

**BRAMS to BaSE SAP WebService**

**Business Layer to Integration Layer Design**

**Version 0.4**

**March 2014**



1.0 Introduction 2

2.0 High Level Requirements 2

3.0 Network Changes 3

3.1 Details of data being tracked 3

3.2 Data Output from BRAMS 5

3.3 Implementation Details 8

3.4 Error Trapping 10

4.0 Documentation Requirements 11

5.0 Assumptions 11

6.0 Conclusion 11

Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Changed by | Notes |
| March 2014 | 0.1 | JMM | Initial Revision |
|  | 0.4 | RE, ZS,JMM | Merged corrections |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Reference Documents:

High Level Design – v1.0: A high level overview of the integration requirements.

# Introduction

# High Level Requirements

The high level Design document established that the main objected of this project is to provide some level of integration between the BRAMS Exor System and the BaSE SAP System. The reason for doing this is to ensure that the linear assets in BRAMS and certain characteristics are updated to BaSE SAP with the use of a WebService.

During high level discussions it was determined that certain characteristics should be tracked and updated for the network types of: Road, Footpath, Kerb and Channel, and Median.

The characteristics that need to be tracked are:

* Condition\_suburb\_level
* Ward\_name
* Region
* Asset\_owner
* Suburb\_name
* Paved\_surface\_material\_source (only for footpath)

If any characteristic or the network has changes all of the information will be returned in the output.

# Data Input

The pre-requisite for the Webservice are

* IIS 7.5: Version of the Web Server
* Web Deploy 3.5: Required for automatic deployment of the WebService
* Oracle Data Access Components for Windows 64-bit: Required to communicate between the Web Server and the BRAMS Database

It is envisaged that the WebService can be called by any SOAP consumer module and there will be only ONE method to call that returns the changes in BRAMS.

The input parameters to the method GetChangesRequest are as follows

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Format** | **Type/Length** | **Optional/**  **Mandatory** |
| Start Date | DATE | YYYY-MM-DD\* | Mandatory |
| End Date | DATE | YYYY-MM-DD | Mandatory |
| BRAMS ID | Number | 10 | Optional |

\*The data format must be in the ISO 8601 (i.e. YYYY-MM-DD) for use in XML. The governing body on XML W3C recommends this format.

An example of the request is a follows

<soapenv:Envelope xmlns:soapenv="<http://schemas.xmlsoap.org/soap/envelope/>" xmlns:tem="<http://tempuri.org/>">

   <soapenv:Header/>

   <soapenv:Body>

      <tem:GetChangesRequest>

         <tem:StartDate>2014-03-28</tem:StartDate>

         <tem:EndDate>2014-03-29</tem:EndDate>

         <tem:BramsId>1234512345</tem:BramsId>

      </tem:GetChangesRequest>

   </soapenv:Body>

</soapenv:Envelope

Once the Web Services receives the request for method call, it will call the tailor made PL/SQL package XBCC\_SAP\_SYNC and pass in the parameters listed above.

# Network Changes

## Details of data being tracked

The Exor System needs be able to provide a list of changes to various networks and Characteristics of those networks. The Network types being tracked are:

|  |  |  |
| --- | --- | --- |
| **Network Name** | **BRAMS Exor**  **Internal Name** | **Comments** |
| Road | ROCO |  |
| Footpath | VECO |  |
| Kerb and Channel | KCOR |  |
| Median | MED\* | \*This Network does not physically exist in the BRAMS Exor system. It is a “Virtual” copy of the Road Network with 999 appended to the beginning of the BRAMS Object ID |

The Characteristics of each of the above network types being tracked are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Characteristic Name** | **BRAMS Exor**  **Asset Name** | **Column** | **XSP if applicable** | **Comments** |
| Condition\_suburb\_level | N/A |  |  | Not in BRAMS |
| Ward\_name | OPWD | Ward name or number, exact unknown |  |  |
| Region | OPWD | LAS Region |  |  |
| Asset\_owner | ASOW | Asset Owner |  |  |
| Suburb\_name |  |  |  |  |
| Paved\_surface\_material\_source | PAVE | Paved Surface Material Source’ |  | (only for footpath)  Might not be needed |

Note: If the BRAMS Exor Asset Name is blank that means that the characteristic is part of the network at the lowest level and retrievable from there.

## Data Output from BRAMS

The output data should contain the following information:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Size** | **Description** |
| INDICATOR | CHAR (upper) | 1 | Values include   * I   + This indicates that the Item listed in that row was inserted into BRAMS * D   + This indicates that the Item listed in that row was End Dated in BRAMS and therefore no longer logically exists * C   + This indicates that the Item listed in that something in for the record in this row was changed in BRAMS |
| BRAMS ID | Number | 10 | Network ID for the system |
| OBJECT | CHAR (upper) | 30 | This includes the object/asset type, such as ROAD, FOOTPATH, KERB and MEDIAN, or the Characteristic type for the objects. See above section for list of characteristics |
| NAME | CHAR(upper) | 30 | This is the value of the characteristic that has changed |
| START | NUMBER | 18,2 | This is start point of the asset ID. The unit will as specified in BRAMS |
| END | NUMBER | 18,2 | This is the end point of the asset ID. The unit will be as specified in BRAMS |

The Start and End Columns will contain the homogeneous locations of characteristics along the object corridor. i.e. neighboring characteristics holding the same value will be combined and reported only once.

When the program is executed it will determine which network had network or characteristic changes to them and output rows that correspond to the changes.

A sample of the type of data to be found in the output would be:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Indicator** | **BRAMS\_ID** | **OBJECT** | **NAME** | **START** | **END** |
| I | 1234512345 | ROAD | ANN STREET | 0 | 300 |
| I | 1234512345 | SUBURB\_NAME | CBD | 0 | 200 |
| I | 1234512345 | SUBURB\_NAME | NEW FARM | 200 | 300 |
| I | 1234512345 | ASSET\_OWNER | BCC | 0 | 250 |
| I | 1234512345 | ASSET\_OWNER | MAINROADS | 250 | 300 |
| C | 5432154321 | ROAD | ROSE STREET | 0 | 300 |
| C | 5432154321 | SUBURB\_NAME | CBD | 0 | 200 |
| C | 5432154321 | SUBURB\_NAME | NEW FARM | 200 | 300 |
| C | 5432154321 | ASSET\_OWNER | BCC | 0 | 250 |
| C | 5432154321 | ASSET\_OWNER | MAINROADS | 250 | 300 |
| D | 9876543210 | ROAD | PETAL STREET |  |  |

Although not shown in the sample, if the indicator is insert (I) or change (C) for a given BRAMS\_ID, there would be a record for each characteristic that has been populated in BRAMS. A delete (D) would should 1 record as shown in the example.

## Implementation Details

The program needs to be able to determine if any tracked items have changed between two dates. Then return the BRAMS Corridor ID and all corresponding characteristics regardless of what has actually changed.

The following are the input parameters for the program:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Format** | **Type/Length** | **Mandatory** | **Comments** |
| Start Date | DATE | DD-MMM-YYYY | Y | Modified or Effective Date |
| End Date | DATE | DD-MMM-YYYY | Y | Modified or Effective Date |
| BRAMS ID | DATE | 10,0 | N | BRAMS Primary Corridor ID, This is a valid ID for a corridor listed in 3.1 |

The general logic of the program will be:

Compile a list of Corridor IDs that had a change occur to them during the inputted date range. Valid BRAMS changes are Insert (I), Updated (C), or End Dated (D). To make this list first the Corridor data is searched for Modified Dates and End Dates in our date range, those items are added directly to the list.

Next Characteristics that are of a group type are scanned for searched for Modified Dates and End Dates in our date range. Once, the records are identified they are translated to a Corridor ID and added to the list.

Next Characteristics that are of an asset type are scanned for searched for Modified Dates and End Dates in our date range. Once, the records are identified they are translated to a Corridor ID and added to the list.

Once the list of Changed Corridor IDs is found, the program will construct the output based on Unique IDs within that list. If during this construction process, the corridor is End Dated the Delete Output (D) type will be invoked and the Characteristics skipped. For the Change Output (C) and the Inserted Output (I) the program will continue to fill in the corresponding characteristics.

If the BRAMS ID parameter is supplied:

The program will only list the information for that BRAMS ID and characteristics if a tracked item has changed. In the unlikely case that the date range includes the time when the corridor was inserted a Changed Output (C) will be listed for that item and not an Insert Output (I).

When output data produced, there can be three different values in the indicator column. These indicators display what operation has happened during the date range.

* Inserted Output
  + If a new object/characteristic is added and effective during the date range and the optional BRAMS\_ID parameter was not used then the following data is filled in.
    - INDICATOR: I
    - BRAMS\_ID: The current BRAMS Corridor ID
    - OBJECT: Current record type, Corridor type/ characteristic type
    - NAME: This is the value of the characteristic that has changed
    - START: This is start point of the item.
    - END: This is end point of the item.
* Changed Output
  + If an object/characteristic is modified and effective during the date range and has not been end dated during the date range.
    - INDICATOR: C
    - BRAMS\_ID: The current BRAMS Corridor ID
    - OBJECT: Current record type, Corridor type/ characteristic type
    - NAME: This is the value of the characteristic that has changed
    - START: This is start point of the item.
    - END: This is end point of the item.
* Deleted Output
  + If a new object/characteristic is end dated during the date range. In this case only the Corridor Record is outputted since all other characteristics are automatically closed/end dated.
    - INDICATOR: D
    - BRAMS\_ID: The current BRAMS Corridor ID
    - OBJECT: Current record type, Corridor type/ characteristic type
    - NAME: This is the value of the characteristic that has changed
    - START: null
    - END: null
* Look for date modified and end dates for corridors between the start and end dates
  + Add BRAMS corridor ID to list
* Look for date modified and end dates for characteristics between the start and end dates
  + If found get the BRAMS corridor ID and add to list
* Take a “distinct” look at the list to get a list of corridors
* Build the Corridor data and the characteristics “table”

## Process Flow

The general process flow would be:

* Determine all the occurrences of modified Corridors in NM\_ELEMENTS\_ALL that fall within the specified date range. These BRAMS IDs will then be added to a list. By using Various Date Fields in NM\_ELEMENTS\_ALL, the proper indicator value will also be placed in the list.
* Determine all the occurrences of an insert or end date of an object (inventory item or group item) relating to a characteristic in NM\_MEMBERS\_ALL. This implies a change to a characteristic. NM\_MEMBERS\_ALL will identify the affected datum which can be used to determine the affected corridor. Determine the ID of the related corridor and add them to the list if they do not already exist. Indicators for characteristics will always be the change (C) type.
* If the optional BRAMS\_ID parameter was supplied then remove all other items form the list.
  + If no items remain in the list then nothing had changed and null is returned.
* If the corridor has a delete indicator (D) then place the data into the output list
* If the corridor has an Insert (I) or change (C) indicator then generate a full set of characteristic information and add it to the output list.
  + If the corridor has and Inset (I) indicator, then update all characteristics to have an Insert (I) indicator.
* Return the final data set to the WebServer for delivery.

## Error Trapping

Currently the types of anticipated error trapping include:

1. START date is less than END Date: The system will check that START Date should be greater than the END Date
2. BRAMS ID does not exist.
3. Catch-all: A general error trap message will also be communicated. Where possible it will include standard Oracle error messages, which will help for debugging.

# Data Output

The output of the data will have the following structure

<s:Envelope xmlns:s="<http://schemas.xmlsoap.org/soap/envelope/>">

   <s:Body xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xmlns:xsd="<http://www.w3.org/2001/XMLSchema>">

      <GetChangesResponse xmlns="<http://tempuri.org/>">

         <TOPITEM BRAMSID="1234512345">

            <ITEM>

               <INDICATOR>D</INDICATOR>

<BRAMSID>1234512345</BRAMSID>

               <OBJECT>ROAD</OBJECT>

               <NAME>NAME4</NAME>

               <START>0</START>

               <END>200</END>

            </ITEM>

            <ITEM>

         </TOPITEM>

         <TOPITEM BRAMSID="5678956789">

            <ITEM>

               <INDICATOR>C</INDICATOR>

               <BRAMSID>5678956789</BRAMSID>

               <OBJECT>ROAD</OBJECT>

               <NAME>NAME4</NAME>

               <START>0</START>

               <END>200</END>

            </ITEM>

               <INDICATOR>C</INDICATOR>

<BRAMSID>5678956789</BRAMSID>

               <OBJECT>SUBURB\_NAME</OBJECT>

               <NAME> CBD </NAME>

               <START>0</START>

               <END>200</END>

            </ITEM>

            <ITEM>

               <INDICATOR>C</INDICATOR>

<BRAMSID>5678956789</BRAMSID>

               <OBJECT>ASSET\_OWNER</OBJECT>

               <NAME> MAINROADS </NAME>

               <START>0</START>

               <END>200</END>

            </ITEM>

       </TOPITEM>

         <TOPITEM BRAMSID="2345623456">

            <ITEM>

               <INDICATOR>I</INDICATOR>

<BRAMSID> 2345623456 </BRAMSID>

               <OBJECT>ROAD</OBJECT>

               <NAME>NAME4</NAME>

               <START>0</START>

               <END>200</END>

            </ITEM>

               <INDICATOR>I</INDICATOR>

<BRAMSID> 2345623456 </BRAMSID>

               <OBJECT>SUBURB\_NAME</OBJECT>

               <NAME> CBD </NAME>

               <START>0</START>

               <END>200</END>

            </ITEM>

            <ITEM>

               <INDICATOR>I</INDICATOR>

<BRAMSID> 2345623456 </BRAMSID>

               <OBJECT>ASSET\_OWNER</OBJECT>

               <NAME> MAINROADS </NAME>

               <START>0</START>

               <END>200</END>

            </ITEM>

       </TOPITEM>

     </GetChangesResponse>

   </s:Body>

</s:Envelope>

The output of each item is as per the description is Section 4.2. Each item is however grouped within a XML tag identified as the TOPITEM. It enables to group all the characteristics of a given Object with an added advantage to preview and debug the results.

[[ WSDL to be included – When Valdas provides the correct format]

# Documentation Requirements

# Assumptions

# Conclusion

This document is the result of a series of conversations between Bentley Systems, Brisbane City Council with the objective of establishing an interface between BaSE SAP and the BRAMS Exor used to manage the road network and key asset data. From these discussions, Bentley Systems has created a software design that meets the agreed scope of this project and the requirements that need to be met in order for the project to be successful.

Using this design, Bentley Systems will create an estimate for the development and deployment of the software that details how the interface software will meet the requirements and an estimate of effort required to complete the project.