

MashZone NextGen Migration Guide

Version 10.1

October 2017

This document applies to MashZone NextGen Version 10.1 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2013-2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Table of Contents

Preface	5
Migrating data feeds	7
Export content in MashZone legacy.....	9
Import content into MashZone NextGen.....	10
Manual steps required.....	10
Specify permissions.....	11
Migrating JDBC connections	13
Adapting data feeds	15
Converted content	17
Adapt PPM connections	25
Migrate custom dashboard components	27

Preface

The MashZone NextGen Migration Guide includes instructions to migrate data feeds, related contents and configurations from MashZone legacy to Software AG MashZone NextGen.

1 Migrating data feeds

- Export content in MashZone legacy 9
- Import content into MashZone NextGen 10

With MashZone NextGen version 9.12 it is possible to import data feeds, file resource aliases, access permissions, database connections, PPM connections and Real-time buffer from MashZone legacy versions.

Currently not all features of data feeds created in MashZone legacy are supported by MashZone NextGen. Thus you can not import data feeds containing not supported features.

The following MashZone legacy operators and features are currently not supported in MashZone NextGen data feeds.

- Operators
 - Data sources
 - Manual data
 - PPM operator (not alias based)
 - wmOptimize
 - BigMemory
 - Text
 - Create URL
 - Date
 - Move single date
 - Replace date fields
 - Geo locations
 - Aggregate geo locations
 - Retrieve geo locations
- Source locations
 - ARIS Design Server
 - Google Docs
 - Uploaded files (attached files)
- Preprocessing in XML operator

Access permissions

Data feed access permissions of individual users will also be imported. Permissions given to user groups can not be migrated. And the following access permissions options are also not supported by MashZone NextGen.

- pass on view privilege
- create copy
- publish

- access to data from assigned data feeds

Content converted automatically

Several features used in MashZone legacy are handled differently in MashZone NextGen. MashZone NextGen converts and reconfigures these data feeds during the import.

See "[Converted content](#)" on page 17 for details.

Export content in MashZone legacy

Before you can import the desired content into MashZone NextGen you have to export the content in MashZone legacy.

Procedure

1. Open a command shell in the runtool folder of the MashZone legacy.

```
<MashZone legacy installation> \ppmmashzone\server\bin\work\work_mashzone_s\tools\runtool
```

2. Execute the following command.

```
exporttool.bat -user Administrator -password manage -t feeds [-folder subfolder]
```

Now you'll find the exported data feeds in the importexport folder of the MashZone legacy installation. If the `-folder` parameter was used, the exported data feeds are located in an importexport subfolder.

```
e. g.<MashZone legacy installation> \ppmmashzone\server\bin\work\work_mashzone_s\mashzone_data\importexport\sub
```

3. If you want to export file resource aliases, PPM connections, database connections and the Real-time buffer, execute the following command.

```
exporttool.bat -user Administrator -password manage -t aliases [-folder subfolder]
```

Now you'll find the exported aliases in the settings.mzp file in the importexport folder of the MashZone legacy installation. If the `-folder` parameter was used, the exported data feeds are located in an importexport subfolder.

```
e. g.<MashZone legacy installation>\ppmmashzone\server\bin\work\work_mashzone_s\mashzone_data\importexport\sub
```

The required content is exported.

Import content into MashZone NextGen

After exporting the desired content in MashZone legacy you can import automatically the content into MashZone NextGen.

Procedure

1. Copy all .mzp files from the importexport folder, or the specified subfolder into the importexport folder of the MashZone NextGen installation. Alternatively you can copy the .mzp files into a subfolder of the importexport folder.

e.g., *<MashZoneNG installation>* \mashzone\data\importexport\subfolder

2. Open a command shell in the runtool folder of the MashZone NextGen installation.

e. g. *<MashZoneNG installation>* \mashzone\tools\runtool

3. Execute the following command.

```
migrationtool.bat -user Administrator -password manage [-folder
subfolder] [-migration] [-o] [-f]
```

- `-folder`: Name of the importexport subfolder where the .mzp files are located.
- `-migration`: Only with this option the content will be imported. Without this option, the migration tool is executed as a test run.
- `-o`: If a data feed already exists in the system, it will be overwritten with this option.
- `-f`: Only "NOT SUCCESSFUL" result are printed to the console and into the migration log file.

The migration process is finished. On the console and in the migration.log file you'll find the migration results.

Data feeds that could not be imported can be adapted in MashZone legacy and be exported again. See "[Adapting data feeds](#)" on page 15 for details.

The name of a .mzp file contains a timestamp. Make sure you copied the latest one to MashZone NextGen. Additionally remove the old .mzp file from the importexport/[subfolder] before copying the new one. Only one -mzp file per data feed should be provided. You can use different subfolders to avoid having multiple .mzp files for the same data feeds.

Manual steps required

To complete the migration you have to perform manually the following steps after the import process.

Procedure

1. Copy the content of the resources folder of MashZone legacy, e.g., *<MashZone legacy installation>* \ppmmashzone\server\bin\work\work_mashzone_s\mashzone_data\resources, into the resources folder of the MashZone NextGen installation, e.g., *<MashZoneNG installation>* \mashzone\data\resources.
2. Additionally copy all resources that are accessed by the file resource aliases of the MashZone legacy system into the appropriate folders of the MashZone NextGen system.
3. Migrate the JDBC connections if required. See ["Migrating JDBC connections" on page 13](#) for details.

The desired data feeds and the related content are migrated.

Specify permissions

MashZone NextGen version 10.1 enables you to assign access permissions for all data sources to individual user and user groups. Exported data sources of MashZone legacy do not contain access permissions. Which means that all imported data sources can be seen and used by all MashZone NextGen users. Thus, it is recommended to assign manually the access permissions for imported data sources.

By using the **Edit permissions** dialog you can manage access permissions for data source aliases and connections, e.g., PPM connections, file resources, or URL aliases. See MashZone NextGen for details.

Note: You can use the **Presto_AuthenticatedUser** user group to assign access permissions for selected data sources to all registered MashZone NextGen users. The user group includes all authenticated MashZone NextGen users. By assigning the user group to a data source all included user have the relevant access permission for the data source selected.

Note: When upgrading a previous MashZone NextGen to the current version, the access permissions for the data sources are updated automatically.

2 Migrating JDBC connections

You can import the JDBC connections of MashZone legacy (versions 9.5 to 9.12) in MashZone NextGen.

Procedure

1. Copy all drivers located in the `jdbcdriers` folder of the MashZone installation directory to the `jdbcdriers` folder of the MashZone NextGen installation and restart the MashZone NextGen server.
2. Export in MashZone all JDBC connections you want to upgrade. See MashZone Online help for details.

The exported JDBC connections are stored as `mzp` files, starting with `A_DATABASE...`, in folder `<MashZone installation> \ppmmashzone \server \bin \work \work_mashzone_m \mashzone_data \importexport`.

3. Copy all `mzp` files starting with `A_DATABASE...` into the `dbconnections` subfolder of the `importexport` folder located in `<MashZone NextGen installation> \mashzone \data \importexport`.
4. Start the MashZone NextGen server if not already done. Then go to the `runtool` folder (located under `\mashzone \tools`), open a dos command line and call:

```
migrationtool -user Administrator -password manage -folder
dbconnections
```

All JDBC connections from the `dbconnections` folder will be imported into MashZone NextGen.

In the MashZone NextGen Admin Console the JDBC connections are separated in two parts, the driver part and the data source part. You can find all JDBC related items in the **JDBC Configuration** tab of the Admin Console.

If you need to upgrade a connection using a JDBC driver that consists of multiple JAR files, you will have to create a new driver JAR which bundles all the individual files into one single file. After that, you will have to copy the newly created JAR file to the MashZone NextGen installation.

3 Adapting data feeds

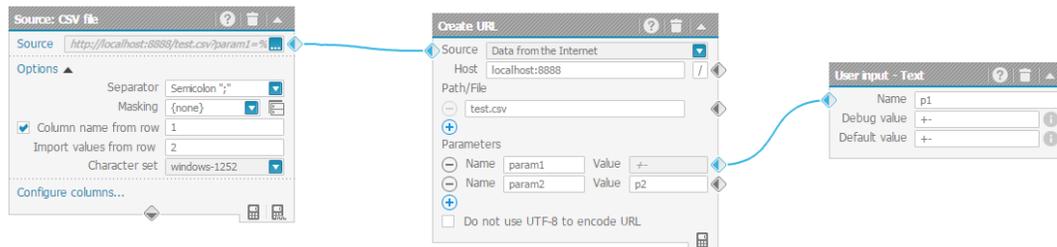
We recommend to adapt data feeds that can not be imported in MashZone NextGen.

You have to reconfigure data feeds containing not supported options or features in MashZone legacy before you can import them in MashZone NextGen.

The following examples show how to adapt not supported data feeds.

Create URL operator

The **Create URL** operator can be adapted as shown in the following example.



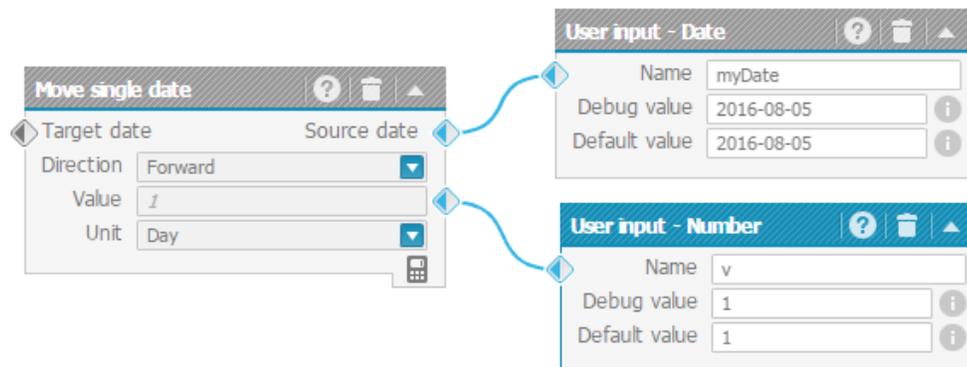
The create URL operator delivers: `http://localhost:8888/test.csv?param1=%2B-¶m2=p2`.

It can be exchanged by a merge single text operator as follows.

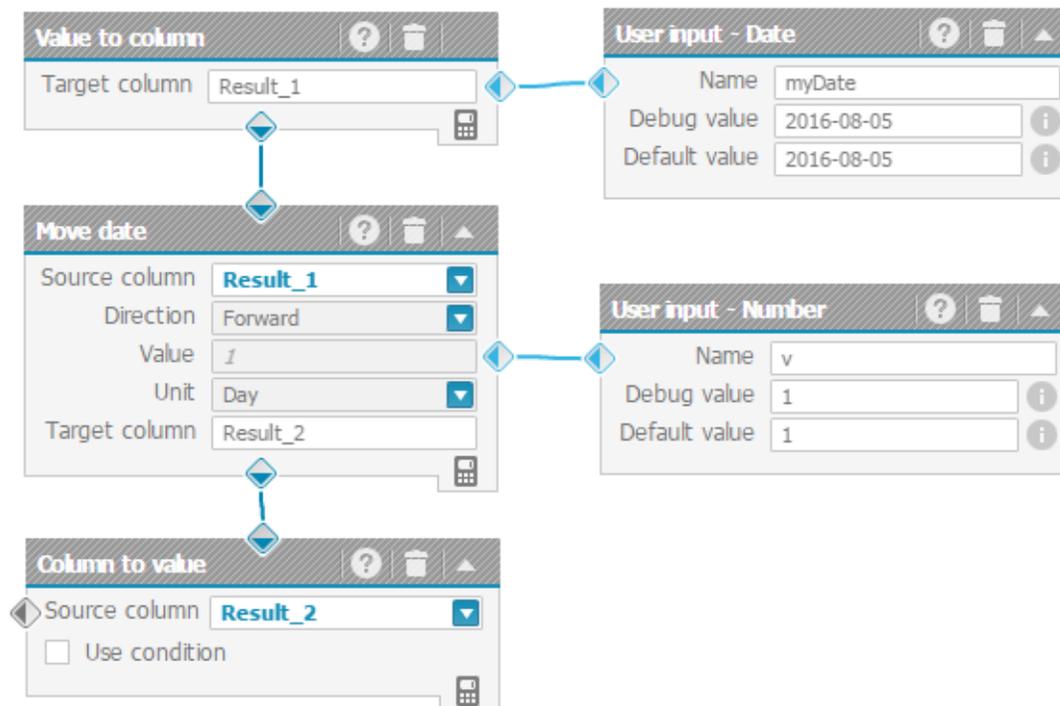


Move single date operator

The **Move single date** operator can be adapted as shown in the following example.



Instead of using the move single date operator it's possible to use the move date operator as shown below.



4 Converted content

Several features used in MashZone legacy are handled differently in MashZone NextGen. MashZone NextGen converts and reconfigures these data feed during the import. The data feed layout is also adapted automatically by MashZone NextGen.

The following features are adapted automatically in MashZone NextGen.

Localized column names for CSV and Excel operator

MashZone legacy used localized column names if either a column name could not be determined from the source or cross table was used in Excel operator (values, horizontal/vertical iteration). MashZone NextGen only uses not localized names for such columns. If a data feed with localized column names is migrated, MashZone NextGen uses the not localized name as initial name and the localized name as new name. This is done only if no new name was specified for that column.

Name	New name	Type	As text
<input checked="" type="checkbox"/> Unbenannte Spalte 1		Text	<input type="checkbox"/>
<input checked="" type="checkbox"/> b		Number	<input type="checkbox"/>

→

Name	New name	Type	As text
<input checked="" type="checkbox"/> b		NUMERIC	<input type="checkbox"/>
<input checked="" type="checkbox"/> Unnamed Column (1)	Unbenannte Spalte 1	TEXT	<input type="checkbox"/>

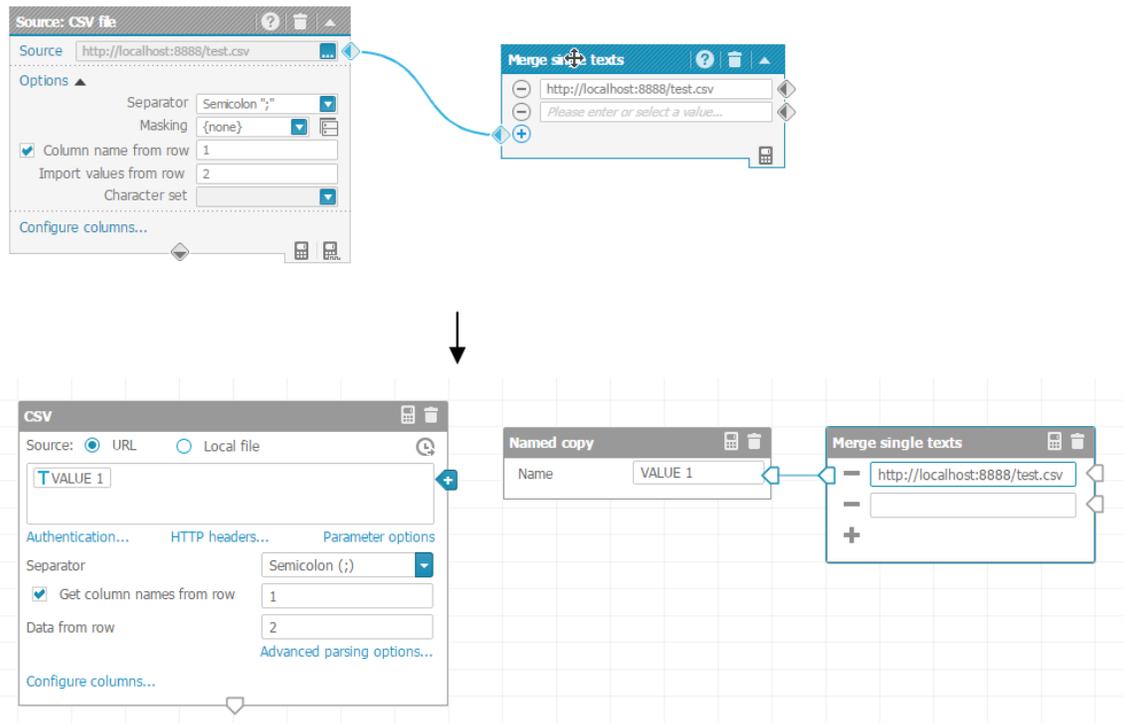
Name	New name	Type	As text
<input checked="" type="checkbox"/> Produkt		Text	<input type="checkbox"/>
<input checked="" type="checkbox"/> Horizontale Iterati	my iteration	Text	<input type="checkbox"/>
<input checked="" type="checkbox"/> Werte		Number	<input type="checkbox"/>

→

Name	New name	Type	As text
<input checked="" type="checkbox"/> Horizontal iteration	my iteration	TEXT	<input type="checkbox"/>
<input checked="" type="checkbox"/> Produkt		TEXT	<input type="checkbox"/>
<input checked="" type="checkbox"/> Values	Werte	NUMERIC	<input type="checkbox"/>

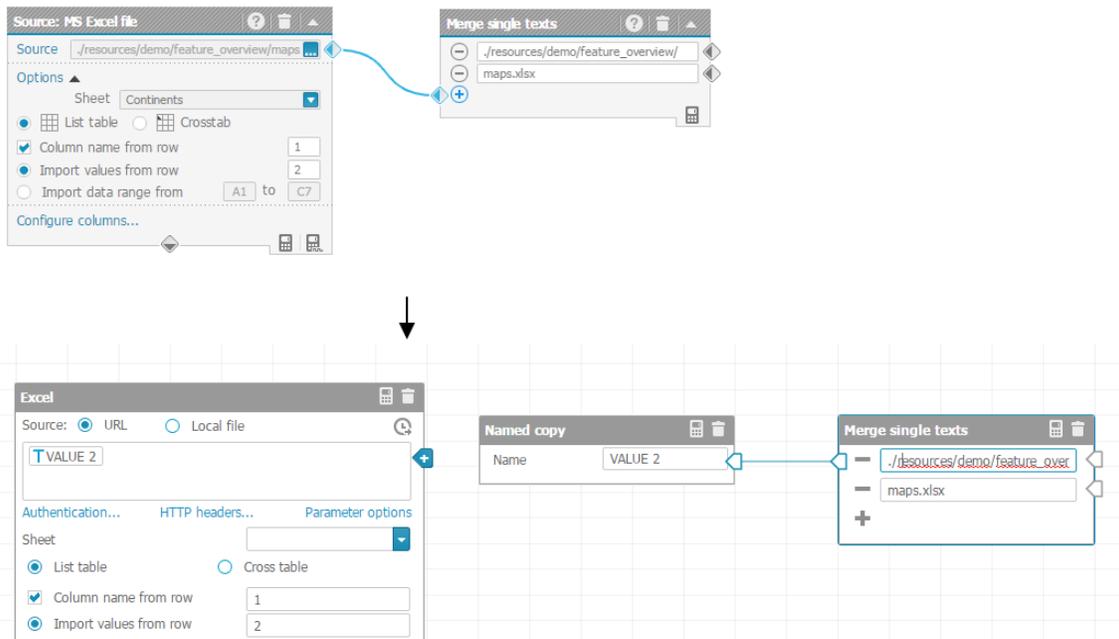
Connecting single values in data source operators

The data source operators in MashZone NextGen don't support edges to connect single values with the source operator. The same applies to the SQL statement in the JDBC operator. Instead a named copy operator is created during the import in MashZone NextGen, as shown below.

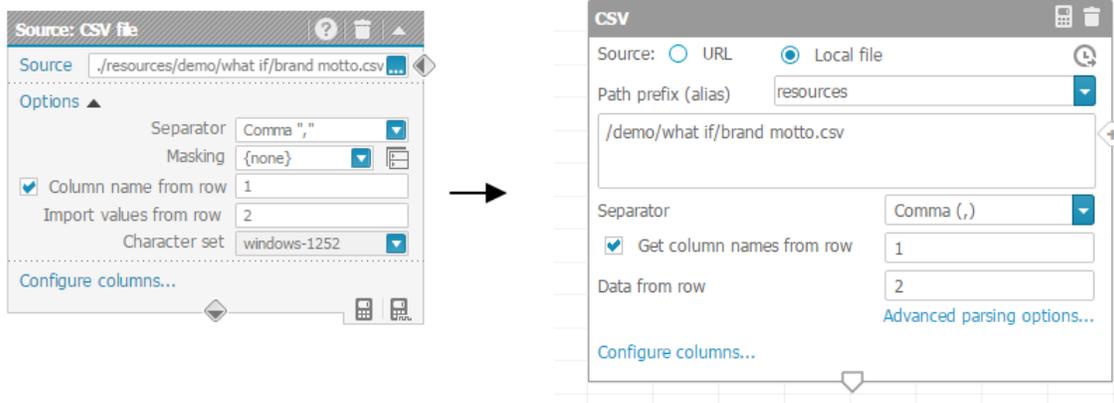


Source location

If in MashZone legacy a single value operator is used to specify the data source for an operator, the source location for that operator will be changed to URL in MashZone NextGen. If the source location was set to local file in genuine data feed, it has to be adapted in the new feed editor.

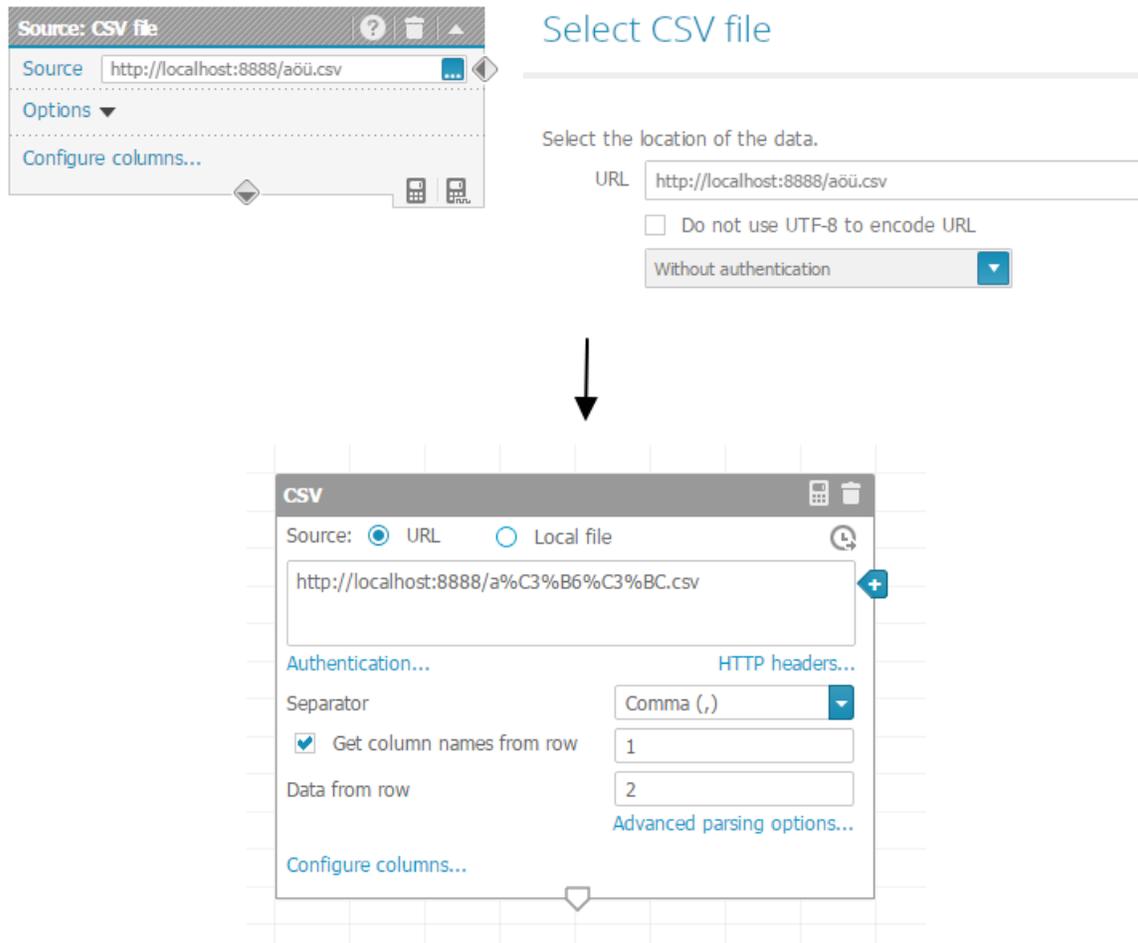


If a local file was specified directly in the operator, the appropriate alias will be selected in the drop-down menu. The input field only contains the additional path information.

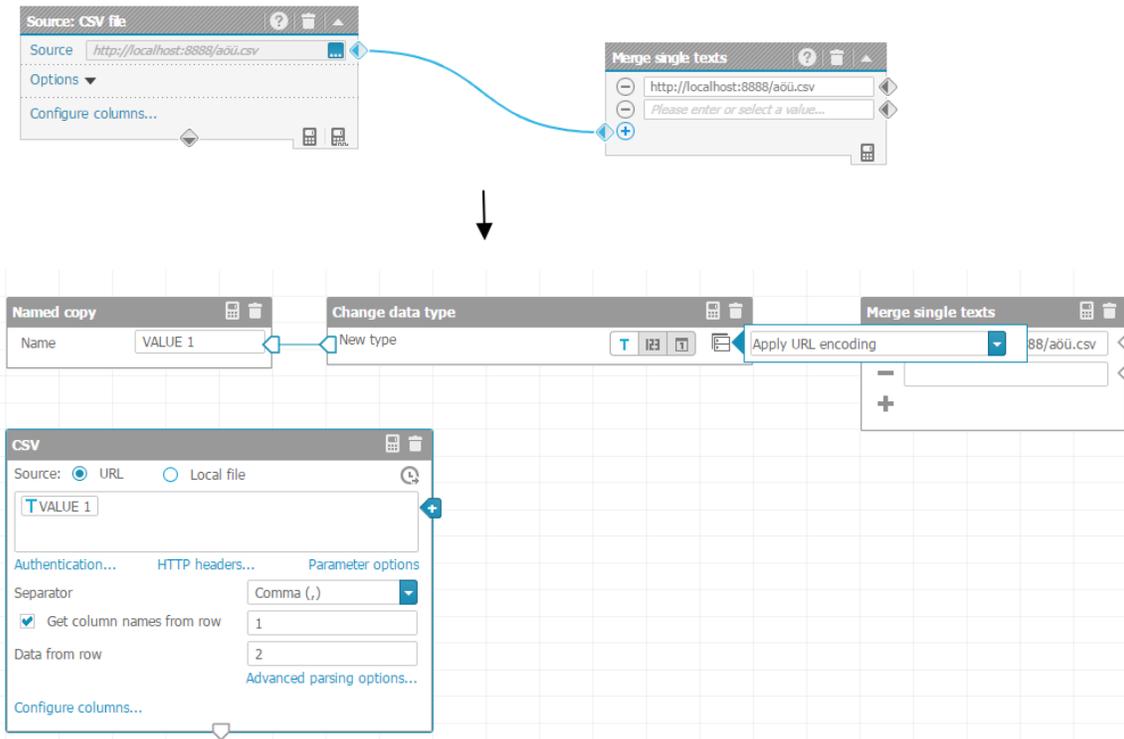


URL encoding of source operators

MashZone NextGen does not support enabling or disabling URL encoding in data source operators. If a source operator in the genuine data feed used URL encoding and the URL was specified directly in the operator, the URL will be automatically URL encoded in the input field.

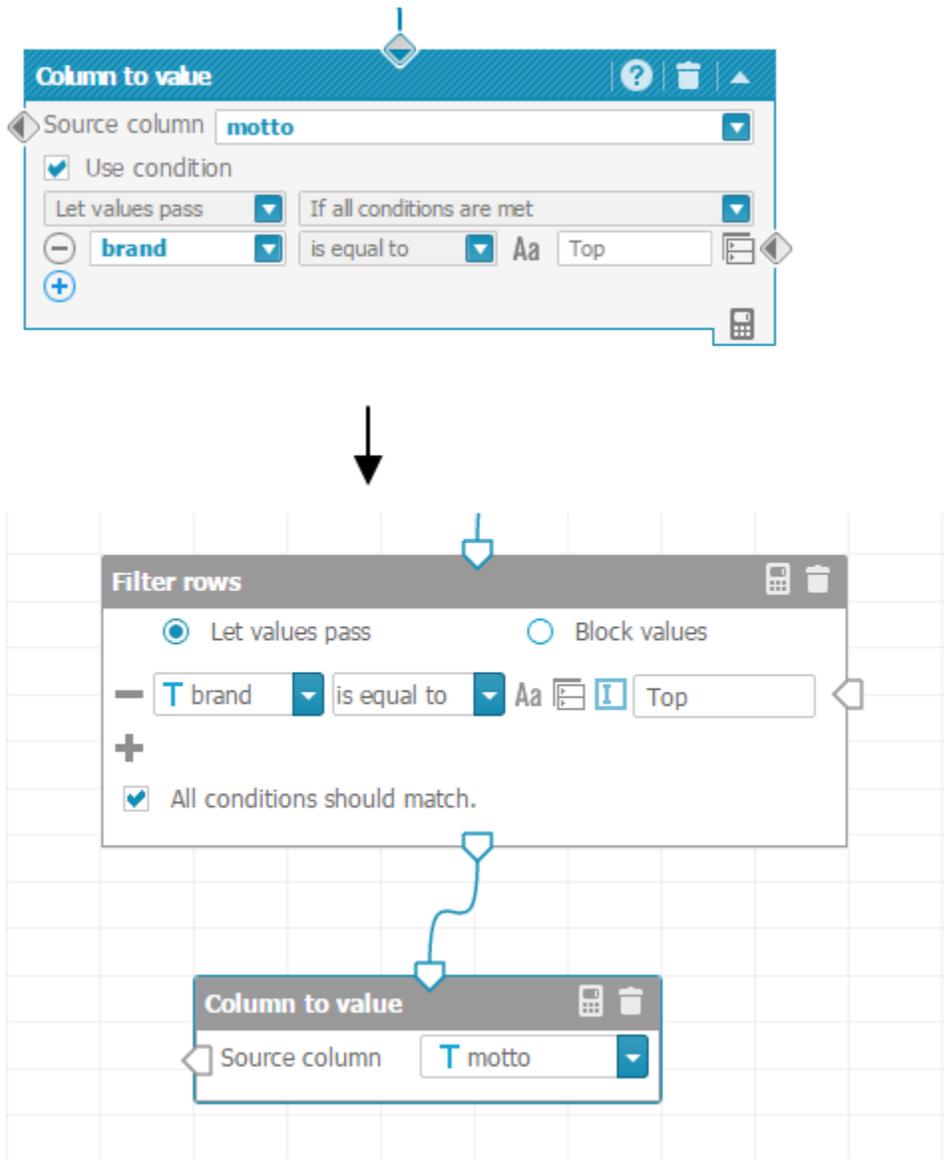


If the URL was specified with a single value operator, a change value type operator will be created with apply URL encoding option.



Condition in Column to value operator

MashZone NextGen does not support conditions in the **Column to value** operator. If the genuine data feed contains a **Column to value** operator with condition, a filter operator which uses the appropriate conditions is created.



PPM operator

MashZone NextGen does not support SSO for the **PPM** operator. If SSO was used, it is necessary to adapt the credentials (user/password) in the operator.

XML operator

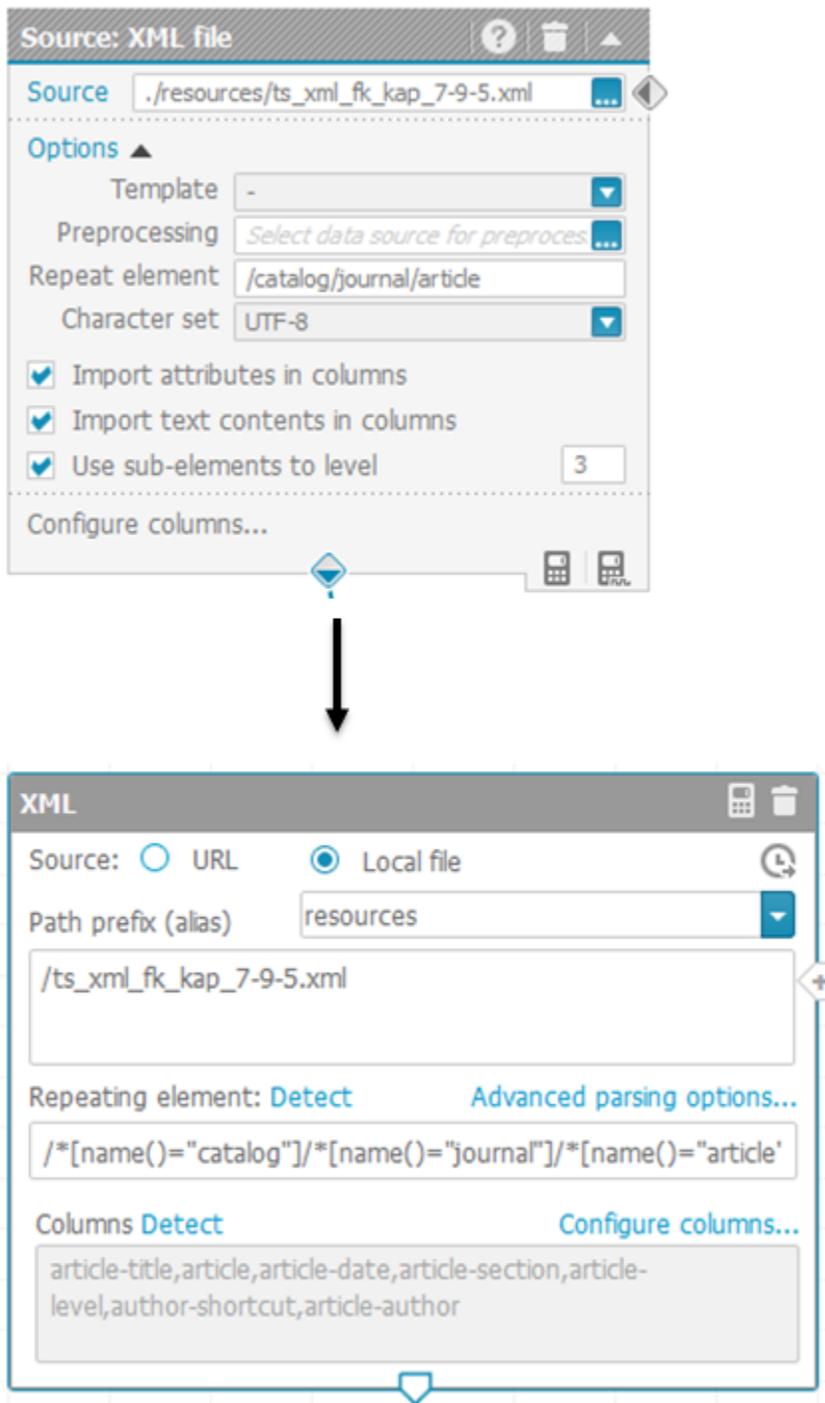
The **XML** operator in MashZone NextGen has been changed. Some options that are not longer required have been removed.

- **Import attributes in columns** checkbox
- **Import text contents in columns** checkbox
- **Use sub-elements to level** checkbox and its associated input field

Configuration like this, where you decide on what columns you want to see, are now done in the XML-specific **Configure columns** dialog.

Character set is not longer configurable, it is determined automatically by inspecting the data source instead.

The **Preprocessing** functionality has been removed, as well as the **Template** drop-down box. There is no replacement in the new operator for these. Data feeds using the **Preprocessing** mechanism will be rejected at migration time.



The existing column configuration is migrated and re-usable in the new operator. The **Repeating element** is automatically updated to reflect the different logic of the new operator.

5 Adapt PPM connections

If you import a dashboard or a data feed of an older version than MashZone NextGen 10.0 you have to reconfigure the PPM connection specified for your PPM data source.

Previously, the PPM connection in MashZone NextGen (and MashZone legacy) had to be configured with a context that the user was permitted to configure freely within PPM client setup (e.g., "ppm/API_umg_en", but also "ppm/myFavoriteContextName"). With MashZone NextGen version 10.0 the context of the PPM connection has been replaced by the PPM client name. Effectively, the **Context** field is renamed to **Client** and it does no longer contain a configurable context, but instead just the <client>. Therefore, the PPM connection settings have to be adapted manually.

Procedure

1. Open the **PPM connection settings** in the MashZone NextGen administration. See MashZone NextGen online help for details.
2. Specify the PPM client name of your PPM connection in the **Client** field.
3. Click **Save**.

Your changes are applied.

6 Migrate custom dashboard components

You can migrate your custom dashboard components (widgets) from MashZone NextGen version 10.0 to the current MashZone NextGen version.

Custom dashboard components directory version MashZone NextGen 10.0:

```
<MashZoneNG installation> \apache-tomee-jaxrs\webapps\presto\hub\dashboard  
\widgets\customWidgets\myCustomWidget
```

Custom dashboard components directory version MashZone NextGen 10.1:

```
<MashZoneNG installation> \apache-tomcat\webapps\mashzone\hub\dashboard  
\widgets\customWidgets\myCustomWidget
```

Procedure

1. Copy the folder containing your custom dashboard components to the new MashZone NextGen installation directory.
2. In case dashboard.html was changed, e.g, in order to import third party libraries for a custom dashboard component, copy all needed changes to the new dashboard.html file.
3. Copy all files required by your custom dashboard, such as third party libs, to the corresponding directory of the current MashZone NextGen directory.
4. MashZone NextGen 10.1 comes with a new version of AngularJS (1.6.4) and Angular UI (2.5.0). So you also may need to update your custom components to work with the updated libraries.

Angular 1.6: Some deprecated APIs have been removed. For example \$http's deprecated custom callback methods success() and error() have been removed. For detailed information please have a look at the official migration guide: <https://docs.angularjs.org/guide/migration#migrating-from-1-5-to-1-6>.

Angular UI 2.5.0: In case your custom component uses Angular UI controls or services like combo boxes, tool tips etc. Please have a look at the official API documentation and the migration guide. In the new Angular UI version components have the "uib" prefix. For further information please have a look at the official website <https://angular-ui.github.io/bootstrap/> and the migration guide <https://github.com/angular-ui/bootstrap/wiki/Migration-guide-for-prefixes>.