




**High Level Design**  
**BRAMS to BaSE SAP WebService**


	High Level Design – BRAMS to BaSE SAP	
	XML WebService	V1.0
	Zubran Solaiman	Page 2 of 11

# 1 Table of Contents

2	Version Control and Distribution .....	3
3	Background.....	4
4	Top Level Architecture Model.....	4
5	Data Flow between BRAMS and BaSE SAP .....	6
5.1	Data Input parameters to BRAMS.....	6
5.2	Business Logic.....	7
6	Data Output from BRAMS.....	10
7	Testing WebServices.....	11

## 2 Version Control and Distribution

Version	Source	Date	Description
0.1	Zubran Solaiman	14-Mar-2014	First draft
0.2	Zubran Solaiman	17-Mar-2014	Updates from review comments
0.3	Zubran Solaiman	18-Mar-2014	Updates from review comments
0.4	Richard Ellis	20 Mar-2014	Update after further discussion with BCC
1.0	Tracey Loynes	24-Mar-2014	Final Version accepted by BCC
<b>For review:</b>			<b>For internal approval:</b>
Richard Ellis  Tracey Loynes			Richard Ellis

	High Level Design – BRAMS to BaSE SAP	
	XML WebService	V1.0
	Zubran Solaiman	Page 4 of 11

### 3 Background

As per Annexure 1 to Customer Contract Four (4), there is a request to integrate between the BRAMS Exor System and BaSE SAP System. The main business driver is to ensure that linear assets in BRAMS and their characteristics are seamlessly updated to BaSE SAP through the utilisation of a WebService. This will provide key stakeholders of BaSE SAP System with up-to-date information of BRAMS. The proposed business functions and design are defined in the following documents

1. AP360 Synchronisation Functional Design INTO14a BRAMS to SAP (Council Service Desktop) BaSE Program
2. AP481 Interface Agreement – INTO41a – BRAMS


The objective of this document is to provide a high level design of the solution to meet the business functions described in the above mentioned documents. As the intention is to provide a WebService that is closely integrated with BRAMS, the high-level design will be based on underlying Exor Product and technology. It will focus only on the topics:

- Necessary parameters and data input to BRAMS
- Necessary output results (BaSe Program SAP requirements)
- A high-level description of the WebService

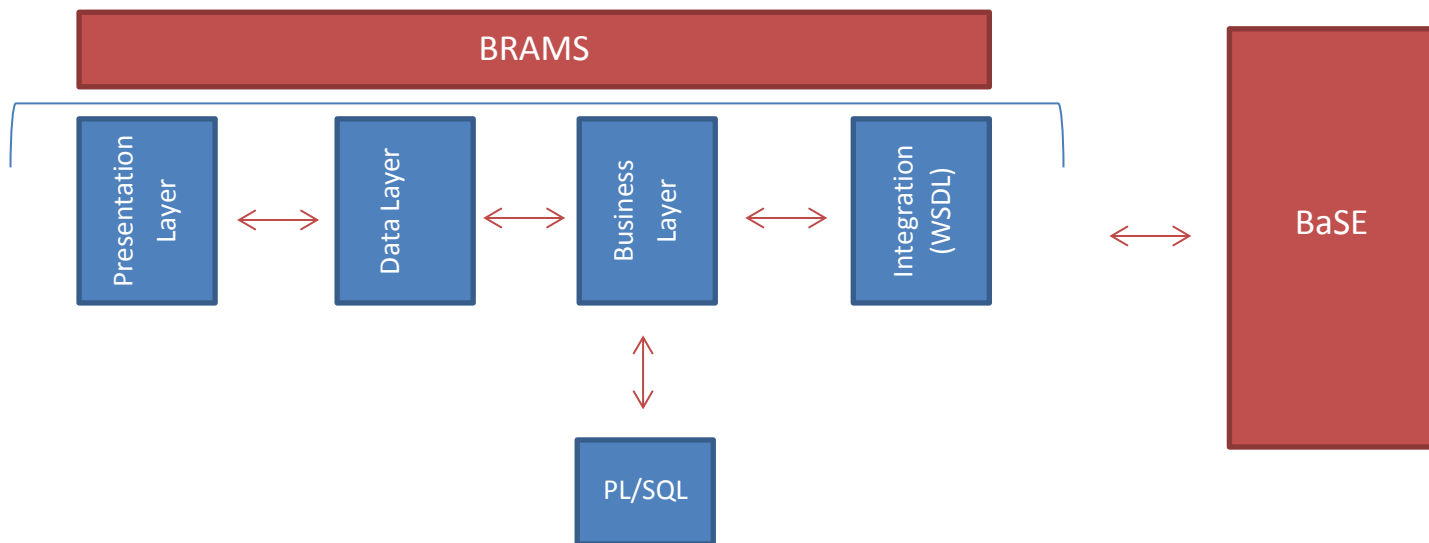
### 4 Top Level Architecture Model

It is envisaged that the SAP will call an XML WebService via HTTPS protocol and returns the data in XML. Thus, the WebService will act as a service provider and will expose a service oriented interface which will be invoked by request from the BaSE SAP System.

The inbound call to BRAMS will only query the necessary business layer based on the input parameters of the WebService. In order to provide a standard, universally recognised and supported programmatic interface, a Web Services

	High Level Design – BRAMS to BaSE SAP	
	XML WebService	V1.0
	Zubran Solaiman	Page 5 of 11

Description Language (WSDL) document will be provided in the technical specification. The WSDL provides standard endpoint description between BRAMS and BaSE SAP System.




**Presentation layer:** The presentation layer constitute the forms, where the BRAMS owner enters/loads/update the BRAMS record. It can be through a variety of forms, such as Spatial Manager, Forms, CSV loaders etc.

**Data Layer:** This is the main repository for the BRAMS data. Typically this includes the Commercial-off-the-shelf (COTS) objects, such as the asset tables.

**Business layer:** This constitutes Oracle translation views based on the asset data. The translation contains the necessary information for the WebService to return the relevant data. It is envisaged that each of the following objects (i.e. asset types) will have a corresponding translation view in the Business layer:

- ROADS
- FOOTPATH
- KERBS and CHANNELS
- MEDIAN

	<b>High Level Design – BRAMS to BaSE SAP</b>	
	<b>XML WebService</b>	V1.0
	Zubran Solaiman	Page 6 of 11

**Integration Layer:** WSDL for providing the service interface between BRAMS and BaSE SAP System.

**PL/SQL:** As part of the development process, PL/SQL procedures will initially be created to verify the business logic and perform unit testing. Once verified, the PL/SQL procedure will be wrapped as a WebService.

## 5 Data Flow between BRAMS and BaSE SAP


As mentioned above, the request for the data will be a XML WebService. This section will describe the necessary components in the following order:

- Data input parameters to BRAMS
- Business Logic
- Data output results from BRAMS

### 5.1 Data Input parameters to BRAMS

The following parameters will be passed into the WebService

Parameter	Format	Type/Length	Optional/ Mandatory	Description
Start Date	DATE	DD.MM.YYYY	Mandatory	Modified or Effective Date
End Date	DATE	DD.MM.YYYY	Mandatory	Modified or Effective Date
BRAMS ID	Number	10,0	Optional	BRAMS Primary ID (This is the nominated Asset ID column)

	High Level Design – BRAMS to BaSE SAP	
	XML WebService	V1.0
	Zubran Solaiman	Page 7 of 11

## 5.2 Business Logic


The general logic of the WebService is to identify changes within BRAMS where modified or effective fit with the data range specified (i.e. between start date and end date). It will primarily look at changes in the following object (asset) types

- Road
- Footpath
- Kerb and Channel
- Median

And, identify where changes have occurred to their corresponding characteristics for the above mentioned object types. The characteristics are as follows:

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

It is unclear if each object has the above mentioned characteristics in BRAMS and it is envisaged that this will be finalised in readiness for the Technical Design Document. The above characteristics should be return on the object types (asset types) Mentioned above. As described at meeting on 4<sup>th</sup> and 11<sup>th</sup> March, changes to an object in BRAMS are recorded at asset level, rather than at characteristic level. Consequently it is not possible to identify which characteristic will have changed for a given object. Therefore it has been agreed that the response from BRAMS will present all characteristics as listed above.

 <b>Bentley®</b>	<b>High Level Design – BRAMS to BaSE SAP</b>	
	<b>XML WebService</b>	V1.0
	Zubran Solaiman	Page 8 of 11

As the BRAMS ID is optional, the following responses are returned

1. When the BRAMS ID is null


The WebService will examine the objects listed above for any changes within the specified date range and will return the following indicators

- a. Return an indicator of 'I': A new object/asset is identified by the WebService. The EFFECTIVE date should be within the date range. If the Asset has been created within the date range. Then all the characteristics for that asset will also have an indicator of 'I'

Note that BRAMS may not be able to distinguish between an insert of a characteristic and the change of a characteristic so the code will force all changes to characteristics of assets created or modified within the date range to have an indicator of 'I'.

- b. Return an indicator of 'D': An object/asset has been deleted from operational use (i.e. closed). The ASSET END date is within the date range. It will only respond with the Asset ID of the object that has been end-dated because by default all the characteristics will also be closed.
- c. It will return an indicator of 'C': An object/asset has been modified either at asset level or at characteristic level. The ASSET END Date value should be NULL, the MODIFIED date is within the date range. Similar to the above, it will respond with the Asset ID and all the relevant characteristics. BaSE should remove all records relating to this asset ID and replace them with the information provided.



 <b>Bentley®</b>	<b>High Level Design – BRAMS to BaSE SAP</b>	
	<b>XML WebService</b>	V1.0
	Zubran Solaiman	Page 9 of 11

## 2. When the BRAMS ID is populated

The WebService will examine the objects listed above for any changes for that BRAMS ID within the specified date range and will return the following indicators

- a. Return an indicator of 'D': An asset has been deleted from operational use (i.e. closed). The ASSET END date is within the date range. It will only respond with the Asset ID of the object that has been end-dated because by default all the characteristics will also been closed.
- b. It will return an indicator of 'C': An object/asset has been modified either at asset level or at characteristic level. The ASSET END Date value should be NULL, the MODIFIED date is within the date range. Similar to the above, it will respond with the Asset ID and all the relevant characteristics. BaSE should remove all records relating to this asset ID and replace them with the information provided.


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 do not exist: if the BRAMS ID is populated in the input parameter, then the WebService will check will verify the BRAMS ID in the system.
3. Format check: The WebService should check if the input parameters are valid
4. 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.

## 6 Data Output from BRAMS

The following results will be returned as XML

Field Name	Type	Size	Description
<XML_TAG>			
<INDICATOR>	CHAR (upper)	1	Values include I, D and C
<BRAMS ID>	Number	10	Asset 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

	High Level Design – BRAMS to BaSE SAP	
	XML WebService	V1.0
	Zubran Solaiman	Page 11 of 11

## 7 Testing WebServices

A WebService test stub will be provided with hard coded values, to enable parallel development of the BaSE SAP Functional requirement as described in Section 3.1 of AP360 Integration Functional Design.

Another efficient means to test a non-instructive WebService is by using an open source solution like SOAP UI (<http://www.soapui.org/>). Bentley has previously used to verify XML WebServices both on test as well as production environments.