

TUTORIAL

BDNA Data Platform 5.0 Console Mashup Configuration

September 2016

LEGAL NOTICES

Use of the BDNA software and products are subject to the terms and conditions of a license agreement found in either a separately executed master license agreement or the click-through master license agreement which is accepted prior to delivery of the BDNA software and/or products.

Copyright © 2001-2016. BDNA Corporation

Information in this manual and all BDNA technical support policies are subject to change without notice. Check with your BDNA authorized representative to ensure that you have the most recent information.

BDNA®, the BDNA logo, Technopedia®, BDNA Discover™, BDNA Normalize®, BDNA Normalize for ConfigMgr, BDNA Analyze™, BDNA Normalize for Purchase Orders, IT Genome Company, Know What You're Made Of are trademarks or registered trademarks of BDNA Corporation in the United States and other countries.

Oracle is a trademark or registered trademark of Oracle Corporation in the United States and in other countries. Red Hat Enterprise Linux (RHEL) is a trademark or registered trademark of Red Hat Incorporated in the United States and in other countries. Microsoft Internet Explorer is a trademark or registered trademark of Microsoft Corporation in the United States or in other countries. All other trademarks appearing herein are the property of their respective owners.

The products described herein may be technically combined with third party or other products not supplied by BDNA, including third party or customer software, hardware and materials. Any combinations or potential combinations described herein are advisory only. BDNA expressly disclaims any liability, and any express or implied representation and warranty, resulting from any combinations of the BDNA products with any products not supplied by BDNA.

This document is provided “as is” and without warranty of any kind, and BDNA and its licensors (hereinafter collectively referred to as “BDNA”) expressly disclaim all warranties, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose and against infringement.

Confidential and Proprietary to BDNA.

BDNA Corporation
339 North Bernardo Avenue, Suite 206
Mountain View, CA 94043
USA
Phone +1 650 625 9530
Fax +1 650 625 9533
<http://www.bdna.com>

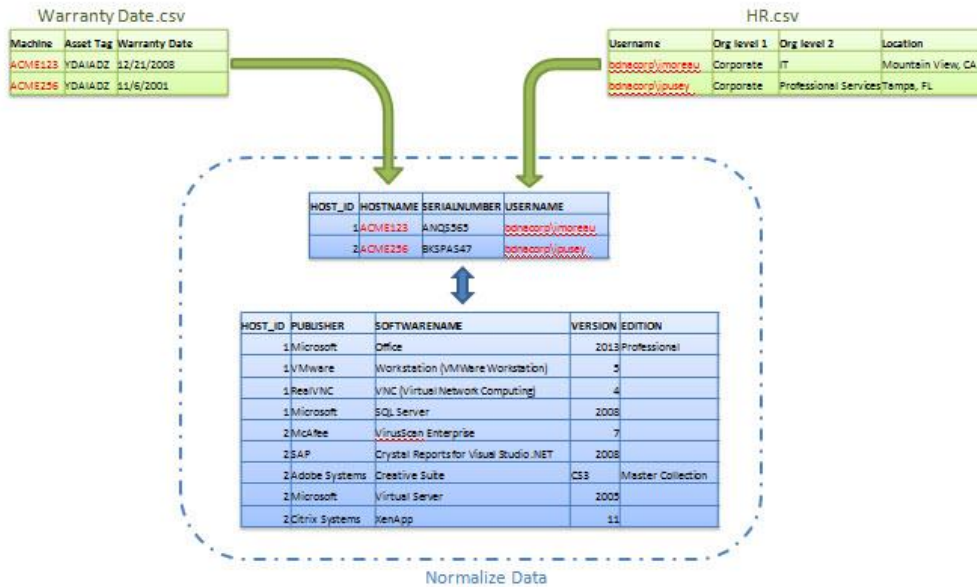
Mashup Steps – Table of Contents

- What is a Mashup?4
- Add a Data Mashup Process5
 - Add Mashup from Database Connection5
 - Create a Mashup CSV File7
 - Add Mashup of CSV File8
 - Add Mashup of Normalized Zip File.....8
- Objects and Mapping Fields9
- Mashup Configuration File12
 - Config File Sections12
 - Sample Mashup Config File14

What is a Mashup?

Mashup is a process that brings in data from another data source to add information that is helpful to know and work with along with your normalized data. An example is to bring in a Warranty date for a host machine or Department and Location of a user.

This data can be matched with specific Normalize object types of Machine, User, Software, Hardware, OS, and CPU. It can come directly from a database of MSSQL, Oracle, DB2, MySQL, PostgreSQL and Sybase. This requires a custom configuration file to pull out the required data. Or it can come from a flat csv file that is linked to a specific object, such as matching Department and Location fields to a user or Warranty Date fields to a machine. Here is an example of how these files are matched with normalized data.



When the mashup process is processed, it becomes part of the BDNA_PUBLISH database and the data is available for reporting in BDNA Analyze. You can add these additional fields as rows to get more detail about your software inventory.

Analyzer Report

Available fields (127) for: Hardware and Software

Find: View

Layout

Rows

- SW Name
- SW Major Version
- DEPT1
- DEPT2

Drop Level Here

Columns

- LOC

Drop Level Here

Measures

- SW Install Count

Drop Measure Here

Unsaved Report

3 Filters in use

- SW Name includes Office
- SW Manufacturer includes Microsoft
- Is Licensable? includes Yes

SW Name	SW Major Version	DEPT1	DEPT2	SW Install Count
Office	2000	Corporate	Sales	2
			Services	2
	2003	Corporate	Sales	3
			Services	1
	2007	External	N/A	1
			Marketing	20
		Corporate	Dev	17
			Services	6
		External	N/A	16
			Federal	3
	97	Corporate	Marketing	1
	XP	Corporate	Sales	1
Services			1	

Add a Data Mashup Process

- Prerequisites**
- License key must grant access to the Data Mashup feature
 - Create at least one IT Discovery Tool, Fingerprint, and/or Normalize CM process
 - Determine your data source
 - Create a mashup configuration file with instructions for retrieving data from the specified data source

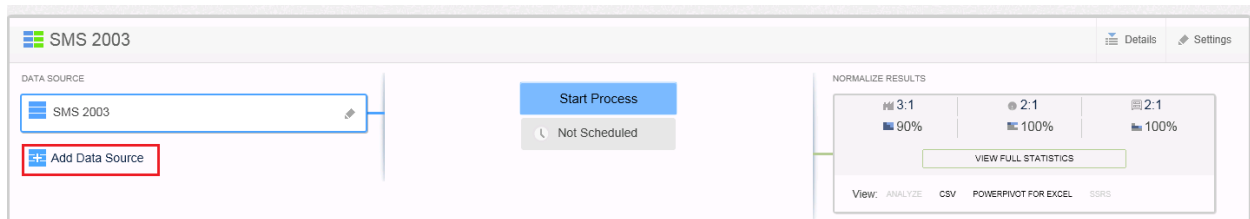
A data mashup allows the import of an external data source to the Normalize database. For example, you can input warranty data from an Accounting database or Human Resource data from a Human Resources database. A data mashup is added as a separate data source to an existing process.

There are three types of data sources used as a mashup source.

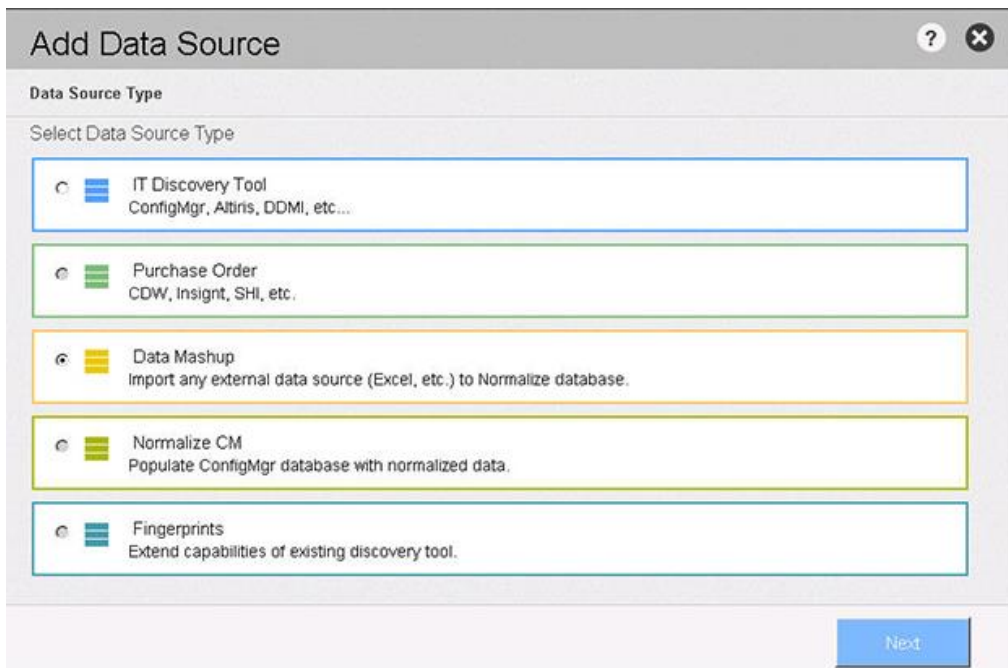
- **Database Connection obtains data from** your MSSQL, Oracle, DB2, MySQL, PostgreSQL, or Sybase database
- **File (csv)** obtains data from a local mashup file or a CSV file located on a network or server local path
- **Zip File obtains data from a Normalize Zip file located on a network or local server**

Add Mashup from Database Connection

1. Click **Add Data Source**, located on the main Data Platform Console.



2. The Data Source type dialog opens. Select the **Data Mashup** radio button.



1. Click **Next**. The Mashup Source dialog opens.
2. Select the **Other Database Connection** radio button and click **Next**.



3. The Select Data Source dialog opens. Select a radio button next to the **data source** that applies to your environment.
4. Click the **Upload Config** button and select your customized configuration file.
5. Enter the **authentication information** required to connect to the data source (this will vary based on the selected data source.)
6. Click the **Test Connection** button to confirm that BDNA Data Platform can connect to the database.
7. Click **Finish** to add the data source to the process.

Create a Mashup CSV File

To add a data mashup, data that is imported from an external data source must be converted to a CSV file format before it can be loaded into BDNA Normalize.

CSV Header

The CSV column header defines which columns are used for mapping and which are used only as additional information. Set the mapping field definition in the CSV Header.

Matching Key Column

Each column used as a mapping field must have a header written in the following format:

```
KEY-XXXXXXX:YYYYYY
```

Where XXXXXXX is the mapping field name and YYYYYY is: the original label of the column

Other Columns

All other header columns must be labeled as standard CSV columns. If BDNA Normalize is used, the labels are used for BDNA Normalize attribute names.

CSV File Examples

The following tables provide examples of Objects and Mapping Fields in a csv file.

Object: User

For a Mashup CSV file, you must use the Data Platform UI to define the main object that the data attaches to. The fields shown below are available as Attributes in the “User” dimension.

KEY-Domain\Username:FullUserName	DEPT1	DEPT2	SITE1	LOC
ACME\jsmith	Corporate	Marketing	US	NY
ACME\adunn	Corporate	Marketing	ASIA	SG

ACME\lthor	Corporate	Marketing	US	WA
ACME\sadams	Corporate	Marketing	US	TX

Mapping Fields

KEY-Domain\Username:FullUserName

Where Domain\Username is the matching key name

Where FullUserName is the label of this key

Add Mashup of CSV File

1. Click **Add Data Source**, located on the main Data Platform Console.
2. Select the **Data Mashup** radio button and click **Next**. The Select a Mashup Source dialog opens.
3. Select the **Other File (csv, txt)** radio button and click **Next**. The Select Data Source dialog opens.
4. Select an **object to link the data file to** from the drop-down list. The objects are:
 - Machine, User, Software, Hardware, OS, CPU, Manufacturer, Other
5. Click the **Upload local data source** checkbox.

Note: For a file located on a network or Data Platform server, leave the box unchecked and enter the **path and authentication** for the file.

6. Click **Upload CSV** to locate and upload the data mashup file.
7. Click **Finish**.

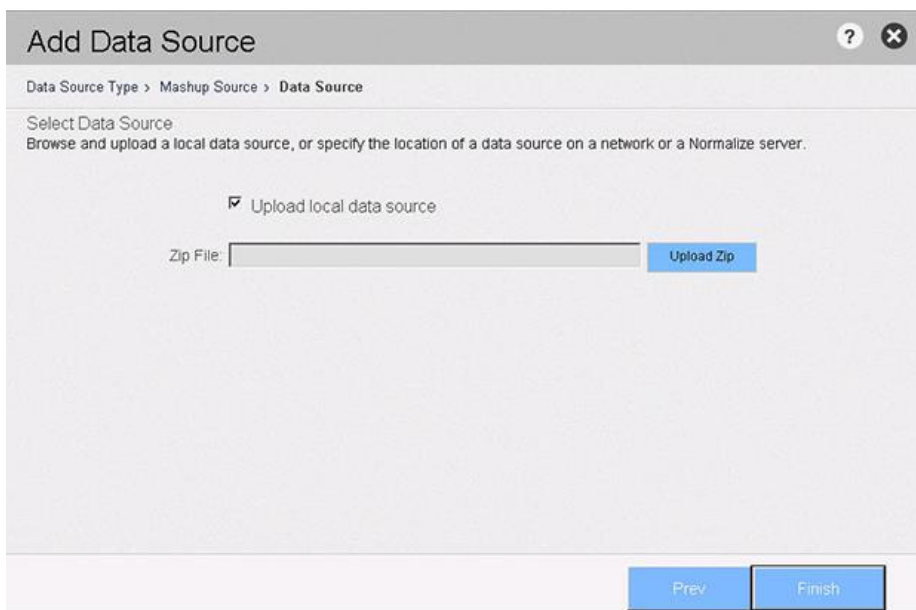
The screenshot shows a 'Create Process' dialog box with the following elements:

- Header: Create Process (with help and close icons)
- Breadcrumbs: Process Type > Mashup Source > Data Source
- Title: Select Data Source
- Instruction: Browse and upload a local data source, or specify the location of a data source on a network or a Normalize server.
- Field: Select an object to link the data to: Machine (dropdown menu)
- Checkbox: Upload local data source
- Field: File: [empty text box] [Upload Csv button]
- Footer: [Prev button] [Finish button]

Add Mashup of Normalized Zip File

1. Click **Add Data Source**, located on the main Data Platform Console.
2. Select the **Data Mashup** radio button and click **Next**. The Select a Mashup Source dialog opens.
3. Select the **Other Normalize Zip File** radio button and click **Next**.

4. The Select Data Source dialog opens. Click the **Upload local data source** checkbox.
5. Click **Upload Zip** to locate and upload the data mashup file.
6. Click **Finish**.



Use a **Mashup** configuration file to “mashup” data from more than one data source.

Objects and Mapping Fields

Here are the reference tables for all available Objects and Mapping Fields that can be used for Data Mashup, Pass-Through, and Purchase Order. They provide the following, plus notes to further explain some items.

TABLE_TYPE—How the data is brought into the system, based on process type.

OBJECT_NAME—Object that you want to attach the data.

TABLE_NAME—Short name for the physical table.

KEY_ITEM— List of available mapping fields for a specific object.

LIMIT_LIST— To indicate a limitation. For example, some fields cannot be standalone and must be associated with other fields.

Object: Machine

TABLE_TYPE	Mashup, Pass-through	
OBJECT_NAME	Machine	
TABLE_NAME	MU_HOST	
KEY_ITEM	KEY	NOTES
	Hostname	

	Domain	Domain cannot be used as a single mapping key. It must be associated with Hostname
	Domain\\Hostname	Double backslash ("\\") is a required escape sequence in an XML file. The escape sequence is not necessary in a CSV file.
	IPAddress	
	MachineID	
	SerialNumber	
LIMIT_LIST	LIST	NOTES
	Hostname	
	Domain\\Hostname	
	IPAddress	
	MachineID	
	SerialNumber	
	Domain,Hostname	

Object: User

TABLE_TYPE	Mashup, Pass-through	
OBJECT_NAME	User	
TABLE_NAME	MU_USR	
KEY_ITEM	KEY	NOTES
	Domain	Domain cannot be used as a single mapping key. It must be associated with Username
	Domain\\Username	Double backslash ("\\") is a required escape sequence in an XML file. The escape sequence is not necessary in a CSV file.
	Username	
LIMIT_LIST	LIST	
	Domain\\Username	
	Username	
	Domain,Username	

Object: Software

TABLE_TYPE	Mashup, Pass-through
OBJECT_NAME	Software
TABLE_NAME	MU_SW
KEY_ITEM	KEY
	ReleaseID
	ProductID
	VersionID
	VersionGroupID
	EditionID
LIMIT_LIST	LIST

	ReleaseID
	ProductID
	VersionID
	VersionGroupID
	EditionID

Object: Hardware

TABLE_TYPE	Mashup, Pass-through
OBJECT_NAME	Hardware
TABLE_NAME	MU_HW
KEY_ITEM	KEY
	ProductID
	ModelID
LIMIT_LIST	LIST
	ProductID
	ModelID

Object: OS

TABLE_TYPE	Mashup, Pass-through
OBJECT_NAME	OS
TABLE_NAME	MU_OS
KEY_ITEM	ReleaseID
	ProductID
	VersionID
	VersionGroupID
	EditionID
LIMIT_LIST	LIST
	ReleaseID
	ProductID
	VersionID
	VersionGroupID
	EditionID

Object: CPU

TYPE	Mashup, Pass-through
OBJECT_NAME	CPU
TABLE_NAME	MU_CPU
KEY_ITEM	KEY
	ModelID
LIMIT_LIST	LIST

	ModelID
--	---------

Mashup Configuration File

Config File Sections

Here is a detailed explanation of the values required in the customizable sections of a configuration file. The sections shown are Header, Fields, and SQL Query.

The customizable portions of the files are color-coded, as follows:

- Black = Standard XML file entries
- Red = Header
- Green = Fields
- Blue = SQL Query

Header

A header defines the type of objects/tables to be imported into Normalize for a Mashup.

```
<Table Type="Mashup" Dynamic_subtype="YYYYY" Name="ZZZZZ" Label="WWWWW">
...
</Table>
```

1. Edit YYYYY and ZZZZZ values based on the OBJECT_NAME per the table.

Dynamic subtype YYYYY OBJECT_NAME	ZZZZZ TABLE_NAME
Machine	MU_HOST
User	MU_USR
Software	MU_SW
Hardware	MU_HW
OS	MU_OS
CPU	MU_CPU
Manufacturer	MU_MFR
Other	MU_OTHER (Or any name limited to 14 characters.)

2. Enter WWWWW as Free Text (30 character maximum) for the table LABEL:

Note: If Analyze is activated, the Label is used as a Dimension name for all objects except “Other” and “Manufacturer”, If TABLE_NAME is “Other”, there is no dynamic view in the Publish database and Analyze database.

Fields

This section describes each field to be imported into Normalize, and whether it is used as a mapping field.

3. Edit the XXXX, YYYYY, Z, W, and VVVV values using the rules below.

```

<Fields>
  <Field Name="XXXXX" DataType="YYYYY" Nullable="Z" DefaultValue=""
  Comment="" Key_Position="W" FileColumnName="VVVVV" />
</Fields>

```

4. Enter Field Name XXXXX. This is the name of the **KEY_ITEM.KEY** in the Objects and Mapping Tables. These values are based on the Dynamic_subtype from the Header.

•If Key_Position W is “1”

If Dynamic_subtype is “Machine”, XXXXX values are:

- Hostname
- Domain
- Domain\Hostname
- IPAddress
- MachineID
- SerialNumber

–Limitation is “Domain,Hostname”

Note: A limitation is the first field which cannot be used without the second. For example, Domain cannot be used without Hostname. You can only define one mapping field at a time, except for the one specified in the Limitation section.

–If Dynamic_subtype is “User”, XXXXX values are:

- Domain
- Domain\Username
- Username

–Limitation is “Domain,Username”

–If Dynamic_subtype is “Software”, XXXXX values are:

- ReleaseID
- ProductID
- VersionID
- VersionGroupID
- EditionID

–If Dynamic_subtype is “Hardware”, XXXXX values are:

- ProductID
- ModelID

–If Dynamic_subtype is “OS”, XXXXX values are:

- ReleaseID
- ProductID
- VersionID
- VersionGroupID
- EditionID

-If Dynamic_subtype is "CPU", XXXXX values are:

- ModelID

-If Dynamic_subtype is "Manufacturer", XXXXX values are:

- ManufacturerID

-If Dynamic_subtype is "Other", it is **NOT AVAILABLE**

5. Where Datatype YYYYYY is the type of the physical column in Publish DB, values are:

- NUMERIC
- NVARCHAR(255)
- DATETIME

Note: For OBJECT_NAME (Machine, User, Software, Hardware, OS, CPU, Manufacturer), the same data type must be used (NUMERIC, NVARCHAR, DATETIME) for all columns—excluding the matching key.

6. Where Nullable=Z defines whether the field can be NULL or NOT, values are:

- 0 if the field can be NULL
- 1 if the field CANNOT be NULL

7. Where Key_Position=W defines whether the field is used as a matching key or not, values are:

- 0 if the field is not used for Matching
- 1 if the field is used for KEY_ITEMS.KEY

8. Where FileColumnName=VVVVV, the value is the name of the column returned by the SQL Query:

SQL Query

This is the SQL Query to be run against the data source that extracts the data and imports it into Normalize.

```
<SQL>  
  <![CDATA [XXXXX] >  
</SQL>
```

9. Enter the XXXXX name of the SQL Query run against the data source.

For a Mashup, the query will return the same number of columns as the number of fields defined in the above Fields section. Each column should return the same name as defined in FileColumnName.

Sample Mashup Config File

Here is a sample completed Config File with the Header, Fields, and SQL Query sections edited for a Mashup.

```
<configuration LoaderConfig="mashup_loader_config_20150124" Disc_Source="Mashup">  
  <Connection Type="MsSqlServer" Impersonate="False" User="" Password="" />  
  <Tables>  
    <Table Type="Mashup" Name="MU_HOST" Dynamic_subtype="Machine" Label="MUNI_LAST">  
      <Fields>  
        <Field Name="MachineID" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""  
Comment="" Key_Position="1" FileColumnName="Key-MachineID:MachineID" />
```

```

    <Field Name="AssetID" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="AssetID" />

    <Field Name="PortfolioItemId" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="PortfolioItemId" />

    <Field Name="Environment" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="Environment" />

    <Field Name="Status" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="Status" />

    <Field Name="City" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="City" />

    <Field Name="State" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="State" />

    <Field Name="Country" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="Country" />

    <Field Name="LastScanDate" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="LastScanDate" />

    <Field Name="PrimaryIPAddress" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="PrimaryIPAddress" />

    <Field Name="AppName" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="AppName" />

    <Field Name="RoadmapIndicator" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="RoadmapIndicator" />

    <Field Name="LASTAppId" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="VASTAppId" />

    <Field Name="AppStatus" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="AppStatus" />

    <Field Name="AppVPName" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="AppVPName" />

    <Field Name="AppOwnerName" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="AppOwnerName" />

    <Field Name="AppCustodianName" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="AppCustodianName" />

    <Field Name="AppITNonIT" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="AppITNonIT" />

    <Field Name="SigAppName" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="SigAppName" />

    <Field Name="SigAppRoadmapIndicator" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="SigAppRoadmapIndicator" />

    <Field Name="SigVASTID" DataType="NVARCHAR(1024)" Nullable="1" DefaultValue=""
    Comment="" Key_Position="0" FileColumnName="SigVASTID" />

    <Field Name="SigAppStatus" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="SigAppStatus" />

    <Field Name="SigAppVPName" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="SigAppVPName" />

    <Field Name="SigAppOwnerName" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="SigAppOwnerName" />

```

```

    <Field Name="SigAppCustodianName" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="SigAppCustodianName" />

    <Field Name="SigAppITNonIT" DataType="NVARCHAR(1024)" Nullable="1"
    DefaultValue="" Comment="" Key_Position="0" FileColumnName="SigAppITNonIT" />

</Fields>

    <SQL><![CDATA[select MachineID as 'Key-
MachineID:MachineID',AssetID,PortfolioItemId,Environment,Status, City,State,Country,

    LastScanDate,PrimaryIPAddress, AppName, RoadmapIndicator, VASTAppId,
AppStatus,AppVPName,

    AppOwnerName,AppCustodianName, AppITNonIT, SigAppName, SigAppRoadmapIndicator,
SigVASTID,

    SigAppStatus, SigAppVPName, SigAppOwnerName, SigAppCustodianName, SigAppITNonIT

from mashup.ItamVastMashup]]></SQL>

</Table>

</Tables>

</configuration>

```