

Appendix B

ODF Processor Implementation

When messages are sent, the document type (retrieved from the OdfBody node) is checked to find out the type of message and then processing carried out through if/else statements e.g.

```
If (DocType == DT_SCHEDULE_UPDATE)
    Process
Else if (DocType == DT_RT_RESULT)
    Process
```

We also get the document code (the unique code identifying the event) from the OdfBody node as this is used in many of the database queries.

Revisions since Prototype

Before implementing processing for the new message types, some adjustments were needed to be made to the current implementation. Firstly, although the start order for the list of teams is populated when a start list is processed, the sort order remains null. This means that as results messages are processed and the database is updated, there will be a short period where the sort order is null. Therefore the ODF Processor has been edited so that when the start order is inserted into the database, the sort order is also updated so that all teams have a value for this field.

It was also realised that at no point in the original database schema has storing the current of an event been considered. This is a field that will be necessary in order to know which dive to be displayed for a team's result. The "round" field has therefore been added to the "events" table in the database. The value of this field is found in the DT_RT_RESULT messages, so will be updated as these messages are processed.

DT_SCHEDULE

```
<?xml version="1.0" encoding="UTF-8"?>
- <OdfBody Version="4" DocumentType="DT_SCHEDULE" LogicalDate="20120723" FeedFlag="P"
  Date="20120723" Time="045744240" Serial="457607" DocumentCode="DV0000000">
  - <Competition Code="OG2012">
    - <Discipline Code="DV">
      - <Gender Code="0">
        - <Event Code="000">
          - <Phase Code="N" Type="7">
            - <Unit Code="01" Location="DV1" Venue="AQC" Medal="0" EstimatedEndDate="false"
              EndDate="2012-07-28T15:00:00+01:00" EstimatedStartDate="false"
              StartDate="2012-07-28T14:00:00+01:00" Status="2">
              <EstimatedStartText Value="" Language="ENG"/>
              <EstimatedStartText Value="" Language="FRA"/>
            </Unit>
          </Phase>
        </Event>
      </Gender>
    - <Gender Code="M">
      - <Event Code="001">
        - <Phase Code="1" Type="3">
          - <Unit Code="01" Location="DV1" Venue="AQC" Medal="1" EstimatedEndDate="false"
            EndDate="2012-08-07T20:08:00+01:00" EstimatedStartDate="false"
            StartDate="2012-08-07T19:00:00+01:00" Status="2">
            <EstimatedStartText Value="" Language="ENG"/>
            <EstimatedStartText Value="" Language="FRA"/>
          </Unit>
        </Phase>
      - <Phase Code="2" Type="3">
        - <Unit Code="01" Location="DV1" Venue="AQC" Medal="0" EstimatedEndDate="false"
          EndDate="2012-08-07T11:36:00+01:00" EstimatedStartDate="false"
```

DT_SCHEDULE message structure

Pseudo Code

```
Go to competition node
Go to discipline node
Get discipline code
For each gender node
  Get gender code
  For each event node
    Get event code
    For each phase node
      Get phase code
      Go to unit node
      Get unit code
      Event ID = discipline + gender + event + phase + unit
      Get start date, end date, start time, end time, status
      Insert into database
    End For
  End For
End For
```

SQL

```
INSERT INTO EVENTS (code,date,start_time,end_time,status)
VALUES ('"+eventID+"', '"+startDate+"', '"+startTime+"', '"+endTime+"', '"+status
+')')
```

To process the DT_SCHEDULE message, we move to the discipline node and get the discipline code. We then loop through each gender, each event and each phase, storing their respective code to

strings. Each phase has one unit, so we go to this node and get the unit code as well as the event details. The start/end date and times are stores as a single string, so we perform a substring operation to return these as separate strings. We also get the status of the event- an integer representing whether the event is scheduled, in progress etc. By combining the discipline, gender, event, phase and unit codes into a single string, we get a unique code for each event e.g. DVM202101 being the Men's 10m Synchronised Final (these codes are in accordance with the ODF documentation and were used at London 2012).

DT_SCHEDULE_UPDATE

```
<?xml version="1.0" encoding="UTF-8"?>
- <OdfBody Serial="2" Version="10" LogicalDate="20120801" Time="145532081" Date="20120801"
FeedFlag="P" DocumentType="DT_SCHEDULE_UPDATE" DocumentCode="DV0000000">
  - <Competition Code="OG2012">
    - <Discipline Code="DV">
      - <Gender Code="M">
        - <Event Code="201">
          - <Phase Code="1" Type="3">
            <Unit Code="01" ModificationIndicator="U" Location="DV1" Venue="AQC" Medal="1"
            EstimatedEndDate="false" EndDate="2012-08-01T15:48:00+01:00"
            EstimatedStartDate="false" StartDate="2012-08-01T15:00:00+01:00"
            Status="4"/>
          </Phase>
        </Event>
      </Gender>
    </Discipline>
  </Competition>
</OdfBody>
```

DT_SCHEDULE_UPDATE message structure

Pseudo Code

```
Go to competition node
Go to discipline node
Get discipline code
For each gender node
  Get gender code
  For each event node
    Get event code
    For each phase node
      Get phase code
      Go to unit node
      Get unit code
      Event ID = discipline + gender + event + phase + unit
      Get status
      Update status in database
    End For
  End For
End For
```

SQL

```
"UPDATE EVENTS SET status='"+status+"' WHERE code='"+eventID+'"
```

The DT_SCHEDULE_UPDATE message has the same structure as the DT_SCHEDULE message, except it only includes information on events that need their status updating. So as before we loop through the nodes to create a string for the event ID, and then get the status and update it in the database.

DT_RT_RESULT

```
Time="150317933" Date="20120801" DocumentType="DT_RT_RESULT">
- <Competition Code="OG2012">
- <UnitInfos>
  <UnitDateTime EndDate="2012-08-01T15:48:00+01:00" StartDate="2012-08-01T15:00:00+01:00"/>
  <UnitInfo Code="DV_ROUND" Value="1" Type="UI_DV"/>
</UnitInfos>
- <Result SortOrder="2">
- <Competitor Code="DVM201USA01" Type="T">
- <ExtendedResults>
  <ExtendedResult Code="DV_CURRENT" Value="N" Type="ER_DV"/>
  <ExtendedResult Code="DV_SCR" Value="51.60" Type="ER_RESULTS" Pos="1">
- <Extensions>
  <Extension Code="DV_IDX" Value="2" Type="DV_SCR"/>
</Extensions>
</ExtendedResult>
</ExtendedResults>
- <Composition>
  <Athlete Code="1132778" Order="1"/>
  <Athlete Code="1132374" Order="2"/>
</Composition>
</Competitor>
</Result>
- <Result SortOrder="3">
- <Competitor Code="DVM201GBR01" Type="T">
- <ExtendedResults>
  <ExtendedResult Code="DV_CURRENT" Value="Y" Type="ER_DV"/>
  <ExtendedResult Code="DV_SCR" Type="ER_RESULTS" Pos="1">
- <Extensions>
  <Extension Code="DV_AVR_H" Value="1" Type="DV_SCR" Pos="1"/>
  <Extension Code="DV_AVR" Value="9.0" Type="DV_SCR" Pos="1"/>
  <Extension Code="DV_AVR_H" Value="2" Type="DV_SCR" Pos="2"/>
  <Extension Code="DV_AVR" Value="8.6" Type="DV_SCR" Pos="2"/>
  <Extension Code="DV_AVR_H" Value="3" Type="DV_SCR" Pos="3"/>
  <Extension Code="DV_AVR" Value="-" Type="DV_SCR" Pos="3"/>
  <Extension Code="DV_IDX" Value="3" Type="DV_SCR"/>
</Extensions>
</ExtendedResult>
</ExtendedResults>
- <Composition>
  <Athlete Code="1070024" Order="1"/>
```

DT_RT_RESULT message structure

Pseudo Code

Go to competition node

For each child node

 If node = UnitInfos

 Go to UnitInfo node

 Get DV_ROUND value

 Update round in database

 End If

 Else if node = Result

 Get Rank, RankEqual, SortOrder, Total

 Go to competitor node

 Get competitor code

 If no rank attribute

 Update SortOrder in database

 End If

 Else If no Total update

 Update Rank, RankEqual, SortOrder in database

 End If

 Else

 Update Rank, RankEqual, SortOrder, Total in database

 End Else

```

        If child node exists
            Go to ExtendedResults node
            Go to ExtendedResult node
            If DV_CURRENT = Y
                Update current in database
            End If
            Else If DV_PREVIOUS = Y
                Update previous in database
                Move to ExtendedResult DV_DIVE node
                Get Value and Pos
                Move to Extensions node
                Move to Extension node
                Get judge's scores, penalty, average
                Update dive scores in database
            End If
            If DV_CURRENT = N
                Update current to NULL in database
            End If
            Else If DV_PREVIOUS = N
                Update previous to NULL in database
            End If
        End If
    End If
End For

```

To process the DT_RT_RESULT, we move to the competition node and then loop through its children. If the child is a UnitInfos node, then we get the value for the current round and update this in the database.

```
UPDATE events SET round='"+round+"' WHERE code='"+DocCode+"'
```

If the node is a result node, then we get and update the results (rank, if the rank is equal, total and sort order), then move to the competitor node to get the team code. As per the ODF definition, rank, rank equal and total are not mandatory attributes so not always present as part of the result node. Therefore, we use if statements to determine if these attributes are present and update the appropriate information in the database.

```

UPDATE RESULTS SET sort_order='"+sortOrder+"' WHERE competitor='"+teamID+"'
UPDATE RESULTS SET
sort_order='"+sortOrder"',rank='"+rank"',equal_rank='"+rankEqual+"' WHERE
competitor='"+teamID+"'
UPDATE RESULTS SET
sort_order='"+sortOrder"',rank='"+rank"',equal_rank='"+rankEqual"',total
='"+totalPoints+"' WHERE competitor='"+teamID+"'

```

We now check if the competitor node has any children. If it does, this is where the current and previous flags with detailed results information are located. If the flag is “DV_CURRENT = Y” then we know to update the current competitor in the database

```
UPDATE RESULTS SET current='X' WHERE competitor='"+teamID+"'
```

If the flag is “DV_PREVIOUS =Y” then we know to update the previous competitor and also to update the dive scores. We move to the extended result node and get the dive score and the dive number from DV_DIVE. We then move to the extension nodes and get the individual scores by each judge, the penalty, and the average score and update these values in the database.

```
UPDATE RESULTS SET previous='X' WHERE competitor='"+teamID+''
```

```
UPDATE DIVES SET E1='"+E1+',E2='"+E2+',E3='"+E3+',E4='"+E4+',
E5='"+E5+',E6='"+E6+',S1='"+S1+',S2='"+S2+',S3='"+S3+',S4='"+S4+',
S5='"+S5+',pen='"+penalty+',avg='"+average+',score='"+diveScore+'
WHERE competitor='"+teamID+'
AND dive_number='"+diveNo+''
```

If the flag is “DV_CURRENT = N” or “DV_PREVIOUS=N” then we remove the flag from the database by setting the field for that team back to NULL.

```
UPDATE RESULTS SET current=NULL WHERE competitor='"+teamID+'
UPDATE RESULTS SET previous=NULL WHERE competitor='"+teamID+''
```

DT_MEDALLISTS

```
<?xml version="1.0" encoding="UTF-8"?>
- <OdfBody Serial="14" Version="1" DocumentCode="DVM201000" FeedFlag="P" Language="ENG" Venue="AQC"
  LogicalDate="20120801" Time="155557431" Date="20120801" DocumentType="DT_MEDALLISTS">
  - <Competition Code="OG2012">
    - <Medal Code="ME_GOLD" Unit="01" Phase="1">
      - <Competitor Code="DVM201CHN01" Order="1" Type="T">
        - <Composition>
          <Athlete Code="1072157" Order="1"/>
          <Athlete Code="1072156" Order="2"/>
        </Composition>
      </Competitor>
    </Medal>
    - <Medal Code="ME_SILVER" Unit="01" Phase="1">
      - <Competitor Code="DVM201RUS01" Order="1" Type="T">
        - <Composition>
          <Athlete Code="1016610" Order="1"/>
          <Athlete Code="1016607" Order="2"/>
        </Composition>
      </Competitor>
    </Medal>
    - <Medal Code="ME_BRONZE" Unit="01" Phase="1">
      - <Competitor Code="DVM201USA01" Order="1" Type="T">
        - <Composition>
          <Athlete Code="1132778" Order="1"/>
          <Athlete Code="1132374" Order="2"/>
        </Composition>
      </Competitor>
    </Medal>
  </Competition>
</OdfBody>
```

DT_MEDALLISTS message structure

Pseudo Code

```
Go to competition node
For each medal node
  Get medal code
  Medal name = substring of medal code
  Go to competitor node
  Get team code
  Update medal field in database
```

End For

To process the DT_MEDALLISTS message, we got to the competition node, then loop through each medal node. We get the medal name by performing a substring operation on the medal code. We then move to the competitor node to get the team code and update the medal field in the database.

```
UPDATE RESULTS SET medal='"+medal+"' WHERE competitor='"+teamMedalCode+"'
```