# Value Mapping with MII 12.2

## Applies to:

SAP Manufacturing Integration and Intelligence (SAP MII) Version 12.2. For more information, visit the Manufacturing homepage.

## Summary

How many times do we create mapping tables to convert Material Number, Equipment Number etc from the Shop Floor representation to the ERP representation and vice versa while creating MII Applications? How many times to need to pad or remove leading zeroes to such numbers so that they will work in ERP? The answer to this would million or zillion. Till MII 12.1 there is no out of the box function that can deliver this. In SAP MII 12.2 a new functionality called "Value Mapping" is added. This document will explain how Value Mapping can help solve this eternal problem.

**Authors:** Partha Sarathi Roy Chowdhury

**Company:** SAP labs, Bangalore

**Created on:** 01 February 2012

## Author Bio

Partha Sarathi Roy Chowdhury is with SAP from last 8 years and based out of SAP labs, Bangalore. He is the Product Architect of Manufacturing Integration and Intelligence (MII). Prior to joining SAP, he spent 2 years with Wipro Technologies.

## Table of Contents

## Introduction

While developing MII applications that integrate the shop floor systems and ERP numerous times we feel the need to convert Equipment, Material, Work Center numbers to and from the ERP IDs to the IDs used in the shop floor systems. For this currently MII developers create mapping tables which store the mapping between shop floor IDs and ERP IDs. At runtime these mappings are read and converted from one to the other.

Numerous times we also need to pad zeros to an ID before we can send it to ERP as the RFC/BAPI we call requires it in that format. Vice versa the leading zeros often need to be stripped before they can be used. This again requires an extra step to be done by the developers which may be error prone and time consuming.

Lastly there are several times we need to change formats like Date Time formats, number formats etc for separate systems and this is again a very tedious task.

To solve all the above use cases and more MII 12.2 offers the Value Mapping functionality. Value Mapping in its simplest form Value Mapping allows you to replace values inside an XML by using XPath lookup but it allows you to provide various combinations for each replacement and applies them in order of matching. With Value Mapping the tedious process of mapping, converting, padding

## Defining the Value Mapping

In MII 12.2 every project already has a file called ValueMap.vmap in the Meta-Inf tab. The path to the file is <Project Name>/Meta-Inf/ValueMap.vmap. This file contains the rules for modifying/substituting the values inside an XML file.

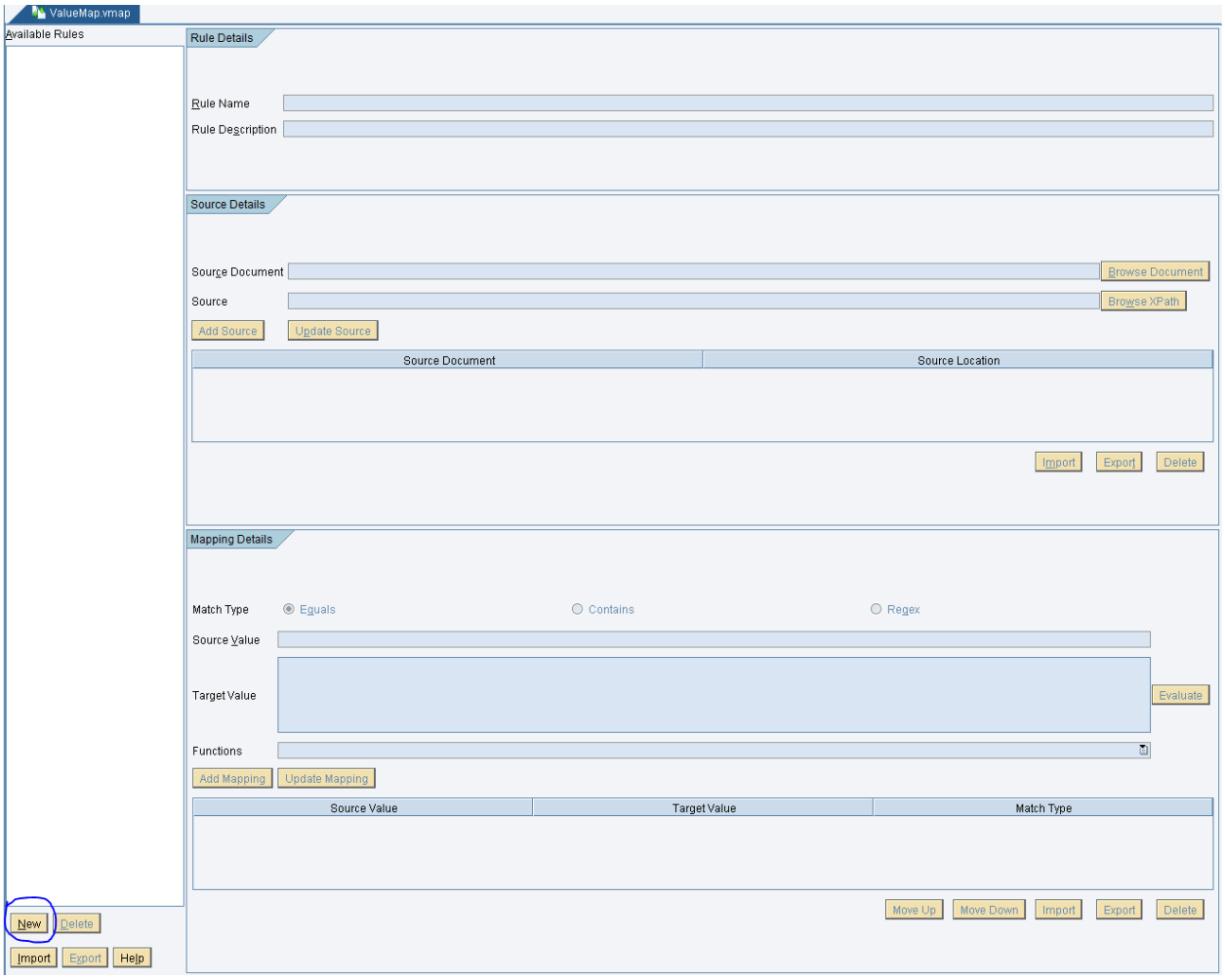
There can be many rules created in a Value Map file but there can only be one Value Map file per project. A Rule is defined with a Name and Description determining what the Rule stands for or does. Inside a Rule you can define multiple source objects. A Source Object is the XML or XSD file which provides the XPath for the fields which are candidates for modifications as mentioned above.

Each Source Object is composed of a Source Document Location and a Source XPath pointing to the field that you need to update. Each Source Object also contains multiple Mapping Objects. Mapping objects or simply Mappings are the conditions for updating the matching field. Each Mapping contains a Source Value, Target Value and a Mapping Type.

Each Mapping states what should be done when updating a field. Source Value is the “from value” in the mapping. Target Value is “To Value” of the mapping. So a Mapping means when the Source Value is “abc” set it to “xyz” in the XML. Where this change is to be done is specified in the XPath of the source object. Now not all cases are as simple. So we have a Match Type which can be either Equals or Contains or Regex. In Equals case the Source Value configured has to match exactly with the value in the XML at runtime. In case of Contains as long as the value in the XML should contain the Source Value defined in the Mapping. In Regex case you can provide any Regex expression to compare the actual value and the configured Source Value. As we all know Regex can be very powerful matching tools we can pretty match do any king of Pattern Matching. To find out more on Regex Patterns please check the References section.

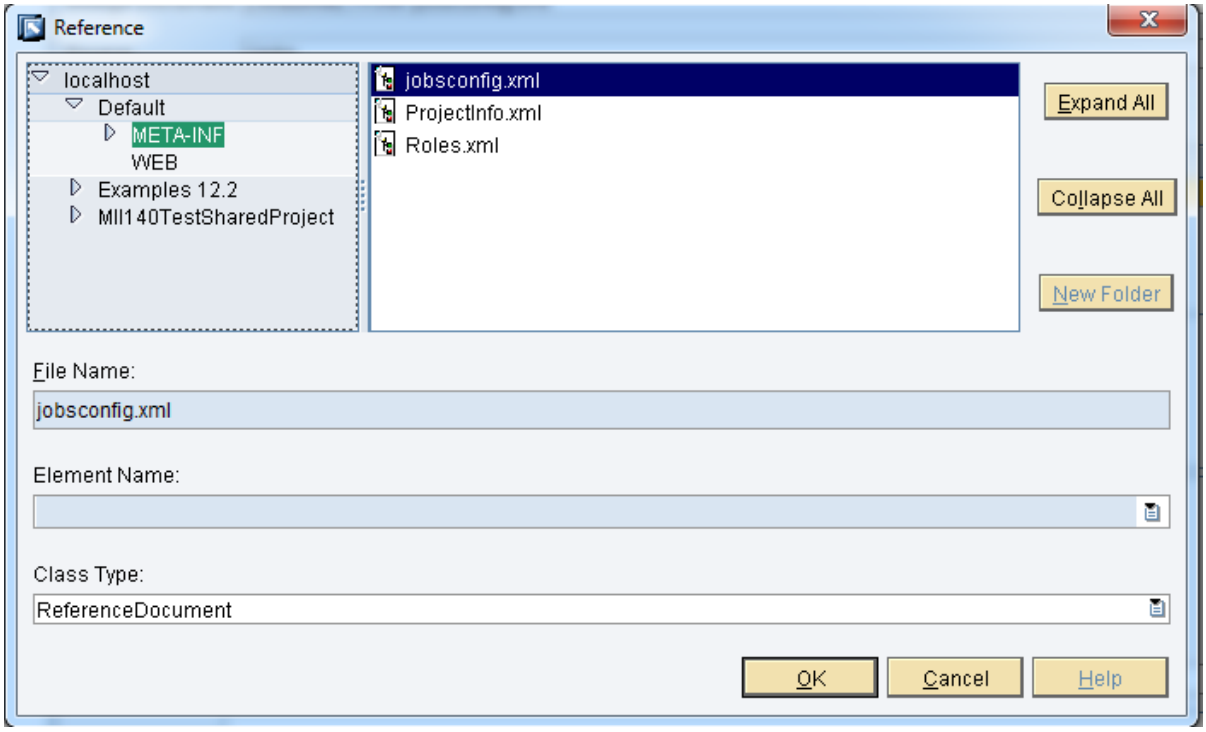
## Create a Rule

Go to the MII Workbench and choose the Meta-Inf Tab. Open your project and open the ValueMap.vmap file. To create a Rule click on the New button below the Available Rules panel as shown in the screenshot below.

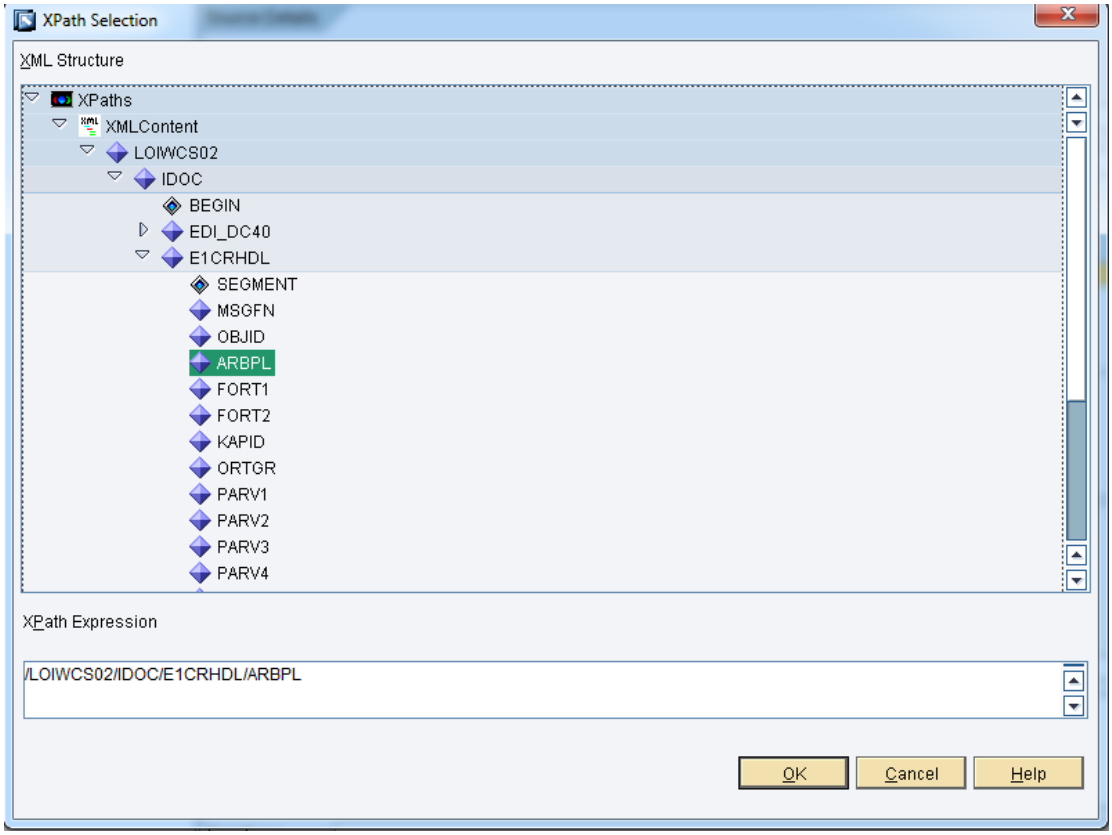



Once you click on New button the right hand side panel becomes editable. Provide a Name and Description for the Rule.

Next you need to add a Source. To do this you should have an XML file or an XSD imported to your project or any other project in your MII System. If the file is not already there import the file into a folder in your project. Click on the “Browse Documents” button and select the XML or XSD file you want to use.

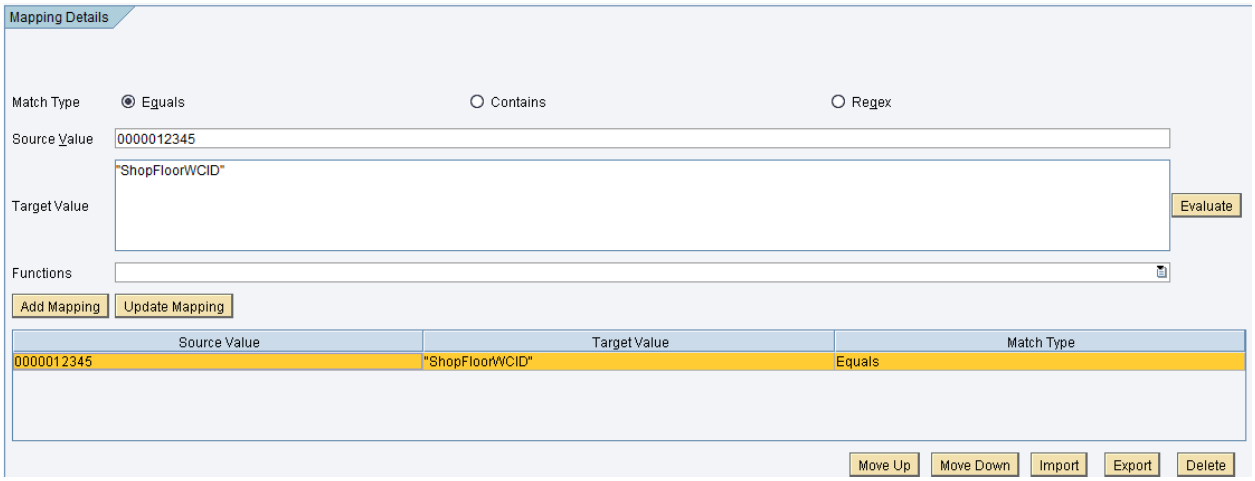



Next you need to choose the XPath inside the XML or XSD which is relevant for mapping or changes. Please note that you will need to choose either “Reference Document” for XML or “Reference Schema” for XSD in the pop up above to see the respective types. This may be an ID field that needs to be mapped or padded or this may be a Timestamp field which needs to be reformatted. Click on the button “Browse XPath” to get the XML or XSD displayed as a Tree and select the node or attribute you want to modify. Alternately you can also type in the XPath if necessary. In the example below I have used the LOIWCS XML file.



Once both the Source Document and Source XPath are filled add this entry by clicking Add Source button. If you want to change this entry select the entry from the below table and make changes. Once done please click the Update Source button.

Next you need to add all the scenarios for the current source (in this case the Work Center ID). I might want to change the ID from ERP ID to the Shop Floor ID. In this case I want exact matches. So I will choose Match Type as Equals. Next I will add the possible Work Center IDs in the Source Value field. This is similar to creating the mapping table you would otherwise need. So if you have multiple Work Centers then add them as separate mappings. In the Target Value field enter the Shop Floor IDs for the corresponding Work Center IDs. In this case you do not need to select anything in the Functions list. Click on Add Mapping and add this to the Mapping Table. If there are more than one Work Center repeat this process till all entries are done.




Let's take another case where I want to strip the leading zeros from the ARBPL field. Let's see how to do this. Assume that our Work Center ID is 000000000010002906. The pattern for this is n number of zeros followed by some numbers (or in some cases even characters). So we would need a Regex pattern to match this. The Regex pattern would be "0*.*" (without the quotes). The Target value should take whatever is the source value and strip all the leading zeros and leave only the characters that matter. This needs use of some functions in MII. So our entry should look like

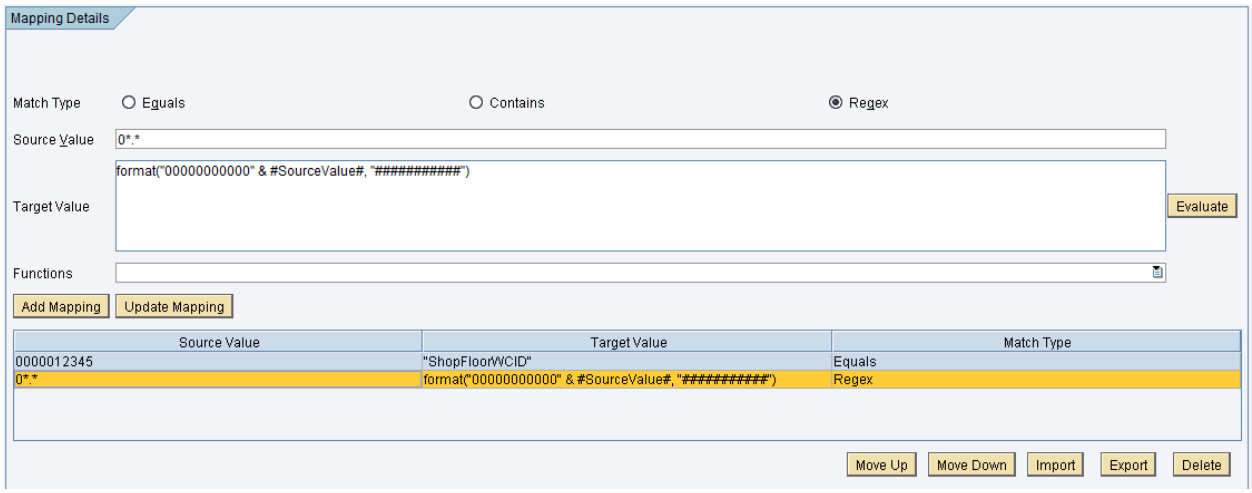
Match Type: Regex

Source Value: 0*.*

Target Value: format("00000000000" & #SourceValue#, "############")

Note that the #SourceValue# command is to tell the execution engine to pass the value of the node at runtime to the function. In this case whatever is the value in the ARBPL field would be passed to the format function. The Target Value field is same as the Expression Editor in Link Editor of MII and supports all expressions/functions supported by the MII Expression Engine.

After making the above entry the Mapping table looks as below



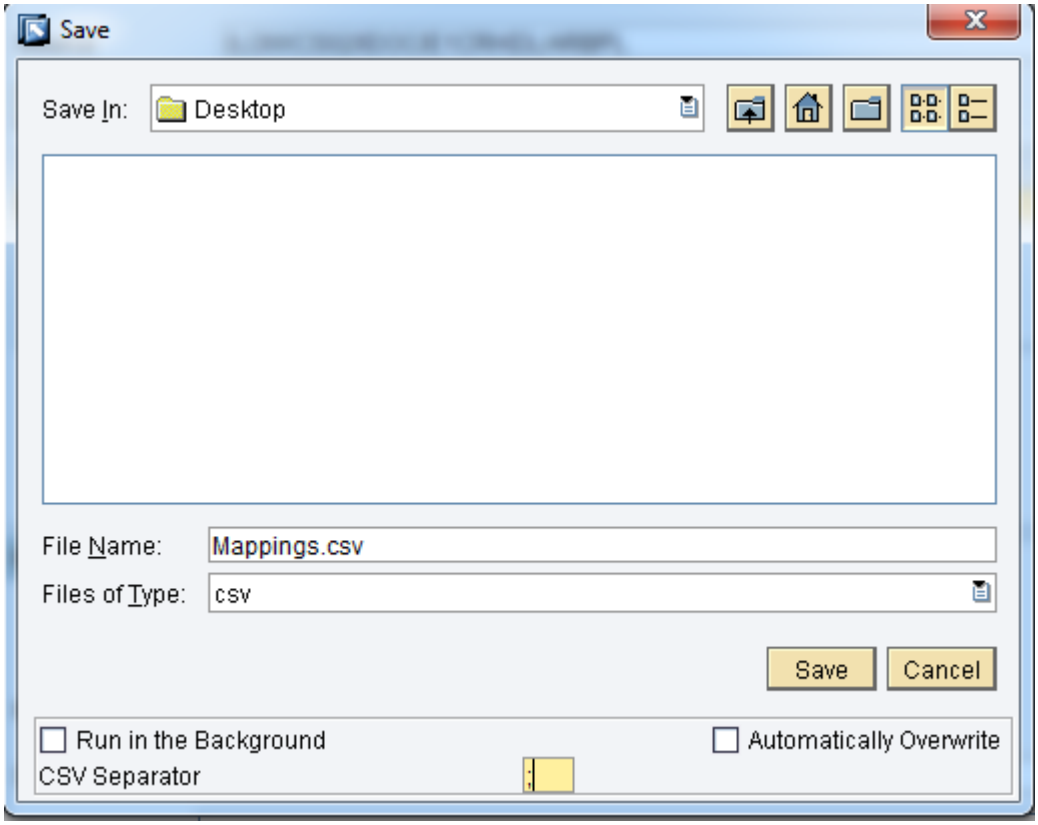

So now we have two mappings created for the same field. Question is which one will be applied at runtime? The rule here is that the execution checks the mappings top down. If the first rule matches the value in the source node then this mapping is applied. The remaining mappings are ignored. So in this case if the value of ARBPL field is 0000012345 then the first mapping is applied and not the second mapping. If the value is 000000000012345 then the second mapping is applied and not the first one as this does not match exactly to the source value in the first mapping. In order to ensure that mappings are applied correctly you should check that they are in the correct order and that they are not overlapping. You can change the order of the mapping by using the Move Up and Move Down buttons.

In the above process you can create multiple Rules. You can add multiple sources to each rule and then you can add multiple mappings per source. Once everything is added you need to save the whole file.

## Import and Export

In the above screenshots you would have seen some other buttons like Export and Import for each Rule, Source and Mapping. This means that you can Import and Export not only the Value Map file (like any other file in MII) you can also import and export each rule, source or mapping. This means that you can export a single (or more than one) rule from a value map file in one project and import them in another project (may be on different MII). You can also export a source or mapping and import them in the value map file of another project. This adds tremendous flexibility in reusing parts of a value map and not having to recreate them every time.

Note that only the Value Map file is exported as an XML. The Rules, Sources and Mappings are all exported and imported as CSV files. If you select one or more of the objects and click export the Export dialog opens up. You can choose the location to save the file. Additionally you can select the CSV separator. By default this value is ‘,’ but you would need to change this to any other character in case comma exists in any entry selected for exporting. For example our second mapping consists of a Target Value which has comma. Hence we will set the separator to ‘;’ as shown below.




If you export a Rule then all Sources and Mappings below it are exported. If a Source is exported then the source and all its mappings are exported. If a Mapping is exported then only the mapping is exported and nothing else.

The formats of the CSVs are given below

### Mappings are exported as

SourceValue;TargetValue;MatchType

0000012345;ShopFloorWCID;1

0*.*;format("00000000000" & #SourceValue#, "############");3

### Sources are exported as

SourcePath;SourceDocument;SourceValue;TargetValue;MatchType

/LOIWCS02/IDOC/E1CRHDL/ARBPL;Default/WEB/LOIWCS.xml;0000012345;ShopFloorWCID;1

/LOIWCS02/IDOC/E1CRHDL/ARBPL;Default/WEB/LOIWCS.xml;0*.*;format("00000000000" & #SourceValue#, "############");3

### Rules are exported as

RuleName;RuleDescription;Source;SourceDocument;SourceValue;TargetValue;MatchType

Rule_1;;/LOIWCS02/IDOC/E1CRHDL/ARBPL;Default/WEB/LOIWCS.xml;0000012345;ShopFloorWCID;1

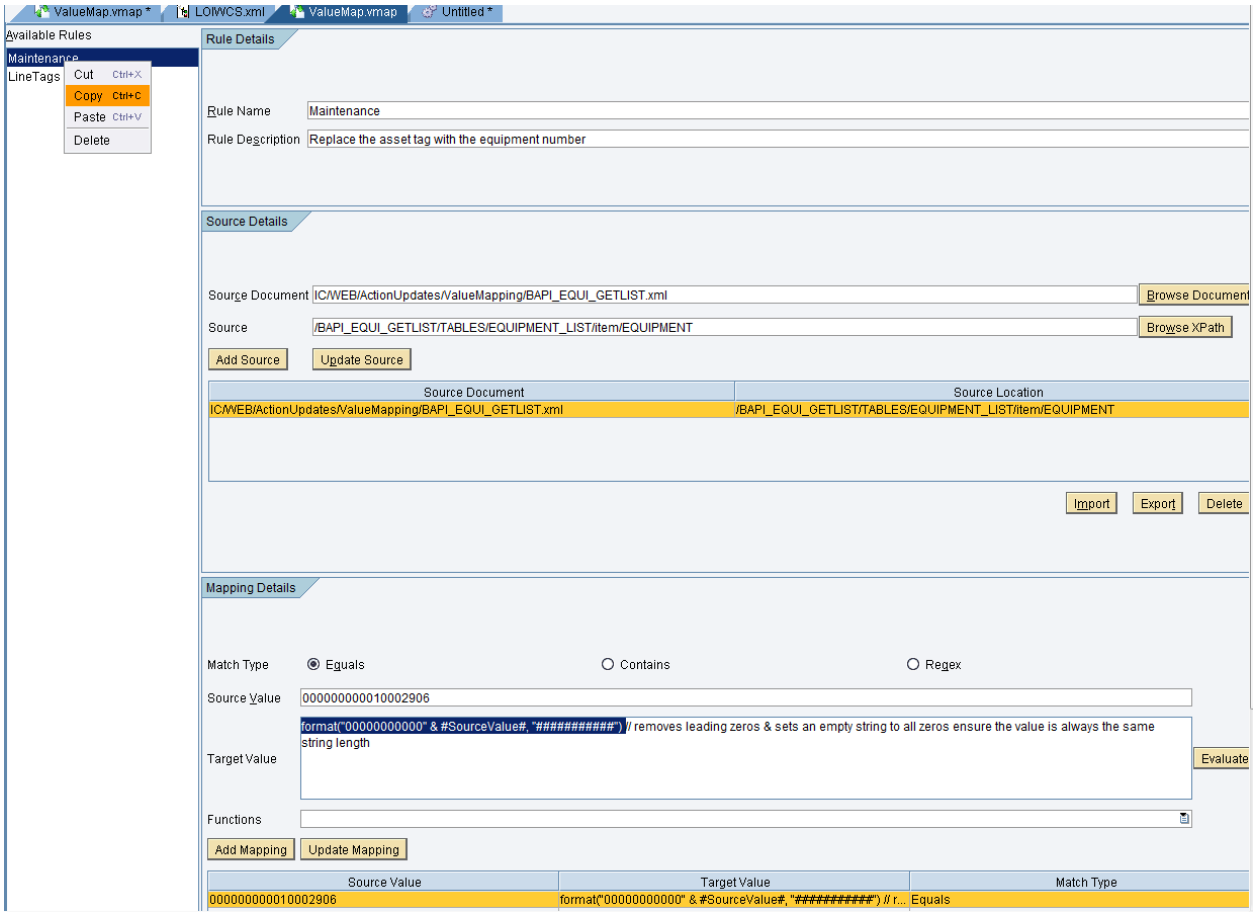
Rule_1;;/LOIWCS02/IDOC/E1CRHDL/ARBPL;Default/WEB/LOIWCS.xml;0*.*;format("00000000000" & #SourceValue#, "############");3

To import sources and mappings you need to select the Rule or Source and click Import button below it. For example to import a Source select the Rule to which you want to add the Source and click Import under the Source Details section. Note that the CSV separator should be provided correctly to ensure that the import happens correctly.

## Cut/Copy and Paste

In addition to the Import and Export of Rules, Sources and Mappings it is also possible to Cut or Copy them from one Value Map file and paste them in another Value Map file. To Cut or Copy a Rule or Source or Mapping choose Cut or Copy from the Context menu. This is available in the Available Rules list and the Source and Mapping Tables. Paste is only enabled if something has been copied or cut from one location.

The behavior of Copy and Paste for Rule, Source and Mapping is same as the export. The context menu also allows Delete of a record.



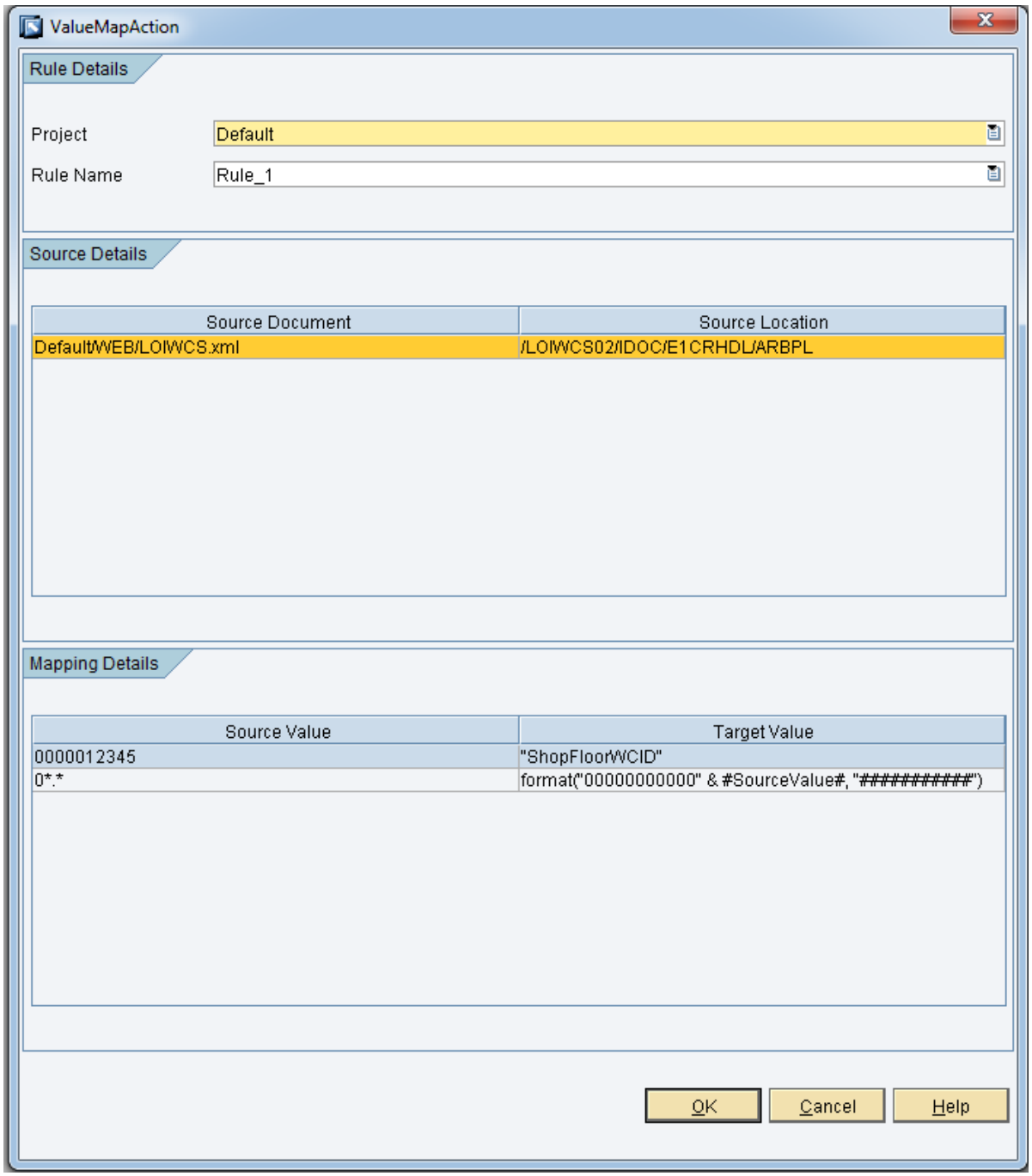
## Using the Value Map

In order to use the Value Map at runtime 2 action blocks are available. These are the Value Map action and the Value Lookup action. These actions are available in the XML Functions action category.

### Value Map Action

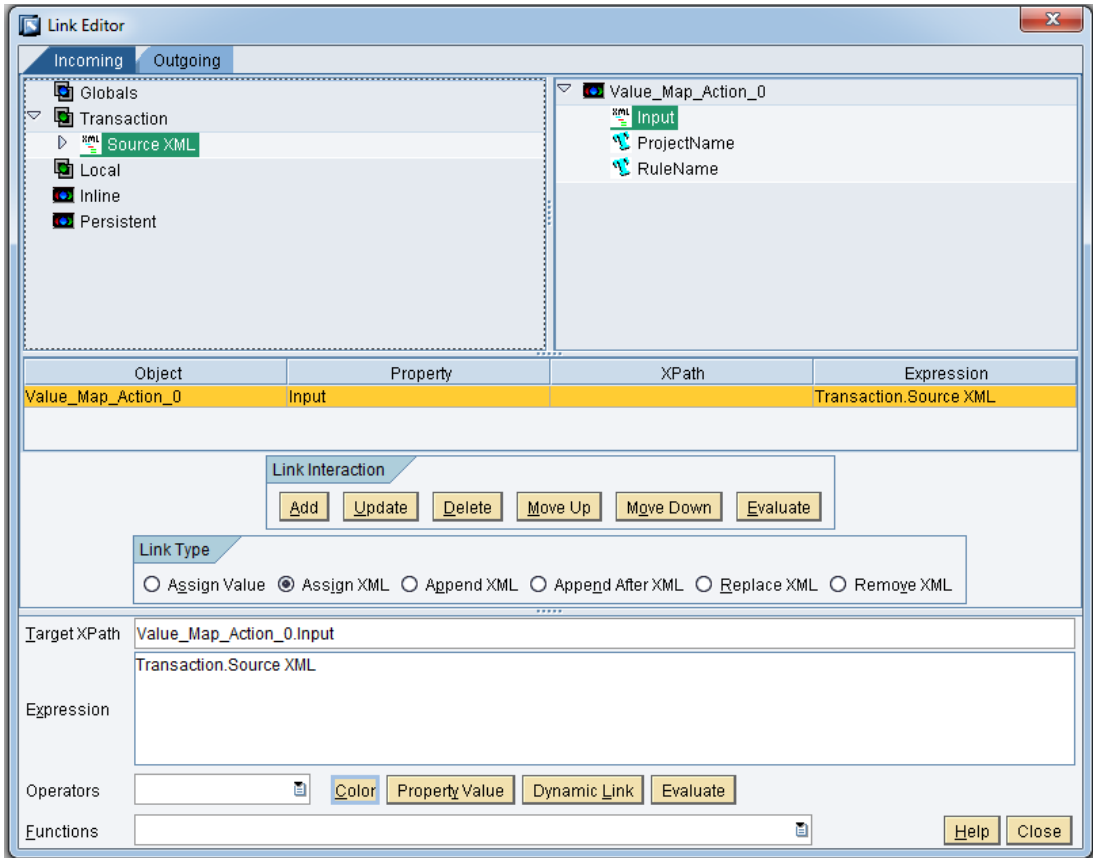
The Value Map action takes in the Source XML and applies the rules and creates the target XML based on the Mappings in the rule. All Source XPaths in a Rule are applied to the source XML and the target XML is generated.

In the configuration screen of the Value Map action you need to select the Project whose Value Map file you want to use and then select the Rule you want to apply. You need not choose Sources or Mappings.

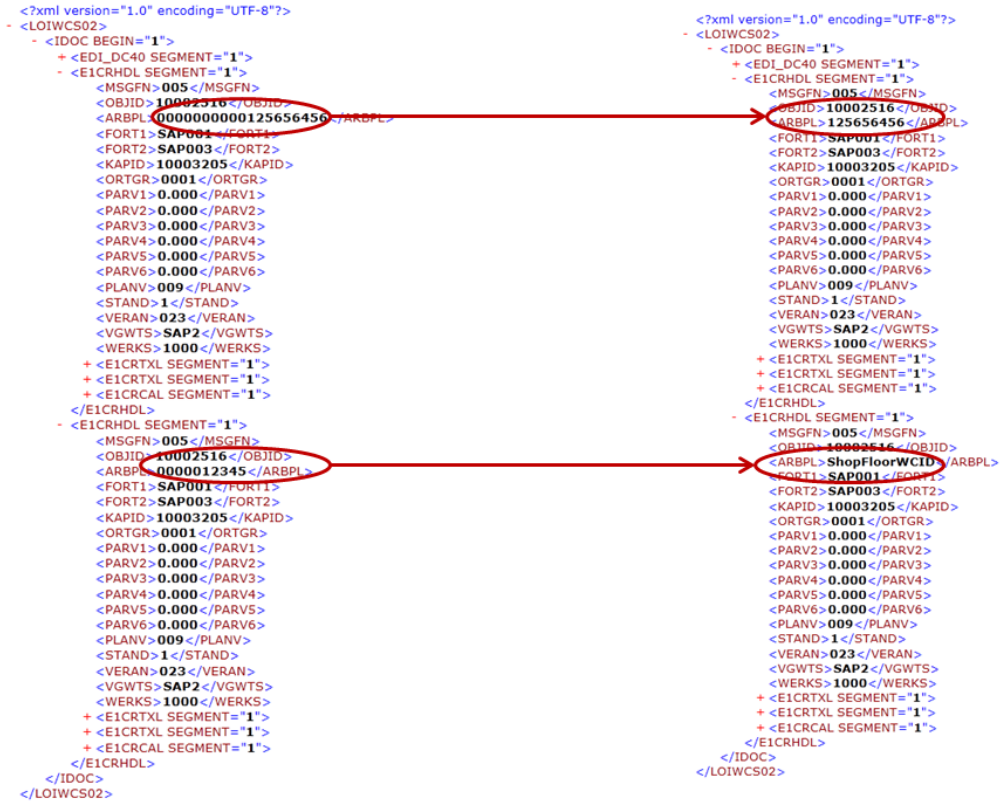


Next in the Link Editor for this action you need to assign the Source XML. For our example I have created a Transaction property called Source XML which I map to this field. The XML looks as below:

```
<?xml version="1.0" encoding="UTF-8"?>
- <LOIWCS02>
  - <IDOC BEGIN="1">
    + <EDI_DC40 SEGMENT="1">
    - <E1CRHDL SEGMENT="1">
        <MSGFN>005</MSGFN>
        <OBJID>10002516</OBJID>
        <ARBPL>125656456</ARBPL>
        <FORT1>SAP001</FORT1>
        <FORT2>SAP003</FORT2>
        <KAPID>10003205</KAPID>
        <ORTGR>0001</ORTGR>
        <PARV1>0.000</PARV1>
        <PARV2>0.000</PARV2>
        <PARV3>0.000</PARV3>
        <PARV4>0.000</PARV4>
        <PARV5>0.000</PARV5>
        <PARV6>0.000</PARV6>
        <PLANV>009</PLANV>
        <STAND>1</STAND>
        <VERAN>023</VERAN>
        <VGWTS>SAP2</VGWTS>
        <WERKS>1000</WERKS>
      + <E1CRTXL SEGMENT="1">
      + <E1CRTXL SEGMENT="1">
      + <E1CRCAL SEGMENT="1">
      </E1CRHDL>
    - <E1CRHDL SEGMENT="1">
        <MSGFN>005</MSGFN>
        <OBJID>10002516</OBJID>
        <ARBPL>ShopFloorWCID</ARBPL>
        <FORT1>SAP001</FORT1>
        <FORT2>SAP003</FORT2>
        <KAPID>10003205</KAPID>
        <ORTGR>0001</ORTGR>
        <PARV1>0.000</PARV1>
        <PARV2>0.000</PARV2>
        <PARV3>0.000</PARV3>
        <PARV4>0.000</PARV4>
        <PARV5>0.000</PARV5>
        <PARV6>0.000</PARV6>
        <PLANV>009</PLANV>
        <STAND>1</STAND>
        <VERAN>023</VERAN>
        <VGWTS>SAP2</VGWTS>
        <WERKS>1000</WERKS>
      + <E1CRTXL SEGMENT="1">
      + <E1CRTXL SEGMENT="1">
      + <E1CRCAL SEGMENT="1">
      </E1CRHDL>
  </IDOC>
</LOIWCS02>
```

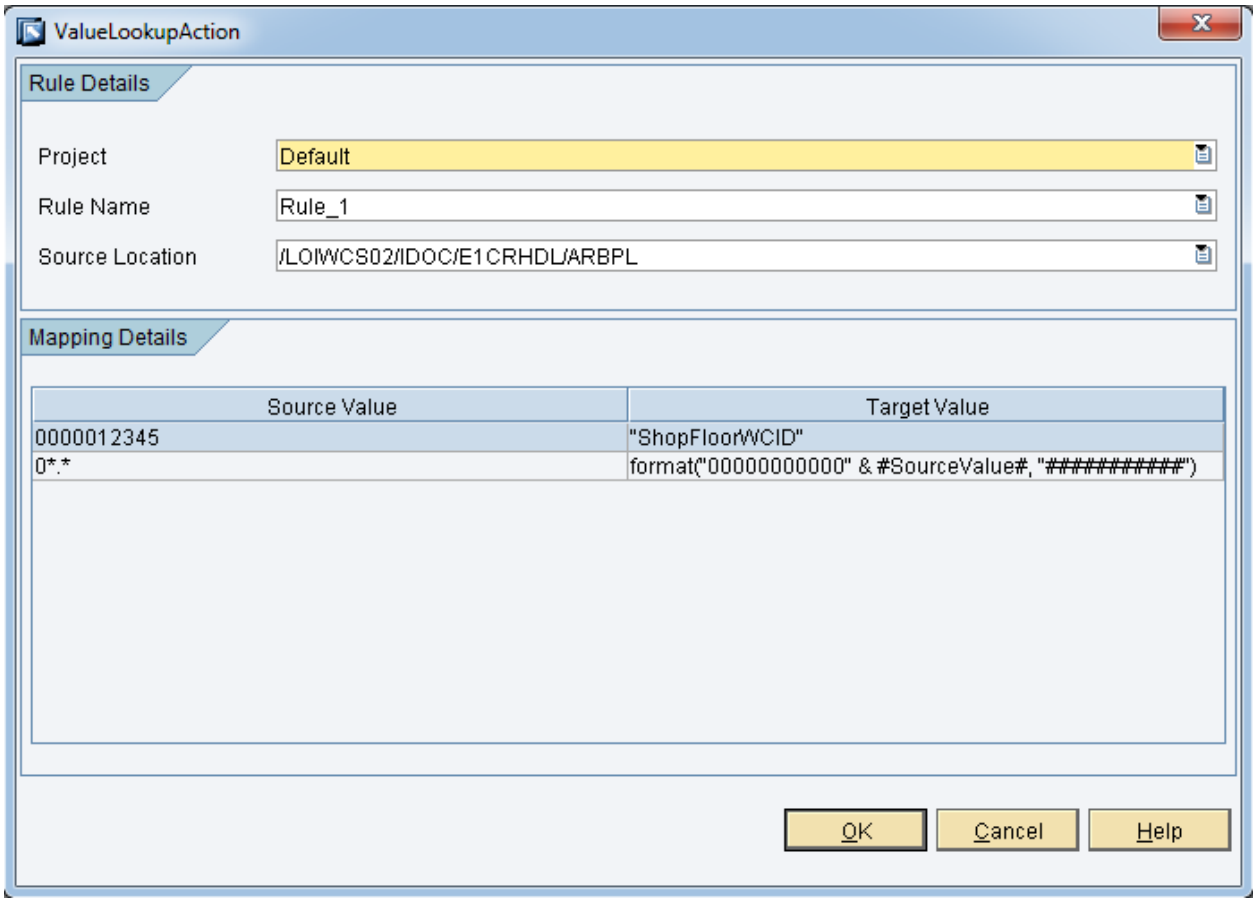



You can also map the ProjectName an RuleName fields in the link editor and pass them at runtime. You can see the effect of the Value Map rules on the Target XML as show below.



### Value Lookup Action

In case you do not have an XML but you still want to use the mappings created in the Value Map you can use the Value Lookup action. The Value Lookup action takes in a List of Source Values and returns a Map containing the Source Value as the key and the Mapped Value as the value in the Map. To configure a Value Lookup Action you will need to also choose the Source (unlike the Value Map action) as there is no XPath here to match which source to use.




In my transaction I have created a transaction property of type List containing the values 000000000010002906 and 0000012345.

The response is as below

| Key | Value |
|---|---|
| 000000000010002906 | 10002906 |
| 0000012345 | ShopFloorWCID |

## Modifying the mapping in Production Environment

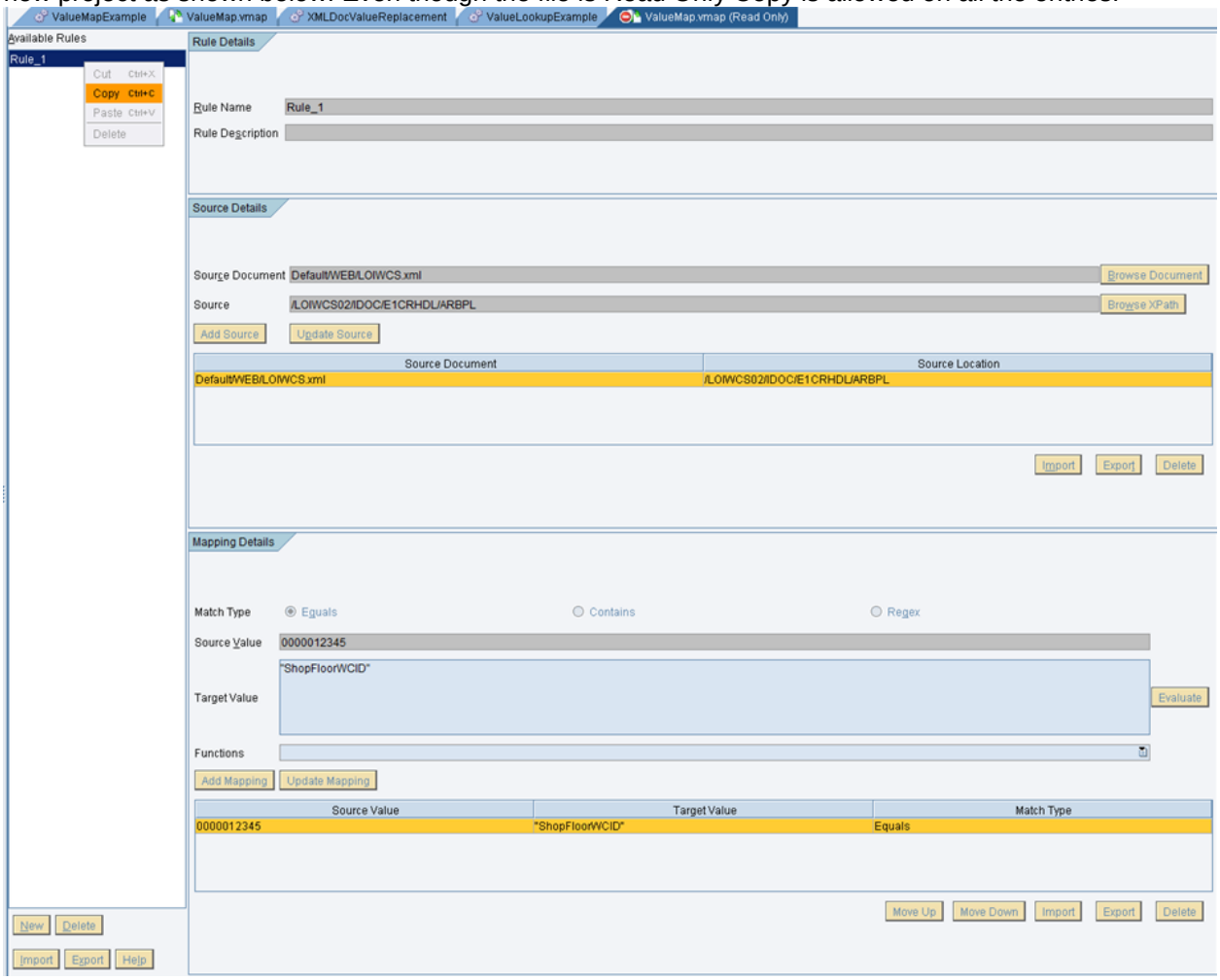
In some cases you might choose to develop the value mapping in the development environment and then deploy the MII project using NWDI to the Quality or Production environment. In this case the Value Map file cannot be modified as the file will be read only. If there is a need to modify the mappings or other areas of the Value Map (due to difference in data from Dev to Prod) then you should follow the below steps.

### In Development

1. Create the Value Map entries based on the data in the DEV environment.
2. Create the MII Transactions that have either the Value Map action or the Value Lookup action
3. Create Shared Properties for Project Name and Rule Name (For Value Lookup create a 3rd property for Source Name) and set default value as the Project Name and Rule name you want to use.
4. In the Link Editor for the Value Map or Value Lookup actions assign these properties to the Project Name and Rule Name field.
5. Test the transactions are working fine.

### In Production

1. Create a new Project in the PROD environment.
2. Copy the rules/sources/mappings you need to change and paste them in the Value Map file in the new project as shown below. Even though the file is Read Only Copy is allowed on all the entries.




3. Update the changes with new values in the mappings
4. Set the Shared Property values to the new Project Name and Rule Name. The Transactions created in DEV will now run with the updated Value Map instead of the old one.

## Related Content

The following outline will document the many available resources to resolve problems, find relevant information and simply to find additional information regarding a certain SAP component or product.

**SAP Developer Network** (http://www.sdn.sap.com)

SAP Developer Network (SDN) is an active online community where ABAP, Java, .NET, and other cutting-edge technologies converge to form a resource and collaboration channel for SAP developers, consultants, integrators, and business analysts. SDN hosts a technical library, expert blogs, exclusive downloads and code samples, an extensive eLearning catalog, and active, moderated discussion forums.

Link for MII Forum to research technical solutions and post questions in a community environment

Link for MII Wiki on SDN

Link for MII Articles on various technical topics

Link for MII sample projects and tools

**Regex Links**

Link – Regex Tutorial

Link – Regex Tutorial

Link – Documentation on Java Pattern class which tells about Regex Syntax

For more information, visit the Manufacturing homepage.

## Copyright