g. 0035679123456).
**Priority Element**
This element is to indicate the priority or order of the message being passed to the operators. This field should be of type integer and the values passed as follows:

- 0 – Normal priority
- 1 – High priority

This element is not required as it is up to the gateway to assign the priority of the message depending on the service sending it.

**State Element**
The status element is to indicate the status of the acknowledgement transaction that is:

- 0 – Successful
- 1 – Unsuccessful
- others – Errors

Refer to the Development Support web page for further details regarding the error codes that are forwarded: http://devdsl

**Subject Element**
The subject element is required for registration and de-registration and if needed to authenticate if a mobile number is registered to a specific subject.

It must be noted that the XML file being sent by a service **cannot** contain messages originating from another service.

Multiple messages are to be grouped together as much as possible before creating an XML file. If an XML file contains more than **1000** messages an error is returned _ERR(2)_ once posted.

The XML file posted has to be in conformance with the defined XSD (see appendix 2).
**XML examples**

**Scenario 1:** Service (e.g. Courts) sends 2 messages, one to be routed to Vodafone and the other to Gomobile:

In the above example the courts application is posting two messages to the gateway. The first message contains a Subject whilst the second does not. This means that when the first message is validated the mobile number provided must be registered to the given subject and service. In the second message the subject is not present therefore the mobile number is just validated against the service.
**Scenario 2:** m-Gov gateway forwards 1 message to a service (e.g. Courts):

Messages from m-Gov to Courts (forwarded from mobile operators). ID of courts is 501600 and ID of Vodafone and Go Mobile are 501502 and 501501 respectively. The trans_id is generated by the mobile provider:

```xml
<PostData>
   <Root>
      <Transaction>
         <TransID>bje360hbdshkt6yu</TransID>
         <Orig>501501</Orig>
         <Dest>501600</Dest>
         <TimeStamp>20/09/2002 11:37:45</TimeStamp>
         <Type>51</Type>
         <Msg>Test Message</Msg>
         <Mobile>0035679000002</Mobile>
      </Transaction>
   </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts a message to the courts application.

**Scenario 3:** m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that a message was successfully processed by the gateway:

Messages from m-Gov to Courts. ID of courts is 501600 and ID m-Gov is 501500 respectively. The trans_id was originally generated by the service:

```xml
<PostData>
   <Root>
      <Transaction>
         <TransID>078ytgf0000000000</TransID>
         <Orig>501500</Orig>
         <Dest>501600</Dest>
         <Type>52</Type>
         <State>0</State>
      </Transaction>
   </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular message was successfully (state = 0) processed by the gateway. The type of the message is 52.
Scenario 4: m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that the mobile recipient successfully received the message.

Messages from m-Gov to Courts. Assume that ID of courts is 501600 and ID of Vodafone and Go Mobile are 501501 and 501502 respectively. The trans_id was originally generated by the service:

```xml
<PostData>
  <Root>
    <Transaction>
      <TransID>000000000asdf</TransID>
      <Orig>501501</Orig>
      <Dest>501600</Dest>
      <Type>53</Type>
      <State>0</State>
    </Transaction>
  </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular message was successfully received (state = 0) on the mobile of the recipient. The type of the message is 53.

Scenario 5: m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that the mobile recipient did not received the message.

Messages from m-Gov to Courts. ID of courts is 501600 and ID of Vodafone and Go Mobile are 501501 and 501502 respectively. The trans_id is generated by the mobile provider:

```xml
<PostData>
  <Root>
    <Transaction>
      <TransID>58hdhtgfdgfd00asdf</TransID>
      <Orig>501501</Orig>
      <Dest>501600</Dest>
      <TimeStamp>20/09/2002 11:51:35</TimeStamp>
      <Type>53</Type>
      <State>1</State>
    </Transaction>
  </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular message was unsuccessfully received (state = 1) on the mobile of the recipient. The type of the message is 53.
Scenario 6: m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that the message generated contains an error (the mobile recipient is not registered in the centralized database).

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular message was unsuccessfully processed by the gateway since the mobile number is not registered to receive messages for that particular service (state = 116). The type of the message is 52.
5.0 Bulk Registration/De-registration

The following figure displays the lifecycle of a bulk registration/de-registration and the related acknowledgements a service will receive on submitting an XML file (containing registrations and de-registrations).

![Diagram of bulk registration/de-registration process]

When an automated service needs to register or de-register a number of mobile users it is to package the mobile details into an XML file separate from the bulk messages (messages and registrations/de-registrations are to be separated into different XML files) and post it to the m-Gov gateway (https://secure.gov.mt/mgov/xml_bulkreg.asp). In return, the m-Gov gateway will validate the XML file and register and/or de-register accordingly. No acknowledgements will be forwarded on successfully registering or de-registering a number, instead acknowledgements will only be sent if the registration or de-registration fails.

Figure 2: Messages and acknowledgements being transferred between the services and m-Gov infrastructures.
The XML file (containing registrations and de-registrations) sent by the services to the gateway will have the same structure as when sending bulk message, following is the XML structure used (refer to section 4.2):

```
<PostData>
  <Params>
    <UserID> </UserID>
    <Password> </Password>
    <Src> </Src>
  </Params>
  <Transaction>
    <TransID> </TransID>
    <Orig> </Orig>
    <Dest> </Dest>
    <TimeStamp> </TimeStamp>
    <Type> </Type>
    <Msg> </Msg>
    <Mobile> </Mobile>
    <Subject> </Subject>
  </Transaction>
</Root>
</PostData>
```

It must be noted that the XML file being sent by a service cannot contain messages originating from another service and the XML file posted has to be in conformance with the defined XSD (refer to appendix 2).

### 5.1 Integrating the Use Of The m-Gov Bulk Registration Service

All integrated systems, using the m-Gov gateway to register mobile users in bulk, have to communicate using XML, that is they are to post registrations/de-registrations via a defined XML schema as previously defined (refer to appendix 2) to an ASP page XML_BulkReg.ASP (full URL is `https://secure.gov.mt/mgov/XML_BulkReg.ASP`) offered by the m-Gov infrastructure. The number of transactions per XML file must not exceed 1000 but it is recommended that fewer transactions should be posted per file.

The ASP page must return an **object state of 200** and a **string containing _ok and the file size** in bytes on successfully receiving an XML file (refer to appendix 1 for code sample). If the file is not valid when checked against the XSD an **object state of 200** and a **string with _ERR(2) and the error message** is to be passed.
5.2 Bulk Registration and De-registration XML examples

Scenario 1: Service (e.g. Courts) sends one registration and one de-registration with no subjects:

Assume that ID of courts is 501601 and ID of m-Gov gateway is 501500. The trans_id is generated by the service:

```
<PostData>
    <Params>
        <UserID>burdv001</UserID>
        <Password>burdv100</Password>
        <Src>A</Src>
    </Params>
    <Root>
        <Transaction>
            <TransID>000000000002</TransID>
            <Orig>501601</Orig>
            <Dest>501500</Dest>
            <Type>54</Type>
            <Msg></Msg>
            <Mobile>0035699000002</Mobile>
        </Transaction>
        <Transaction>
            <TransID>000000000003</TransID>
            <Orig>501601</Orig>
            <Dest>501500</Dest>
            <TimeStamp>20/09/2002 11:37:35</TimeStamp>
            <Type>55</Type>
            <Msg></Msg>
            <Mobile>0035679000001</Mobile>
        </Transaction>
    </Root>
</PostData>
```

In the above example the courts application is posting a registration and a de-registration to the gateway. In both cases no subjects are inserted therefore after validating the XML file the appropriate user is either registered or de-registered from the service.
**Scenario 2:** Service (e.g. Courts) sends one registration and one deregistration with subjects:

Assume that ID of courts is 501601 and ID of m-Gov gateway is 501500. The trans_id is generated by the service:

```xml
<PostData>
<Params>
  <UserID>burdv001</UserID>
  <Password>burdv100</Password>
  <Src>A</Src>
</Params>
</Root>
</PostData>
```

In both cases the messages contain a Subject. This means that when registering or de-registering, validation checks on whether the subjects exist apply. In the first transaction the mobile number 0035699000002 is being registered for the courts service as well as to subject ‘case001’. If the subject does not exist then the user cannot be registered therefore appropriate errors are to be issued. In the second transaction the user 0035679000001 is being de-registered from the court’s subject ‘case005’. In this scenario if the subject does not exist then the de-registration will fail and an appropriate error message will be raised and sent to the courts service.
**Scenario 3:** m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that the registration failed since the user is not registered (the mobile recipient is not registered in the centralized database).

Messages from m-Gov to Courts. Assume that ID of courts is 501601 and m-Gov is 501500. The trans_id was originally generated by the service (courts):

```
<PostData>
  <Root>
    <Transaction>
      <TransID>078ytgf0000000000</TransID>
      <Orig>501500</Orig>
      <Dest>501601</Dest>
      <Type>54</Type>
      <State>1</State>
    </Transaction>
  </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular registration was unsuccessfully processed by the gateway (state = 1). The type of the message is 54.
6.0 Appendices

6.1 Appendix 1

```vba
Dim x As New MSXML2.XMLHTTP
'object found within the MSXML3.dll
Dim xmlDoc As New DOMDocument
Dim strErr As String

blnLoad = xmlDoc.Load("d:\temp\mgov\xmlfiles\testtrans2.xml")
If Not blnLoad Then
    MsgBox "error loading"
    Set xmlDoc = Nothing
    Set x = Nothing
    Exit Sub
End If

x.open "POST", "http://testserver/mgov/xml_gp.asp", False
x.send xmlDoc

If x.Status <> 200 Then
    'something wrong
    MsgBox x.responseText, vb_Exclamation
Else
    'should return OK and no. of bytes
    MsgBox x.responseText
End If

Set x = Nothing
Set xmlDoc = Nothing
```
6.2 Appendix 2

Schema mgov_to_services.xsd

Elements

PostData

**Diagram**

- **PostData**
  - Each XML file contains 1 params section and 1 root section

**Children**

- **Params Root**
  - The root element contains one or more msgs and acts

**Source**

```
<xs:element name="PostData">
  <xs:annotation>
    <xs:documentation>Each XML file contains 1 params section and 1 root section</xs:documentation>
    <xs:annotation>
      <xs:documentation>Params Root</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Params" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="UserID" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="8"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="Password" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="10"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="Src" minOccurs="0">
                <xs:complexType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="1"/>
                    <xs:pattern value="[A M]"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="Session" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="256"/>
                    <xs:pattern value="[A M]"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Session" minOccurs="0">
          <xs:complexType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="256"/>
              <xs:pattern value="[A M]"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:annotation>
</xs:element>
```
<xs:element name="SessionID" minOccurs="0">
  <xs:restriction base="xs:long">
    <xs:minInclusive value="0"/>
  </xs:restriction>
</xs:element>

<xs:element name="Root">
  <xs:documentation>The root element contain one or more msgs and acks</xs:documentation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Transaction" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="TransID">
              <xs:annotation>
                <xs:documentation>Transaction Hash</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="256"/>
                  <xs:minLength value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Orig">
              <xs:annotation>
                <xs:documentation>Originating ID</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:integer">
                  <xs:totalDigits value="6"/>
                  <xs:minExclusive value="99999"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Dest">
              <xs:annotation>
                <xs:documentation>Destination ID</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:integer">
                  <xs:totalDigits value="6"/>
                  <xs:minExclusive value="99999"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="TimeStamp">
              <xs:annotation>
                <xs:documentation>Time Stamp</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:pattern value="[0-9][0-9][0-9][0-9][0-9][0-9] [0-2][0-9]:[0-5][0-9]"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Type">
              <xs:annotation>
                <xs:documentation>Type: 51 = transaction is a msg; 52,53 = transaction is an ack; 54 = transaction is a registration; 55 = transaction is a de-registration;</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:nonNegativeInteger">
                  <xs:enumeration value="51"/>
                  <xs:enumeration value="52"/>
                  <xs:enumeration value="53"/>
                  <xs:enumeration value="54"/>
                  <xs:enumeration value="55"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Msg" minOccurs="0">
              <xs:simpleType>
                <xs:restriction base="xs:string"/>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

mobile numbers must be in international format e.g. 0035699123456

0 = Normal Priority ; 1 = High Priority

0 = Successful ; others = errors and other statuses

To be used when registering and de-registering

element PostData/Params
All services posting an XML file have to pass a userid, password and a src.

```xml
<xs:element name="Params" minOccurs="0">
  <xs:annotation>
    <xs:documentation>All services posting an XML file have to pass a userid, password and a src</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="UserID" minOccurs="0">
        <xs:restriction base="xs:string">
          <xs:maxLength value="8"/>
        </xs:restriction>
      </xs:element>
      <xs:element name="Password" minOccurs="0">
        <xs:restriction base="xs:string">
          <xs:maxLength value="10"/>
        </xs:restriction>
      </xs:element>
      <xs:element name="Src" minOccurs="0">
        <xs:annotation>
          <xs:documentation>A = Automated service ; M = Manual service</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="1"/>
            <xs:minLength value="1"/>
            <xs:pattern value="[A M]"/>
          </xs:restriction>
        </xs:complexType>
      </xs:element>
      <xs:element name="Session" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Session Hash</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="256"/>
            <xs:minLength value="1"/>
          </xs:restriction>
        </xs:complexType>
      </xs:element>
      <xs:element name="SessionID" minOccurs="0">
        <xs:restriction base="xs:long">
          <xs:minInclusive value="0"/>
        </xs:restriction>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
element **PostData/Params/UserID**

<table>
<thead>
<tr>
<th>diagram</th>
<th>UserID</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>facets</td>
<td>maxLength 8</td>
</tr>
</tbody>
</table>
| source  | <xs:element name="UserID" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element> |

element **PostData/Params/Password**

<table>
<thead>
<tr>
<th>diagram</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>facets</td>
<td>maxLength 10</td>
</tr>
</tbody>
</table>
| source  | <xs:element name="Password" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="10"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element> |

element **PostData/Params/Src**

<table>
<thead>
<tr>
<th>diagram</th>
<th>Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>facets</td>
<td>minLength 1, maxLength 1, pattern [A M]</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation = A = Automated service ; M = Manual service</td>
</tr>
</tbody>
</table>
| source  | <xs:element name="Src" minOccurs="0">
  <xs:annotation>
    <xs:documentation>A = Automated service ; M = Manual service</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="1"/>
      <xs:pattern value="[A M]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element> |
element PostData/Params/Session

<table>
<thead>
<tr>
<th>Diagram</th>
<th>Session Hash</th>
</tr>
</thead>
</table>

**Type** restriction of xs:string

**Facets**
- minLength: 1
- maxLength: 256

**Annotation** documentation Session Hash

**Source**
```xml
<xs:element name="Session" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Session Hash</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="256"/>
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

---

element PostData/Params/SessionID

<table>
<thead>
<tr>
<th>Diagram</th>
<th>SessionID</th>
</tr>
</thead>
</table>

**Type** restriction of xs:long

**Facets**
- minInclusive: 0

**Source**
```xml
<xs:element name="SessionID" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:long">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

---

element PostData/Root

<table>
<thead>
<tr>
<th>Diagram</th>
<th>Transaction</th>
</tr>
</thead>
</table>

**Children** Transaction

**Annotation** documentation The root element contain one or more msgs and acks

**Source**
```xml
<xs:element name="Root">
  <xs:annotation>
    <xs:documentation>The root element contain one or more msgs and acks</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Transaction" maxOccurs="unbounded">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="256"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Orig">
<xs:simpleType>
<xs:restriction base="xs:integer">
<xs:totalDigits value="6"/>
<xs:minExclusive value="99999"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Dest">
<xs:simpleType>
<xs:restriction base="xs:integer">
<xs:totalDigits value="6"/>
<xs:minExclusive value="99999"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="TimeStamp">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="[0-3][0-9]/[0-1][0-9]/[0-9][0-9][0-9][0-9][0-2][0-9][0-5][0-9][0-5][0-9]"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Type">
<xs:annotation>
<xs:documentation>51 = transaction is a msg ; 52,53 = transaction is an ack; 54 = transaction is a registration; 55 = transaction is a de-registration;</xs:documentation>
</xs:annotation>
<xs:simpleType>
<xs:restriction base="xs:nonNegativeInteger">
<xs:enumeration value="51"/>
<xs:enumeration value="52"/>
<xs:enumeration value="53"/>
<xs:enumeration value="54"/>
<xs:enumeration value="55"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Msg" minOccurs="0">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:maxLength value="160"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Mobile" minOccurs="0">
<xs:annotation>
<xs:documentation>mobile numbers must be in international format e.g. 003569123456</xs:documentation>
</xs:annotation>
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:length value="13"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Priority" minOccurs="0">
<xs:annotation>
<xs:documentation>0 = Normal Priority ; 1 = High Priority</xs:documentation>
</xs:annotation>
<xs:simpleType>
<xs:restriction base="xs:nonNegativeInteger">
<xs:pattern value="[0-1]"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="State" minOccurs="0">
<xs:annotation>
<xs:documentation>0 = Successful ; others = errors and other statuses</xs:documentation>
</xs:annotation>
</xs:element>
<xs:simpleType>
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="9999"/>
  </xs:restriction>
</xs:element>
<xs:element name="Subject" minOccurs="0">
  <xs:annotation>
    <xs:documentation>To be used when registering and de-registering</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="160"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

**element** PostData/Root/Transaction

**Diagram**

- **TransID**
  - Transaction Hash

- **Orig**
- **Dest**
- **TimeStamp**
- **Type**
  - S1 = transaction is a msg;
  - S2,S3 = transaction is an add; S4 = transaction is a registration; S5 = transaction is an invalid registration; S6 = transaction is a de-registration;

- **Msg**
- **Mobile**
  - Mobile numbers must be in international format e.g., 003519123456

- **Priority**
  - 0 = Normal Priority; 1 = High Priority

- **State**
  - 0 = Successful; others = error and other statuses

**Subject**
- To be used when registering and de-registering

**Children**
- TransID
- Orig
- Dest
- TimeStamp
- Type
- Msg
- Mobile
- Priority
- State
- Subject
<xs:element name="Transaction" maxOccurs="unbounded">
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:string">
        <xs:maxLength value="256"/>
        <xs:minLength value="1"/>
      </xs:restriction>
    </xs:complexType>
  </xs:element>
</xs:element>
<xs:element name="TransID">
  <xs:annotation>
    <xs:documentation>Transaction Hash</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="256"/>
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Orig">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="6"/>
      <xs:minExclusive value="999999"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Dest">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="6"/>
      <xs:minExclusive value="999999"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="TimeStamp">
  <xs:annotation>
    <xs:documentation>0035699123456</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="160"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Mobile">
  <xs:annotation>
    <xs:documentation>Mobile numbers must be in international format e.g. 0035699123456</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Priority">
  <xs:annotation>
    <xs:documentation>0 = Normal Priority ; 1 = High Priority</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Type">
  <xs:annotation>
    <xs:documentation>51 = transaction is a msg ; 52,53 = transaction is an ack; 54 = transaction is a registration; 55 = transaction is a de-registration;</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:nonNegativeInteger">
      <xs:enumeration value="51"/>
      <xs:enumeration value="52"/>
      <xs:enumeration value="53"/>
      <xs:enumeration value="54"/>
      <xs:enumeration value="55"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Msg" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="160"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Mobile" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Priority" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

### element PostData/Root/Transaction/TransID

**Diagram**

```
<xs:element name="TransID">
  <xs:annotation>
    <xs:documentation>Transaction Hash</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="256"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Type**  
restriction of xs:string

**Facets**

<table>
<thead>
<tr>
<th>facet</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>minLength</td>
<td>1</td>
</tr>
<tr>
<td>maxLength</td>
<td>256</td>
</tr>
</tbody>
</table>

**Annotation**

documentation: Transaction Hash

**Source**

```
<xs:element name="TransID">
  <xs:annotation>
    <xs:documentation>Transaction Hash</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="256"/>
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### element PostData/Root/Transaction/Orig

**Diagram**

```
<xs:element name="Orig">
  <xs:annotation>
    <xs:documentation>Transaction Hash</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minExclusive value="99999"/>
      <xs:totalDigits value="6"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Type**  
restriction of xs:integer

**Facets**

<table>
<thead>
<tr>
<th>facet</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>minExclusive</td>
<td>99999</td>
</tr>
<tr>
<td>totalDigits</td>
<td>6</td>
</tr>
</tbody>
</table>

**Source**

```
<xs:element name="Orig">
  <xs:annotation>
    <xs:documentation>Transaction Hash</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minExclusive value="99999"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
### element `PostData/Root/Transaction/Dest`

<table>
<thead>
<tr>
<th>diagram</th>
<th>postdata/root/transaction/dest</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:integer</code></td>
</tr>
<tr>
<td>facets</td>
<td><code>minExclusive</code> 99999</td>
</tr>
<tr>
<td></td>
<td><code>totalDigits</code> 6</td>
</tr>
</tbody>
</table>
| source  | <xs:element name="Dest">
|         |   <xs:simpleType>
|         |     <xs:restriction base="xs:integer">
|         |       <xs:totalDigits value="6"/>
|         |       <xs:minExclusive value="99999"/>
|         |     </xs:restriction>
|         |   </xs:simpleType>
|         | </xs:element> |

### element `PostData/Root/Transaction/TimeStamp`

<table>
<thead>
<tr>
<th>diagram</th>
<th>postdata/root/transaction/timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:string</code></td>
</tr>
<tr>
<td>facets</td>
<td><code>pattern</code> [0-3][0-9]/[0-1][0-9]/[0-9][0-9][0-9][0-9] [0-2][0-9][0-9][0-9][0-9][0-9]</td>
</tr>
</tbody>
</table>
| source  | <xs:element name="TimeStamp">
|         |   <xs:simpleType>
|         |     <xs:restriction base="xs:string">
|         |       <xs:pattern value="[0-3][0-9]/[0-1][0-9]/[0-9][0-9][0-9][0-9]/[0-2][0-9]/[0-9][0-9][0-9][0-9]/[0-9]/[0-9][0-9][0-9]/[0-9][0-9]/[0-9][0-9][0-9][0-9][0-9]"/>
|         |     </xs:restriction>
|         |   </xs:simpleType>
|         | </xs:element> |

### element `PostData/Root/Transaction/Type`

<table>
<thead>
<tr>
<th>diagram</th>
<th>postdata/root/transaction/type</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:nonNegativeInteger</code></td>
</tr>
</tbody>
</table>
| facets  | `enumeration` 51
|         | `enumeration` 52
|         | `enumeration` 53
|         | `enumeration` 54
|         | `enumeration` 55
| annotation | documentation 51 = transaction is a msg ; 52,53 = transaction is an ack ; 54 = transaction is a registration; 55 = transaction is a de-registration; |
| source  | <xs:element name="Type">
|         |   <xs:annotation>
|         |     <xs:documentation>51 = transaction is a msg ; 52,53 = transaction is an ack ; 54 = transaction is a registration; 55 = transaction is a de-registration;</xs:documentation>
|         |   </xs:annotation>
|         |   <xs:simpleType>
|         |     <xs:restriction base="xs:nonNegativeInteger">
|         |       <xs:enumeration value="51"/>
|         |       <xs:enumeration value="52"/>
|         |       <xs:enumeration value="53"/>
|         |       <xs:enumeration value="54"/>
|         |       <xs:enumeration value="55"/>
|         |   </xs:simpleType>
|         | </xs:element> |
### element `PostData/Root/Transaction/Msg`

**Diagram:**

```xml
<xs:element name="Msg" minOccurs="0">
   <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:maxLength value="160"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>
```

**Type:** restriction of `xs:string`

**Facets:** maxLength 160

**Source:**

```xml
<xs:element name="Msg" minOccurs="0">
   <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:maxLength value="160"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>
```

### element `PostData/Root/Transaction/Mobile`

**Diagram:**

Mobile numbers must be in international format e.g. 0035699123456

**Type:** restriction of `xs:string`

**Facets:** length 13

**Annotation:** mobile numbers must be in international format e.g. 0035699123456

**Source:**

```xml
<xs:element name="Mobile" minOccurs="0">
   <xs:annotation>
      <xs:documentation>mobile numbers must be in international format e.g. 0035699123456</xs:documentation>
   </xs:annotation>
   <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:length value="13"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>
```

### element `PostData/Root/Transaction/Priority`

**Diagram:**

`0 = Normal Priority ; 1 = High Priority`

**Type:** restriction of `xs:nonNegativeInteger`

**Facets:** pattern `[0-1]`

**Annotation:** 0 = Normal Priority ; 1 = High Priority

**Source:**

```xml
<xs:element name="Priority" minOccurs="0">
   <xs:annotation>
      <xs:documentation>0 = Normal Priority ; 1 = High Priority</xs:documentation>
   </xs:annotation>
   <xs:simpleType>
      <xs:restriction base="xs:nonNegativeInteger">
         <xs:pattern value="[0-1]"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>
```
Refer to Development Support web page http://edev.mita.gov.mt for the actual XSD