**Coma separator number format customized in Oracle XML Publisher**

**Number Formatting in XML Publisher**

While designing XML Publisher reports, many a times we need to play with numbers and change their formats as part of the client’s requirements. Here is a brief explanation of couple of techniques that should be used while doing any number format change in XML Publisher.  
XML Publisher supports two methods for specifying the number format:

* Oracle’s format-number function
* Microsoft Word’s Native number format mask

Now…before going into details please remember few things like:

* To use the Oracle format mask or the Microsoft format mask, the numbers in your data source (ex XML File) must be in a raw format, with no formatting applied.
* Use only one of these methods. If the number format mask is specified using both methods, the data will be formatted twice and it may cause unexpected behavior.
* If you are designing a template to be translatable, using currency in the Microsoft format mask is not recommended unless you want the data reported in the same currency for all translations. Using the MS format mask sets the currency in the template so that it cannot be updated at runtime.

**Using the Oracle Format Mask:**

To apply the Oracle format mask to a form field:  
1. Open the Form Field Options dialog box for the placeholder field.  
2. Set the Type to “Regular text”.  
3. In the Form Field Help Text field, enter the mask definition according to the following example:

**<?format-number:fieldname;’999G999D99′?>**

Where fieldname is the XML tag name of the data element you are formatting and 999G999D99 is the mask definition.

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| **0** | Digit. Each explicitly set 0 will appear, if no other number occupies the position. Format mask: 00.0000 (Ex: Change 1.234 to 01.2340) |
| **9** | Digit. Returns value with the specified number of digits with a leading space if positive or a leading minus if negative. Leading zeros are blank, except for a zero value, which returns a zero for the integer part of the fixed-point number. Format mask: 99.9999 (Ex: Change 1.234 to 1.234) |
| **C** | Returns the ISO currency symbol in the specified position. |
| **D** | Determines the placement of the decimal separator. The decimal separator symbol used will be determined at runtime based on template locale. Format mask: 9G999D99 (Ex: Change 1234.56 to 1,234.56( For English locale:) or to 1.234,56 (For German locale)) |
| **EEEE** | Returns a value in scientific notation. |
| **G** | Determines the placement of the grouping (thousands) separator. The grouping separator symbol used will be determined at runtime based on template locale. Format mask: 9G999D99 (Ex: Change 1234.56 to 1,234.56( For English locale:) or to 1.234,56 (For German locale)) |
| **L** | Returns the local currency symbol in the specified position. |
| **MI** | Displays negative value with a trailing “-”. |
| **PR** | Displays negative value enclosed by <> |
| **PT** | Displays negative value enclosed by () |
| **S (before number)** | Displays positive value with a leading “+” and negative values with a leading “-” |
| **S (after number)** | Displays positive value with a trailing “+” and negative value with a trailing “-” |

**Using the Microsoft Number Format Mask:**

To format numeric values, use Microsoft Word’s field formatting features available from the Text Form Field Options dialog box. The following graphic displays an example:  
To apply a number format to a form field:

1. Go into the Word Properties for the field.
2. Change the Type drop-down box to Number (it defaults in as Regular Text)
3. From the Number format drop-down box choose the format you want.

**Native XSL format-number function:**

You can also use the native XSL format-number function to format numbers. The native XSL format-number function takes the basic format:

**format-number(number,format,[decimalformat])**

Where:  
number (Required) – Specifies the number to be formatted.  
Format (Required)  - Specifies the format pattern. Use the following characters to specify the pattern:

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| **#** | Denotes a digit. Example: #### |
| **0** | Denotes leading and following zeros. Example: 0000.00 |
| **.** | The position of the decimal point Example: ###.## |
| **,** | The group separator for thousands. Example: ###,###.## |
| **%** | Displays the number as a percentage. Example: ##% |
| **;** | Pattern separator. The first pattern will be used forpositive numbers and the second for negative numbers |

**Currency Formatting:**

This is another feature of XML Publisher that enables you to define specific currency format masks to apply to your published data at runtime.  
To utilize currency formatting, you must:  
1. Define your currency formats in XML Publisher’s Administration interface.  
2. Assign the Currency Format Set as a configuration property at the desired level (site, data definition, or template).  
3. Enter the format-currency command in your RTF template to apply the format to the field at runtime.  
To use the format-currency command:  
In the form field dialog of the field you want to format, enter the following syntax:

**<?format-currency:ELEMENT\_NAME;’currency-format-code’?>**

For example:  
<?format-currency:BALACE\_TOTAL;’USD’?>  
The currency code must correspond to a currency format that is defined in the Currency Format Set to be used with this report. The Currency Format Set can be specified at the site level, data definition level, or template level.

# **Formatting Numbers, Dates, and Currencies**

This section provides details for formatting numbers, dates, and currencies.

It contains the following topics:

* [Formatting Numbers](https://docs.oracle.com/middleware/12211/bip/BIPRD/GUID-7BD3C665-851A-43FE-BA2D-7501390D95AC.htm#GUID-CF4D8CFE-E69F-4A8E-B1F0-48A9E4CED83F)

**Formatting Numbers**

BI Publisher supports two methods for specifying the number format.

* Oracle's format-number function (recommended).
* Microsoft Word's Native number format mask.

**Note:**

You can also use the native XSL format-number function to format numbers. For information, see [Formatting Native XSL Numbers](https://docs.oracle.com/middleware/12211/bip/BIPRD/GUID-17C05832-0E37-4D3F-AE20-D9E6AACB0E63.htm#GUID-2025E929-93D1-4231-8607-0FEADEF070FC).

Use only one of these methods. If the number format mask is specified using both methods, then the data is formatted twice, causing unexpected behavior.

The group separator and the number separator are set at runtime based on the template locale. If you are working in a locale other than en-US, or the templates require translation, use the Oracle format masks.

## **Data Source Requirements**

To use the Oracle format mask or the Microsoft format mask, the numbers in the data source must be in a raw format, with no formatting applied (for example: 1000.00). If the number has been formatted for European countries (for example: 1.000,00) then the format does not work.

The Oracle BI Publisher parser requires the Java BigDecimal string representation. This consists of an optional sign ("-") followed by a sequence of zero or more decimal digits (the integer), optionally followed by a fraction, and optionally followed by an exponent. For example: -123456.3455e-3.

## **Localization Considerations**

If you are working in a locale other than en-US, or the templates require translation, then use the Oracle format masks.

The Microsoft format masks can generate unexpected results in templates run in different locale settings.

Do not include "%" in the format mask because this fixes the location of the percent sign in the number display, while the desired position could be at the beginning or the end of a number, depending on the locale.

## **Using the Microsoft Number Format Mask**

To format numeric values, use Microsoft Word's field formatting features available from the Text Form Field Options dialog.

To apply a number format to a form field:

1. Open the Form Field Options dialog for the placeholder field.
2. Set the **Type** to Number.
3. Select the appropriate **Number format** from the list of options.

## **Supported Microsoft Format Mask Definitions**

Several format mask definitions can be used to standardize output.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Symbol 0  Location Number  Meaning:   |  |  |  | | --- | --- | --- | | Digit. Each explicitly set 0 appears, if no other number occupies the position.  Example:  Format mask: 00.0000  Data: 1.234  Display: 01.2340  Symbol #  Location Number  Meaning:  Digit. When set to #, only the incoming data is displayed.  Example:  Format mask: ##.####  Data: 1.234  Display: 1.234 |  |  | |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Note:**

Subpattern boundary: A pattern contains a positive and negative subpattern, for example, "#,##0.00;(#,##0.00)". Each subpattern has a prefix, numeric part, and suffix. The negative subpattern is optional. If absent, the positive subpattern prefixed with the localized minus sign ("-" in most locales) is used as the negative subpattern. That is, "0.00" alone is equivalent to "0.00;-0.00". If there is an explicit negative subpattern, it serves only to specify the negative prefix and suffix. The number of digits, minimal digits, and other characteristics are all the same as the positive pattern. That means that "#,##0.0#;(#)" produces precisely the same behavior as "#,##0.0#;(#,##0.0#)".

## **Using the Oracle Format Mask**

You can use the Oracle format mask in form fields.

To apply the Oracle format mask to a form field:

1. Open the Form Field Options dialog box for the placeholder field.
2. Set the **Type** to "Regular text".
3. In the Form Field Help Text field, enter the mask definition according to the following example:

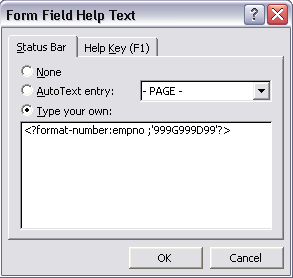
<?format-number:*fieldname*;'*999G999D99*'?>

where

*fieldname* is the XML tag name of the data element you are formatting and

*999G999D99* is the mask definition.

The following illustration shows an example Form Field Help Text dialog entry for the data element "empno".

  
[Description of the illustration GUID-06367FBE-640A-4A71-B275-9677D7D2DA20-default.gif](https://docs.oracle.com/middleware/12211/bip/BIPRD/img_text/GUID-06367FBE-640A-4A71-B275-9677D7D2DA20-default.htm)

The following table lists the supported Oracle number format mask symbols and their definitions.

| **Symbol** | **Meaning** |
| --- | --- |
| 0 | Digit. Each explicitly set 0 appears, if no other number occupies the position. Example: Format mask: 00.0000 Data: 1.234 Display: 01.2340 |
| 9 | Digit. Returns value with the specified number of digits with a leading space if positive or a leading minus if negative. Leading zeros are blank, except for a zero value, which returns a zero for the integer part of the fixed-point number. Example: Format mask: 99.9999 Data: 1.234 Display: 1.234 |
| C | Returns the ISO currency symbol in the specified position. |
| D | Determines the placement of the decimal separator. The decimal separator symbol used is determined at runtime based on template locale. For example: Format mask: 9G999D99 Data: 1234.56 Display for English locale: 1,234.56 Display for German locale: 1.234,56 |
| EEEE | Returns a value in scientific notation. |
| G | Determines the placement of the grouping (thousands) separator. The grouping separator symbol used is determined at runtime based on template locale. For example: Format mask: 9G999D99 Data: 1234.56 Display for English locale: 1,234.56 Display for German locale: 1.234,56 |
| L | Returns the local currency symbol in the specified position. |
| MI | Displays negative value with a trailing "-". |
| PR | Displays negative value enclosed by <> |
| PT | Displays negative value enclosed by () |
| S (before number) | Displays positive value with a leading "+" and negative values with a leading "-" |
| S (after number) | Displays positive value with a trailing "+" and negative value with a trailing "-" |