

11g TO 12c MIGRATION IN ORACLE BAM 12.1.3

ORACLE WHITE PAPER | DECEMBER 2014





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



Table of Contents

Disclaimer	1
Introduction	2
BAM 11g DO Definition Upgrade to 12c	2
11g Simple Data Object Upgrade	2
2.2 11g FACT Data Object Upgrade	4
11g External Data Object Upgrade	8
11g Security Filter Upgrade	12
BAM 11g EMS Upgrade to 12c	14
BAM 11g DO Data Migration	17

Introduction

The Oracle BAM 12c command line utility – BAMCommand, supports the upgrade and migration of 11g artifacts into the 12c environment. The import operation with the -upgrade parameter is used to upgrade the Oracle BAM 11g artifacts (DO and EMS) metadata into Oracle BAM 12c. At the time of metadata upgrade, some information is modified. The import operation with the -migrate parameter is used to migrate Oracle BAM 11g Data Object data into Oracle BAM 12c.

BAM 11g DO Definition Upgrade to 12c

11g Simple Data Object Upgrade

The BAMCommand utility is used to upgrade the 11g Data Object metadata to 12c.

The “-upgrade” option from BAMCommand is used to import the 11g Data Object definition to 12c.

The Command: `./bamcommand -cmd import -file <11g DO XML file> -upgrade 1`

The following rules apply to 11g Simple Data Object upgrade process.

1. The 11g Data Object ID is converted to a 12c Data Object internal name. All leading underscores are removed as part of conversion.
2. The 11g Data Object paths and names are combined to form the 12c Data Object display name.
3. The 11g column IDs are converted to 12c column internal names. All leading underscores are removed as part of conversion.
4. If an “auto-incr-integer” column is present in the 11g Data Object XML file, its values populate the BEAM_ID column in the 12c Data Object.
5. If a “Timestamp” column is present in the 11g Data Object XML file, its values populate the DATAOBJECT_CREATED column in the 12c Data Object.
6. An 11g String column is converted to a 12c Varchar column with a width of no more than 2000 characters.

BAM11gSimpleDOSample.xml [11g Simple DO sample XML file]

```
<?xml version="1.0"?>
<OracleBAMExport Version="2025">
  <DataObject      Version="14"      Name="AllDataTypesDO"      ID="_AllDataTypesDO"
Path="/bam/samples" External="0">
  <Layout>
    <TipText>Data Object with all data types</TipText>
    <Column Name="StringField" ID="_StringField" Type="string" MaxSize="100" Nullable="1">
```

```

Public="1"/>
  <Column Name="LongStringField" ID="_LongStringField" Type="string" MaxSize="10000"
Nullable="1" Public="1"/>
  <Column Name="IntegerField" ID="_IntegerField" Type="integer" Nullable="0" Public="1"/>
  <Column Name="FloatField" ID="_FloatField" Type="string" MaxSize="100" Nullable="0"
Public="1"/>
  <Column Name="DecimalField" ID="_DecimalField" Type="decimal" Scale="10" Nullable="1"
Public="1"/>
  <Column Name="DateTimeField" ID="_DateTimeField" Type="datetime" Nullable="1"
Public="1"/>
  <Column Name="BooleanField2" ID="_BooleanField" Type="boolean" Nullable="1"
Public="1"/>
  <Column Name="AutoIncrementingIntegerField" ID="_AutoIncrementingIntegerField"
Type="auto-incr-integer" Nullable="0" Public="1"/>
  <Column Name="TimestampField" ID="_TimestampField" Type="timestamp" Nullable="0"
Public="1"/>
  <Indexes>
  </Indexes>
  <AccessControlList/>
</Layout>
<Contents>
</Contents>
</DataObject>
</OracleBAMExport>

```

After importing the 11g DO file "**BAM11gSimpleDOSample.xml**" (./bamcommand -cmd import -file BAM11gSimpleDOSample.xml -upgrade 1), you can login to the Administrator tab to view the "AllDataTypesDO" Data Object in 12c.

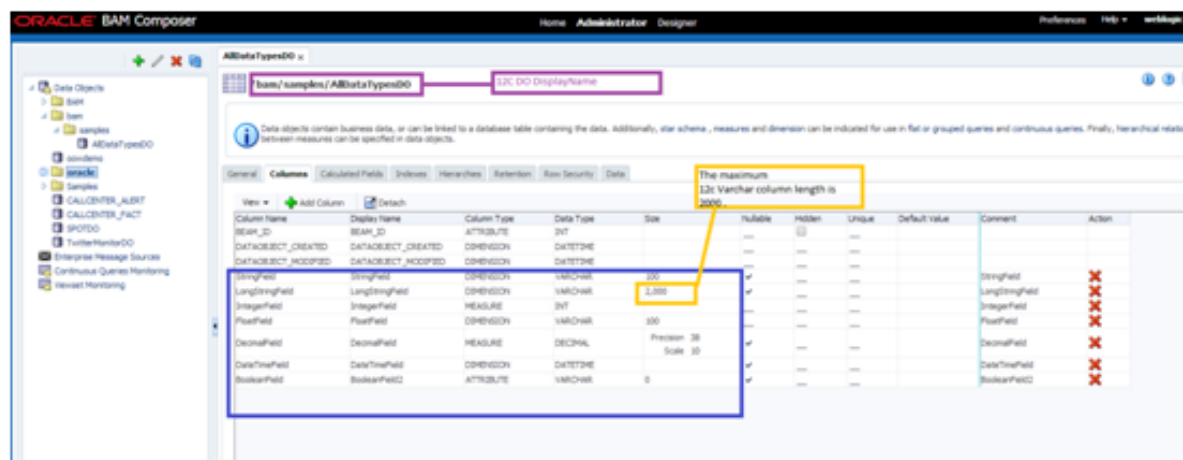


Figure 1 – Accessing the Administrator Tab

The creation of “AllDataTypesDO” Dataobject in 12c abides by all the six rules mentioned previously.

The 12c Data Object’s Name (AllDataTypesDO) is changed by removing leading underscores from 11g Data Object ID (_AllDataTypesDO), as per rule 1.

The 12c Data Object’s Display Name (/bam/samples/AllDataTypesDO) is changed by combining the 11g Data Object’s path (/bam/samples/) and the 11g Data Object’s Name (AllDataTypesDO), as per rule 2.

The 12c Data Object’s Column Name (StringField) is changed by removing leading underscores from the 11g Data Object Column ID (_StringField), as per rule 3.

The 11g Data Object XML has “AutoIncrementingIntegerField”, which is of column type “auto-incr-integer”, which is why no new column is created corresponding to this 11g column in 12c. This column gets mapped to the 12c BEAM_ID column, as per rule 4.

The 11g Data Object XML has “TimestampField”, which is of column type “timestamp”. Again, no new column is created corresponding to this 11g column in 12c. This column gets mapped to the 12c DATAOBJECT_CREATED column, as per rule 5.

The length of the 11g Data Object XML column “LongStringField” is specified as 10000, but in 12c it is truncated to 2000 because that’s the most that Varchar can support, as per rule 6.

2.2 11g FACT Data Object Upgrade

The rules defined for 11g Simple Data Object hold good for the 11g Fact DO upgrade as well. The additional rules which apply for 11g Fact Data Objects upgrade to 12c are as follows.

1. An 11g fact Data Object is converted into two 12c Data Objects:
 - a. A 12c Simple Data Object consisting of the 11g non-lookup columns
 - b. A 12c Logical Data Object consisting of the 11g lookup columns
2. The lookup columns must be present in the 11g fact Data Object XML file or in the 12c schema.

BAM11gFACTDOSample.xml [11g FACT DO sample XML file]

```
<?xml version="1.0"?>
<OracleBAMExport Version="2025">
<DataObject Version="14" Name="BAM11gFact" ID="_BAM11gFact" Path="/Samples"
External="0">
  <Layout>
    <Column Name="CompanyName" ID="_CompanyName" Type="string" MaxSize="100"
Nullable="1" Public="1"/>
    <Column Name="CompanyHq" ID="_CompanyHq" Type="string" MaxSize="100" Nullable="1"
Public="1"/>
    <Column Name="EmployeeId" ID="_EmployeeId" Type="integer" Nullable="1" Public="1"/>
    <Column Name="EmpName" ID="_EmpId" Type="lookup">
      <Lookup>
        <DataObject>
          <Name>BAM11gEmp</Name>
          <Path>/Samples</Path>
          <ID>_BAM11gEmp</ID>
        </DataObject>
        <LookupFieldID>_EmpId</LookupFieldID>
        <LookupFieldName>EmpId</LookupFieldName>
        <MatchFields>
          <KeyPair>
            <PrimaryKeyID>_EmpId</PrimaryKeyID>
            <PrimaryKeyName>EmpId</PrimaryKeyName>
            <ForeignKeyID>_EmployeeId</ForeignKeyID>
            <ForeignKeyName>EmployeeId</ForeignKeyName>
          </KeyPair>
        </MatchFields>
      </Lookup>
    </Column>
    <Column Name="EmpLocation" ID="_Location" Type="lookup">
      <Lookup>
```

```

<DataObject>
  <Name>BAM11gLocation</Name>
  <Path>/Samples</Path>
  <ID>_BAM11gLocation</ID>
</DataObject>
<LookupFieldID>_Location</LookupFieldID>
<LookupFieldName>Location</LookupFieldName>
<MatchFields>
  <KeyPair>
    <PrimaryKeyID>_Empld</PrimaryKeyID>
    <PrimaryKeyName>Empld</PrimaryKeyName>
    <ForeignKeyID>_EmployeeId</ForeignKeyID>
    <ForeignKeyName>EmployeeId</ForeignKeyName>
  </KeyPair>
</MatchFields>
</Lookup>
</Column>
<Indexes/>
</Layout>
<Contents>
</Contents>
</DataObject>
<DataObject Version="14" Name="BAM11gEmp" ID="_BAM11gEmp" Path="/Samples"
External="0">
  <Layout>
    <Column Name="Name" ID="_Name" Type="string" MaxSize="100" Nullable="1" Public="1"/>
    <Column Name="Age" ID="_Age" Type="string" MaxSize="100" Nullable="1" Public="1"/>
    <Column Name="Empld" ID="_Empld" Type="auto-incr-integer" Nullable="0" Public="1"/>
  <Indexes/>
</Layout>
<Contents>
</Contents>

```



```

</DataObject>
  <DataObject Version="14" Name="BAM11gLocation" ID="_BAM11gLocation" Path="/Samples"
External="0">
  <Layout>
    <Column Name="Empld" ID="_Empld" Type="auto-incr-integer" Nullable="0" Public="1"/>
    <Column Name="Location" ID="_Location" Type="string" MaxSize="100" Nullable="1"
Public="1"/>
    <Column Name="Pin" ID="_Pin" Type="integer" Nullable="1" Public="1"/>
  </Layout>
  <Indexes/>
</DataObject>
</OracleBAMExport>

```

After importing the 11g Fact DO XML “BAM11gFACTDOSample.xml” (./bamcommand –cmd import –file BAM11gFACTDOSample.xml –upgrade 1 –mode append), you must login to the Administrator tab to view the “BAM11gFact” and “BAM11gFact_BASE” Data Objects in 12c.

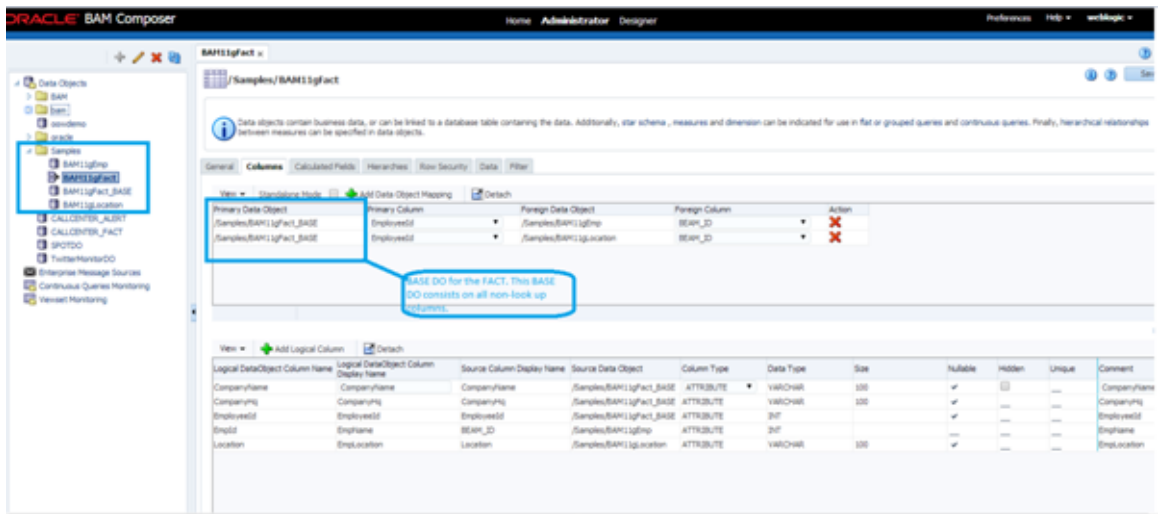


Figure 2 – Accessing the Administrator Tab to View Data Objects in 12c

The 12c Logical DO doesn’t support non–lookup columns. Thus, while upgrading the 11g Fact DO “BAM11gFact”, two Data Objects are created in 12c.

1. The first is the BASE Data Object “BAM11gFact_BASE”, which consists of all 11g FACT DO non-lookup columns.
2. The second is the 12c Logical DO “BAM11gFact”, which consists of 11g FACT DO lookup columns.

11g External Data Object Upgrade

The rules defined for the 11g Simple Data Object hold good for the 11g External DO upgrade. The additional rules which apply for 11g External Data Objects upgrade to 12c are as follows.

1. The BAM 12c BAMCommand utility does not upgrade the external data source referenced by an 11g external Data Object. You must recreate the external data source before upgrading the 11g external Data Object, which is mapped to a 12c external Data Object.
2. An 11g external Data Object referenced by a standard BAM 11g FACT Data Object is mapped to a 12c simple Data Object, not to a 12c external Data Object. You must use a separate operation to migrate the data.

BAM11gExternalDOSample.xml [11g External DO sample XML file]

```
<?xml version="1.0" encoding="utf-8"?>
<OracleBAMExport Version="1003.0" Build="3.5.5750.0">
<DataObject Version="14" Name="EXTDO" ID="_EXTDO" Path="/bam/samples" External="1"
ExternalName="EXTERNAL_TABLE" ExternalDataSource="BamDataSource">
  <Layout>
    <Description><![CDATA[External Data Source fuer METRO POS PoC]]></Description>
    <Column Name="ID" ID="_ID" External="1" ExternalName="EXTERNAL_TABLE_ID"
Type="integer" Nullable="1" Public="1" />
    <Column Name="NAME" ID="_NAME" External="1"
ExternalName="EXTERNAL_TABLE_NAME" Type="string" MaxSize="120" Nullable="1"
Public="1" />
    <Indexes />
  </Layout>
<Contents>
</Contents>
</DataObject>
</OracleBAMExport>
```

This is a simple 11g External DO. After importing this 11g DO XML, BAM11gExternalDOSample.xml" (./bamcommand -cmd import -file BAM11gExternalDOSample.xml -upgrade 1), you can login to the Administrator tab to view the external Data Object in 12c.

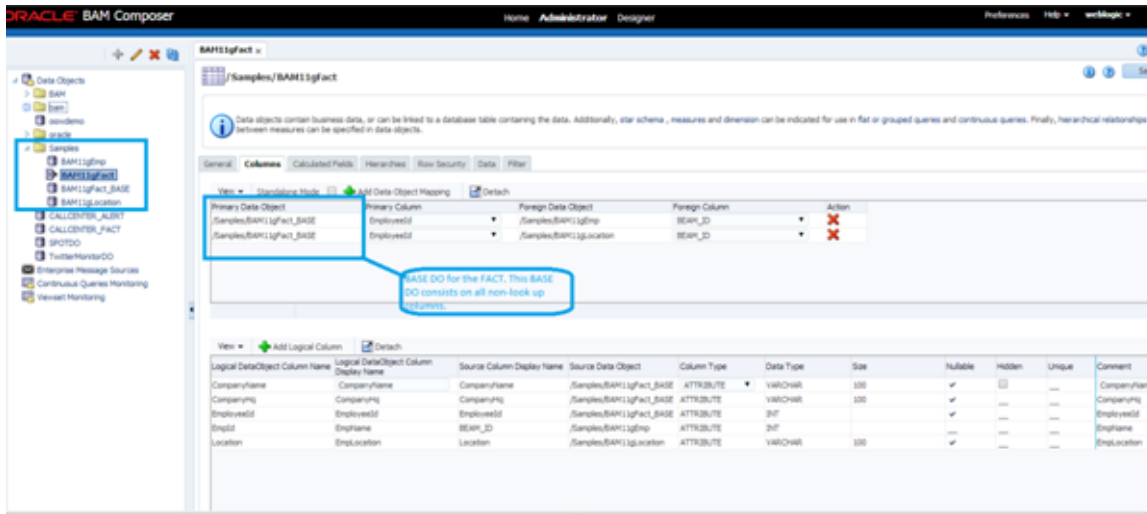


Figure 3 – Viewing the External DO in BAM Composer

You will now find that a new External DO “EXTDO2” has been created in 12c. Let’s take another example of an 11g External DO.

BAM11gExternalDOSample2.xml [11g External DO sample XML file]

```

<?xml version="1.0"?>
<OracleBAMExport Version="2025">
  <DataObject Version="14" Name="FactDOwithExternalIDORef" ID="_FactDOwithExternalIDORef"
  Path="/bam/samples" External="0">
    <Layout>
      <Column Name="ID" ID="_ID" Type="auto-incr-integer" Nullable="0" Public="1"/>
      <Column Name="EmpName" ID="_EmpName" Type="lookup">
        <Lookup>
          <DataObject>
            <Name>EXTDO2</Name>
            <Path>/bam/samples</Path>
            <ID>_EXTDO2</ID>
          </DataObject>
          <LookupFieldID>_NAME</LookupFieldID>
          <LookupFieldName>NAME</LookupFieldName>
          <MatchFields>
            <KeyPair>
              <PrimaryKeyID>_ID</PrimaryKeyID>
            </KeyPair>
          </MatchFields>
        </Lookup>
      </Column>
    </Layout>
  </DataObject>
</OracleBAMExport>

```

```
<PrimaryKeyName>ID</PrimaryKeyName>
  <ForeignKeyID>_ID</ForeignKeyID>
  <ForeignKeyName>ID</ForeignKeyName>
</KeyPair>
</MatchFields>
</Lookup>
</Column>
<Column Name="LocationName" ID="_LocationName" Type="lookup">
  <Lookup>
    <DataObject>
      <Name>location</Name>
      <Path>/bam/samples</Path>
      <ID>_location</ID>
    </DataObject>
    <LookupFieldID>_LocationName</LookupFieldID>
    <LookupFieldName>LocationName</LookupFieldName>
    <MatchFields>
      <KeyPair>
        <PrimaryKeyID>_ID</PrimaryKeyID>
        <PrimaryKeyName>ID</PrimaryKeyName>
        <ForeignKeyID>_ID</ForeignKeyID>
        <ForeignKeyName>ID</ForeignKeyName>
      </KeyPair>
    </MatchFields>
  </Lookup>
</Column>
<Indexes/>
</Layout>
<Contents>
</Contents>
</DataObject>
```

```

<DataObject Version="14" Name="location" ID="_location" Path="/bam/samples" External="0">
  <Layout>
    <Column Name="ID" ID="_ID" Type="auto-incr-integer" Nullable="0" Public="1"/>
    <Column Name="LocationName" ID="_LocationName" Type="string" MaxSize="100"
    Nullable="1" Public="1"/>
    <Column Name="PinCode" ID="_PinCode" Type="string" MaxSize="100" Nullable="1"
    Public="1"/>
    <Indexes/>
  </Layout>
</DataObject>

<DataObject Version="14" Name="EXTDO2" ID="_EXTDO2" Path="/bam/samples" External="1"
ExternalName="EXTERNAL_TABLE" ExternalDataSource="BamDataSource">
  <Layout>
    <Description><![CDATA[External Data Source fuer METRO POS PoC]]></Description>
    <Column Name="ID" ID="_ID" External="1" ExternalName="EXTERNAL_TABLE_ID"
    Type="integer" Nullable="1" Public="1" />
    <Column Name="NAME" ID="_NAME" External="1" ExternalName="EXTERNAL_TABLE_NAME"
    Type="string" MaxSize="120" Nullable="1" Public="1" />
    <Indexes />
  </Layout>
<Contents>
</Contents>
</DataObject>
</OracleBAMExport>

```

In this case, the 11g External DO "EXTDO2" is being referred by the 11g FACT DO "FactDOwithExternalDORef". After importing the 11g DO XML "BAM11gExternalDOSample2.xml" (./bamcommand -cmd import -file BAM11gExternalDOSample2.xml -upgrade 1 -mode append), the 11g External Data Object is mapped to 12c Simple Data Object.

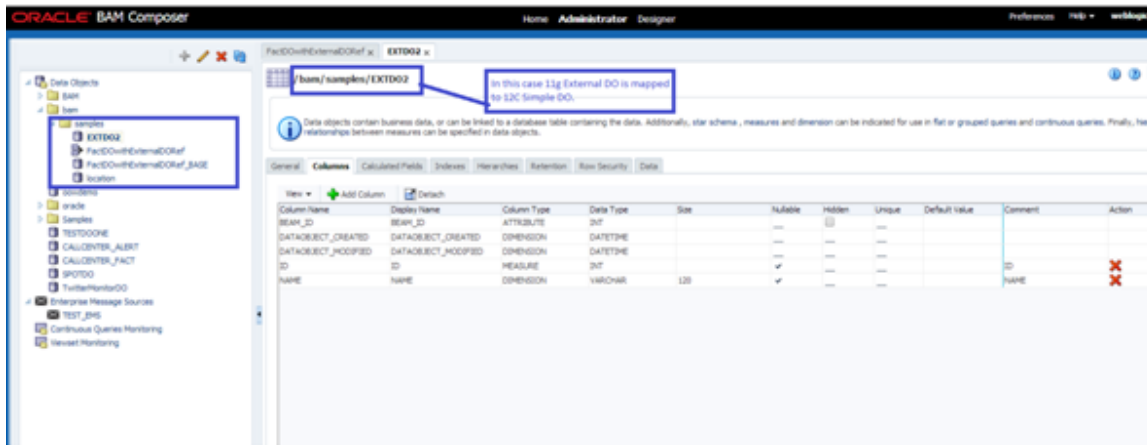


Figure 4 – 11g EDO Mapped to a 12c SDO

This is because the 11g FACT DO gets mapped to 12c Logical DO, and the 12c Logical DO can't refer to an External DO.

11g Security Filter Upgrade

The 11g Security filters can also be migrated to the 12c environment using BAMCommand.

The key point to note while importing the 11g Security filter is that the Security DO, which is being referred by the 11g Security filter, needs to have the data.

If 11g Security DO doesn't have the data, then security filter can't be created in 12c.

The customer is responsible for creating the user roles, which are specified in the 11g security DO, in 12c environment.

BAM11gSecurityFilter.xml [11g Security Filter sample XML file]

```
<?xml version="1.0"?>
<OracleBAMExport Version="2025">
  <SecurityFilters      DataObjectName="Employees"      DataObjectID="_Employees"
DataObjectPath="/bam/samples">
  <SecurityFilter ID="2">
    <Name>TestSecFilter</Name>
    <SecurityDO>secEmployees</SecurityDO>
    <SecurityDOPath>/bam/samples</SecurityDOPath>
    <IdentificationType>USER,OR</IdentificationType>
    <SecurityDOUserColumn>_Username</SecurityDOUserColumn>
    <SecurityDOUserColumnName>Username</SecurityDOUserColumnName>
    <SecurityDOMatchColumn>_ID</SecurityDOMatchColumn>
```

```

<SecurityDOMatchColumnName>ID</SecurityDOMatchColumnName>
<FilteredDOMatchColumn>_Sales_Area</FilteredDOMatchColumn>
<FilteredDOMatchColumnName>Sales Area</FilteredDOMatchColumnName>
</SecurityFilter>
</SecurityFilters>
<DataObject Version="14" Name="secEmployees" ID="_secEmployees" Path="/bam/samples"
External="0">
  <Layout>
    <Column Name="Username" ID="_Username" Type="string" MaxSize="100" Nullable="1"
Public="1"/>
    <Column Name="ID" ID="_ID" Type="string" MaxSize="100" Nullable="1" Public="1"/>
    <Indexes/>
  </Layout>
  <Contents>
    <Row ID="1">
      <Column ID="_Username" Value="weblogic"/>
      <Column ID="_ID" Value="Northeast"/>
    </Row>
  </Contents>
</DataObject>
<DataObject Version="14" Name="Employees" ID="_Employees" Path="/bam/samples"
External="0">
  <Layout>
    <Column Name="Salesperson" ID="_Salesperson" Type="string" MaxSize="100" Nullable="1"
Public="1"/>
    <Column Name="Sales Area" ID="_Sales_Area" Type="string" MaxSize="100" Nullable="1"
Public="1"/>
    <Column Name="Sales Number" ID="_Sales_Number" Type="integer" Nullable="1"
Public="1"/>
    <Column Name="Timestamp" ID="_Timestamp" Type="timestamp" Nullable="0" Public="1"/>
    <Indexes/>
  </Layout>
  <Contents>

```

```

</Contents>
</DataObject>
</OracleBAMExport>

```

After importing this 11g Security Filter XML “**BAM11gSecurityFilter.xml**” (./bamcommand -cmd import -file BAM11gSecurityFilter.xml –upgrade 1 -mode append), you can login to the Administrator tab and click on the row security tab of the Data Object to view security filters.

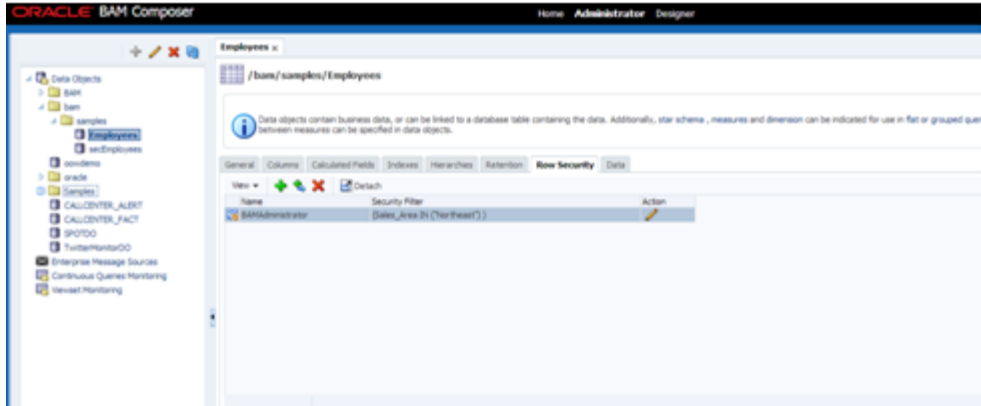


Figure 5 – Viewing Row Security Filters in BAM Composer

BAM 11g EMS Upgrade to 12c

The 11g EMS Connection Factory details are not retained. The Outbound Connection JNDI is mapped to NONE. You must create these details manually and choose the 12c JNDI name.

The 11g EMS field “Start when BAM Server starts” is not mapped as part of upgrade process; so the “Auto-start” field is set to “false” in 12c. You will have to start the EMS manually.

The 11g EMS definition XML file lacks the Data Object column data types. Therefore, you must specify Data Object and EMS definitions in the same XML file if the base Data Object is not present in the 12c database.

BAM11gEMSDO.xml [11g Simple DO sample XML file]

```

<?xml version="1.0"?>
<OracleBAMExport Version="2025">
  <DataObject Version="14" Name="TESTDOONE" ID="_TESTDOONE" External="0">
    <Layout>
      <Column Name="regionone" ID="_region" Type="string" MaxSize="100" Nullable="1" Public="1"/>
      <Column Name="stateone" ID="_state" Type="string" MaxSize="100" Nullable="1" Public="0"/>
      <Column Name="sales one" ID="_sales" Type="integer" Nullable="1" Public="1"/>
    </Layout>
  </DataObject>
</OracleBAMExport>

```



```

<Column Name="price" ID="_price" Type="float" Nullable="1" Public="1"/>
<Column Name="available" ID="_available" Type="boolean" Nullable="1" Public="1"/>
<Column Name="serialId" ID="_serialId" Type="auto-incr-integer" Nullable="0" Public="1"/>
<Indexes/>
</Layout>
<Contents>
</Contents>
</DataObject>
</OracleBAMExport>

```

BAM11gEMS.xml [11g EMS sample XML file. This EMS refers to the DO defined in BAM11gEMSDO.xml]

```

<?xml version="1.0"?>
<OracleBAMExport Version="2025">
  <EMS Version="2" Name="TEST_EMS" ID="1"
  ><EnterpriseMessageSource
  version="8"><Description><EMSIIsDurableSubscribed>true</EMSIIsDurableSubscribed><EMSTyp
  e>EMST.ORACLE</EMSType><Location>
    <Part1>weblogic.jndi.WLInitialContextFactory</Part1>
    <Part2>t3://slc03rzk.us.oracle.com:7001</Part2>
    <Part3>weblogic.jms.ConnectionFactory</Part3>
    <Part4>TestQueueOne</Part4>
    <Part5/>
    <Part6/>
    <Part7>8</Part7>
    <Part8/>
    <Part9/>
    <Part10>/TESTDOONE</Part10>
    <Part11>2</Part11>
    <Part12>1</Part12>
    <Part13>1</Part13>
    <Part14>Yes</Part14>
  </EnterpriseMessageSource>
  </EMS>
</OracleBAMExport>

```

```

<Part15/>
<Part16/>
</Location><Fields><Field
System="0"><ID>DummyName</ID><Key>off</Key><ProvidersName>DummyName</Providers
Name><DisplayName>DummyName</DisplayName><Type>String</Type><MaxSize>0</MaxSiz
e><PackingSpec
Type="XML"><BatchNamespaceURI/><MsgNamespaceURI/><BatchNode/><RootNode>row</Ro
otNode><Path/><DTPattern/><DTLanguage/><DTCountry/><DTVariant/><Fields
Type="Tags"><Field                                     Key="false"
Type="String"><Name>region</Name><DisplayName>regionone</DisplayName></Field><Field
Key="true"
Type="String"><Name>state</Name><DisplayName>stateone</DisplayName></Field><Field
Key="false"                                     Type="String"><Name>sales</Name><DisplayName>sales
one</DisplayName></Field><Field                                     Key="false"
Type="String"><Name>price</Name><DisplayName>price</DisplayName></Field><Field
Key="false"
Type="String"><Name>available</Name><DisplayName>available</DisplayName></Field><Field
Key="false"
Type="String"><Name>serial</Name><DisplayName>serialId</DisplayName></Field></Fields></
PackingSpec></Field></Fields><Fault includeFault="true">

<LOG includeLog="true">
  <IncludePayload includePayloadValue="true"/>
</LOG>
<WRITE writeMessage="false">
  <WriteToDo writeMessageToDo="false">
    <DOName dataObjectName=""/>
    <DOID dataObjectID=""/>
    <DOField dataObjectField=""/>
  </WriteToDo>
  <WriteToJMS writeMessageToJMS="false">
    <JMSInitialContextFactory contextFactoryName=""/>
    <JNDIUrl jndiUrlValue=""/>
    <ConnectionFactory connectionFactoryName=""/>
    <TopicQueueName destinationName=""/>
    <JNDIUsername jndiUsernameValue=""/>
    <JNDIPassword jndiPasswordValue=""/>
    <JMSUsername jmsUsernameValue=""/>

```

```

    <JMSPassword jmsPasswordValue=""/>
  </WriteToJMS>
</WRITE>
</Fault></Description></EnterpriseMessageSource>
</EMS>
</OracleBAMExport>

```

You must import the Base DO “BAM11gEMSDO.xml” (./bamcommand -cmd import -file BAM11gEMSDO.xml -upgrade 1) before importing the 11g EMS.

After importing the 11g EMS XML “BAM11gEMS.xml” (./bamcommand -cmd import -file BAM11gEMS.xml -upgrade 1 -mode append), you can login to the Administrator tab and click EMS “TEST_EMS” to view the new 12c EMS.

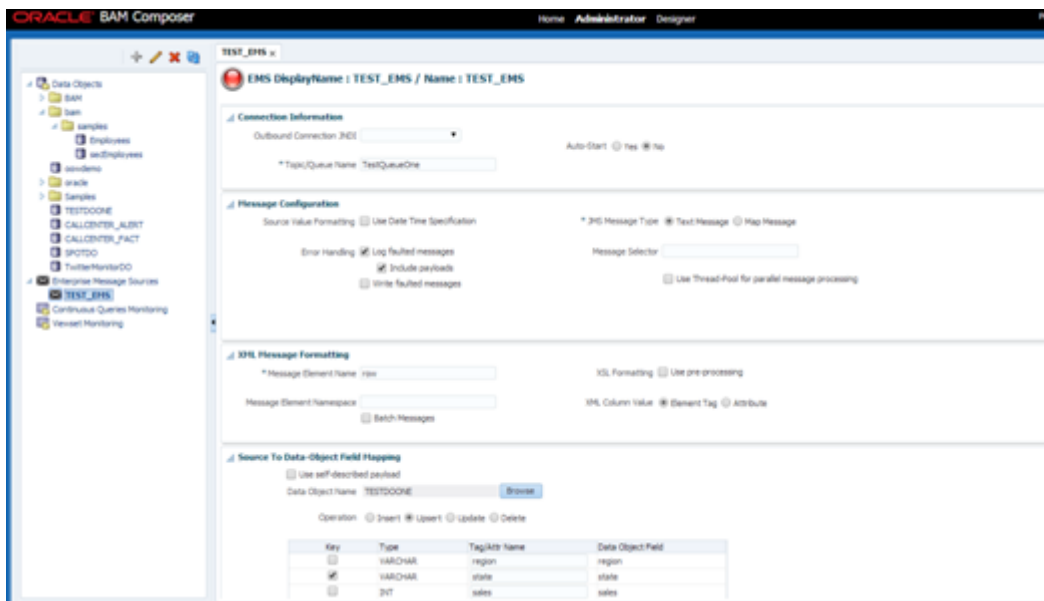


Figure 6 – Viewing the 12c EMS in BAM Composer

You must select the proper Outbound JNDI name, and verify that Queue/Topic name has been created before starting the EMS.

BAM 11g DO Data Migration

The BAMCommand utility is used to migrate the 11g Data Object data to 12c.

The “-migrate” option of BAMCommand is used to migrate the 11g Data Object data to 12c.

The Command: ./bamcommand -cmd import -file <11g DO XML file> -migrate 1

The following considerations apply to 11g Data Object migration:

1. If an “auto-incr-integer” type column is present in the 11g Data Object XML file, its values populate the BEAM_ID column in the 12c Data Object.
2. If a “Timestamp” type column is present in the 11g Data Object XML file, its values populate the DATAOBJECT_CREATED column in the 12c Data Object.

The records are inserted in batches. The default batch size is 10000, which can be overridden (-batchsize 100) using command line.

The server need not be up and running at the time of data migration.

Before migrating 11g data, you must ensure that the DB details such as dbusername and dburl, mentioned in the BAMCommandConfig.xml, are valid.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<BAMCommandConfig>
<host>localhost</host>
<port>7001</port>
<dbusername>DEV12_SOAINFRA</dbusername>
<dburl>jdbc:oracle:thin:@localhost:1521:orcl</dburl>
</BAMCommandConfig>
```

BAM11gSimpleDOWithData.xml [11g Simple DO sample XML file]

```
<?xml version="1.0"?>
<OracleBAMExport Version="2025">
  <DataObject      Version="14"      Name="AllDataTypesDO"      ID="_AllDataTypesDO"
  Path="/bam/samples" External="0">
    <Layout>
      <TipText>Data Object with all data types</TipText>
      <Column Name="StringField" ID="_StringField" Type="string" MaxSize="100" Nullable="1"
  Public="1"/>
        <Column Name="LongStringField" ID="LongStringField" Type="string" MaxSize="10000"
  Nullable="1" Public="1"/>
      <Column Name="IntegerField" ID="_IntegerField" Type="integer" Nullable="0" Public="1"/>
      <Column Name="FloatField" ID="_FloatField" Type="string" MaxSize="100" Nullable="0"
  Public="1"/>
      <Column Name="DecimalField" ID="_DecimalField" Type="decimal" Scale="10" Nullable="1"
  Public="1"/>
    </Layout>
  </DataObject>
</OracleBAMExport>
```

```

    <Column Name="DateTimeField" ID="_DateTimeField" Type="datetime" Nullable="1"
Public="1"/>
    <Column Name="BooleanField2" ID="_BooleanField" Type="boolean" Nullable="1"
Public="1"/>
    <Column Name="AutoIncrementingIntegerField" ID="_AutoIncrementingIntegerField"
Type="auto-incr-integer" Nullable="0" Public="1"/>
    <Column Name="TimestampField" ID="_TimestampField" Type="timestamp" Nullable="0"
Public="1"/>

<Indexes>
</Indexes>
<AccessControlList/>
</Layout>
<Contents>
<Row ID="1">
    <Column ID="_StringField" Value="ABC"/>
        <Column ID="_LongStringField" Value="DEF"/>
    <Column ID="_IntegerField" Value="100"/>
    <Column ID="_FloatField" Value="10.00"/>
    <Column ID="_DecimalField" Value="100"/>
    <Column ID="_DateTimeField" Value="2012-07-09T07:12:05.0000000PDT"/>
    <Column ID="_BooleanField" Value="true"/>
    <Column ID="_AutoIncrementingIntegerField" Value="8"/>
    <Column ID="_TimestampField" Value="2012-07-09T07:12:39.0000820PDT"/>

</Row>
<Row ID="2">
    <Column ID="_StringField" Value="GHI"/>
        <Column ID="_LongStringField" Value="JKL"/>
    <Column ID="_IntegerField" Null="1"/>
    <Column ID="_FloatField" Null="1"/>
    <Column ID="_DecimalField" Null="1"/>
    <Column ID="_DateTimeField" Null="1"/>

```

```
<Column ID="_BooleanField" Value="false"/>
<Column ID="_AutoIncrementingIntegerField" Value="9"/>
<Column ID="_TimestampField" Value="2012-07-09T07:13:42.0000486PDT"/>
```

```
</Row>
```

```
<Row ID="3">
```

```
<Column ID="_StringField" Value="MNO"/>
  <Column ID="_LongStringField" Value="PQR"/>
<Column ID="_IntegerField" Value="100"/>
<Column ID="_FloatField" Value="10.00"/>
<Column ID="_DecimalField" Value="100"/>
<Column ID="_DateTimeField" Value="2012-07-09T07:12:05.0000000PST"/>
<Column ID="_BooleanField" Value="true"/>
<Column ID="_AutoIncrementingIntegerField" Value="10"/>
<Column ID="_TimestampField" Value="2012-07-09T07:12:39.0000820PST"/>
```

```
</Row>
```

```
<Row ID="4">
```

```
<Column ID="_StringField" Value="STU"/>
  <Column ID="_LongStringField" Value="VWXYZ"/>
<Column ID="_IntegerField" Value="100"/>
<Column ID="_FloatField" Value="10.00"/>
<Column ID="_DecimalField" Value="100"/>
<Column ID="_DateTimeField" Value="2012-07-09T07:12:05.0000000IST"/>
<Column ID="_BooleanField" Value="true"/>
<Column ID="_AutoIncrementingIntegerField" Value="11"/>
<Column ID="_TimestampField" Value="2012-07-09T07:12:39.0000820IST"/>
```

```
</Row>
```

```
</Contents>
```

```
</DataObject>
```

```
</OracleBAMExport>
```

You must upgrade the DO “BAM11gSimpleDOWithData.xml” (./bamcommand -cmd import -file BAM11gSimpleDOWithData.xml -upgrade 1) before migrating the 11g DO data.

After migrating 11g DO data using “BAM11gSimpleDOWithData.xml” (./bamcommand -cmd import -file BAM11gSimpleDOWithData.xml –migrate 1), you can login to the BAM server and click the Data tab of the new 12c Data Object.

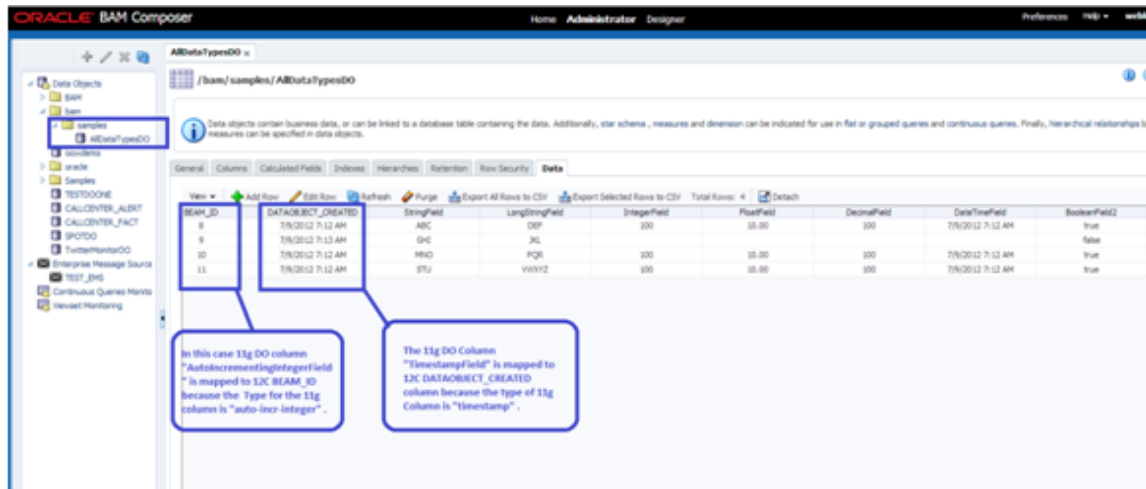


Figure 7 – Accessing the Data Tab of a New 12c Data Object Through BAM Composer

You can perform different data operations such as update and overwrite. You can make changes in the BAM11gSimpleDOWithData.xml file, and update the records using the “-datamode” option.

```
./bamcommand -cmd import -file BAM11gSimpleDOWithData.xml –migrate 1 –datamode update
```

Similarly, you can specify the “-datamode” option as ‘overwrite’ to delete the existing data before importing new records.

```
./bamcommand -cmd import -file BAM11gSimpleDOWithData.xml –migrate 1 –datamode overwrite
```

If 11g DO XML doesn't have any column with type "auto-incr-integer", the BEAM_ID column value is generated in sequential order.

BAM11gSimpleDOWithData2.xml [11g Simple DO sample XML file]

```
<?xml version="1.0"?>
<OracleBAMExport Version="2025">
  <DataObject Version="14" Name="Employee" ID="_Employee" Path="/bam/samples"
  External="0">
    <Layout>
      <Column Name="EmpId" ID="_EmpId" Type="decimal" Nullable="0" Public="1"/>
      <Column Name="EmpName" ID="_EmpName" Type="string" MaxSize="100" Nullable="1"
      Public="1"/>
    </Layout>
  </DataObject>
</OracleBAMExport>
```

```

<Column Name="EmpAge" ID="_EmpAge" Type="string" MaxSize="100" Nullable="1"
Public="1"/>
<Indexes/>
</Layout>
<Contents>
<Row ID="1">
<Column ID="_EmpId" Value="1"/>
<Column ID="_EmpName" Value="A1"/>
<Column ID="_EmpAge" Value="10"/>
</Row>
<Row ID="2">
<Column ID="_EmpId" Value="2"/>
<Column ID="_EmpName" Value="B2"/>
<Column ID="_EmpAge" Value="11"/>
</Row>
</Contents>
</DataObject>
</OracleBAMExport>

```

After importing the 11g DO definition BAM11gSimpleDOWithData2.xml (./bamcommand -cmd import -file BAM11gSimpleDOWithData2.xml -upgrade 1) and migrating the 11g DO data (./bamcommand -cmd import -file BAM11gSimpleDOWithData2.xml -migrate 1), you will see the following under the new 12c DO data tab.

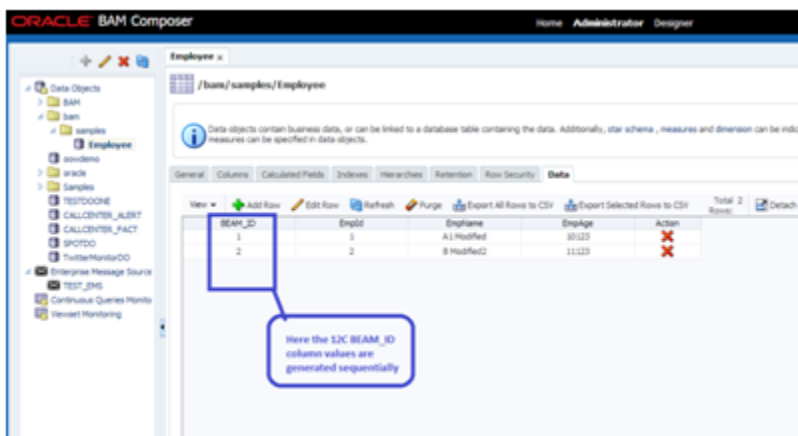


Figure 8 – Accessing the Overwritten Data





CONNECT WITH US.
ORACLE
blogs.oracle.com/oracle
B
facebook.com/oracle
f
twitter.com/oracle
t
oracle.com
O

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

Hardware and Software, Engineered to Work Together

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0914

 | Oracle is committed to developing practices and products that help protect the environment