

# iDempiere

## Apply currency precision for display Customization Reference



- OverView
- (Step1)Basic modification
- (Step2)Apply currency precision For Web-UI
- (Step3)Apply currency precision For Swing-UI
- (Step4)Apply currency precision For Reports

## (Fundamental Rule) Apply "Currency Precision" for display, if Format Pattern is Null.

- In spite of Display type is "Amount" or "Costs+Prices", Display currency precision of iDempiere apply the decimal point second place. I think this specification need to improve for currency that is not the decimal point second place.
- I think best that if "Format Pattern" of "Amount" and "Costs+Prices" are null, apply "Currency Precision" for display.
  - \*If "Format Pattern" is not null, Apply "Format Pattern" for display basically.
- "Web-UI" and "Swing-UI" refer to "Format Pattern" at table of AD\_Column. If "Format Pattern" at table of AD\_Column is null and "Display Type" is "Amount" or "Costs+Prices", apply "Currency Precision" for Display.
- "Reports" refer to "Format Pattern" at table of AD\_PrintFormatItem at first. If "Format Pattern" at table of AD\_PrintFormatItem is null, "Reports" refer to "Format Pattern" at table of AD\_Column at second. If "Format Pattern" at table of AD\_Column is null, apply "Currency Precision" for Display thirdly.

## I think that this rule keeps consistency to display of currency in iDempiere.

### Example of Web-UI and Swing-UI

The screenshot shows a 'Sales Order for Demo' form in the iDempiere Web-UI. The 'Currency' field is set to 'JPY'. The 'Grand Total' field displays '52,500.0000'. A technical details panel on the right shows the 'Format Pattern' for the 'Grand Total' field as '###,###,###.0000'. Red boxes and arrows highlight these elements to illustrate the application of currency precision when the format pattern is null.

"Format Pattern" of AD\_Column is null.  
Apply "Currency precision" for display.  
Currency precision of JPY is 0.

Apply "Format Pattern"

## Example of Reports

Report: 受発注明細レポート SuperUse...  
 Report: 受発注明細レポート2 SuperUser@株式会社 OpenBiz 本社 [Hagiwara-PC{localhost-idempiere-adempiere}]

PDF 受発注明細レポート2 Summary

Page 8 of 8

Order	Business Partner	Warehouse	Currency	Total Lines	Grand Total
POO-1000112 ✗	仕入先A	倉庫A	JPY	400.0000	420
POO-1000113 ✗	仕入先A	倉庫A	JPY	480.0000	504
SOO-1000066 ✓	得意先A	倉庫A	JPY	900.0000	900
POO-1000115 ✗	仕入先A	倉庫A	JPY	480.0000	504
POO-1000116 ✗	仕入先A	倉庫A	JPY	480.0000	504
SOO-1000064 ✓	得意先		JPY	900.0000	945
SOO-1000065 ✓	得意先		USD	360.0000	360.00

Apply "Format Pattern" of AD\_PrintFormatItem

"Format Pattern" of AD\_PrintFormatItem is null, and "Format Pattern" of AD\_Column is null.  
 Apply "Currency precision" for display.  
 Currency precision of JPY is 0.  
 Currency precision of USD is 2.

Print Format > Format Item

Client\* 株式会社 OpenBiz Organization\* 本社

Print Format\* 受発注明細レポート2

Sequence\* 60

Name\* Total Lines

Print Text Total Lines

Active  Centrally maintained

Suppress Null  Suppress Repeats

Format Type\* Field Column\* TotalLines\_Total Lines

Format Pattern ###,###,###.0000

Barcode Type

## (Step1)Basic modification : Add New Method to MCurrency Class

- Add New Method named “getCostingPrecision()” to MCurrency Class.

\*Please refer to “public static int getStdPrecision (Properties ctx, int C\_Currency\_ID)” method that same function.

```
MCurrency.java x
160
161 > /**
162 > * > Get Standard Precision.
163 > * @param ctx Context
164 > * @param C_Currency_ID currency
165 > * @return Standard Precision
166 > */
167 > public static int getStdPrecision (Properties ctx, int C_Currency_ID)
168 > {
169 > > MCurrency c = get(ctx, C_Currency_ID);
170 > > return c.getStdPrecision();
171 > } // getStdPrecision
172
173
174 > /**
175 > * Get Costing Precision.
176 > * @param ctx Context
177 > * @param C_Currency_ID currency
178 > * @return Costing Precision
179 > */
180 > public static int getCostingPrecision (Properties ctx, int C_Currency_ID)
181 > {
182 > > MCurrency c = get(ctx, C_Currency_ID);
183 > > return c.getCostingPrecision();
184 > } //getCostingPrecision
185
186
```

Add this method to MCurrency class to apply currency precision(**CostingPrecision**) for display

- For Copy

```
/**
 * Get Costing Precision.
 * @param ctx Context
 * @param C_Currency_ID currency
 * @return Costing Precision
 */
public static int getCostingPrecision (Properties ctx, int C_Currency_ID)
{
    MCurrency c = get(ctx, C_Currency_ID);
    return c.getCostingPrecision();
} //getCostingPrecision
```

# (Step1)Basic modification : Modify DisplayType Class

- Modify “public static DecimalFormat getNumberFormat(int displayType, Language language, String pattern)” method
- Add “public static DecimalFormat getNumberFormat(int displayType, Language language, String pattern, int C\_Currency\_ID)” method

```
import org.compiere.model.MCurrency;
```

```
/*Return Format for numeric DisplayType
 * @param displayType Display Type (default Number)
 * @param language Language
 * @param pattern Java Number Format pattern e.g. "#,##0.00"
 * @param C_Currency_ID int
 * @return number format
 */
public static DecimalFormat getNumberFormat(int displayType, Language language, String pattern, int C_Currency_ID)
{
    Language myLanguage = language;
    if (myLanguage == null)
        myLanguage = Language.getLoginLanguage();
    Locale locale = myLanguage.getLocale();
    DecimalFormat format = null;
    if (locale != null)
        format = (DecimalFormat)NumberFormat.getNumberInstance(locale);
    else
        format = (DecimalFormat)NumberFormat.getNumberInstance(Locale.US);
    //
    if (pattern != null && pattern.length() > 0)
    {
        try {
            format.applyPattern(pattern);
            return format;
        }
        catch (IllegalArgumentException e) {
            s_log.log(Level.WARNING, "Invalid number format: " + pattern);
        }
    }
    else if (displayType == Integer)
    {
        format.setParseIntegerOnly(true);
        format.setMaximumIntegerDigits(INTEGER_DIGITS);
        format.setMaximumFractionDigits(0);
    }
    else if (displayType == Quantity)
    {
        format.setMaximumIntegerDigits(MAX_DIGITS);
        format.setMaximumFractionDigits(MAX_FRACTION);
    }
    else if (displayType == Amount)
    {
        format.setMaximumIntegerDigits(MAX_DIGITS);
        format.setMaximumFractionDigits(MAX_FRACTION);
        if(C_Currency_ID==0)
        {
            format.setMinimumFractionDigits(AMOUNT_FRACTION);
        }else{
            format.setMinimumFractionDigits(MCurrency.getStdPrecision(Env.getCtx(), C_Currency_ID));
        }
    }
    else if (displayType == CostPrice)
    {
        format.setMaximumIntegerDigits(MAX_DIGITS);
        format.setMaximumFractionDigits(MAX_FRACTION);
        if(C_Currency_ID==0)
        {
            format.setMinimumFractionDigits(AMOUNT_FRACTION);
        }else{
            format.setMinimumFractionDigits(MCurrency.getCostingPrecision(Env.getCtx(), C_Currency_ID));
        }
    }
    else //if (displayType == Number)
    {
        format.setMaximumIntegerDigits(MAX_DIGITS);
        format.setMaximumFractionDigits(MAX_FRACTION);
        format.setMinimumFractionDigits(1);
    }
    return format;
} //getDecimalFormat
```

Add logic to apply currency precision for display

Add logic to apply currency precision for display

- Copy “public static DecimalFormat getNumberFormat(int displayType, Language language, String pattern)” method. And Create “public static DecimalFormat getNumberFormat(int displayType, Language language, String pattern, int C\_Currency\_ID)” method.

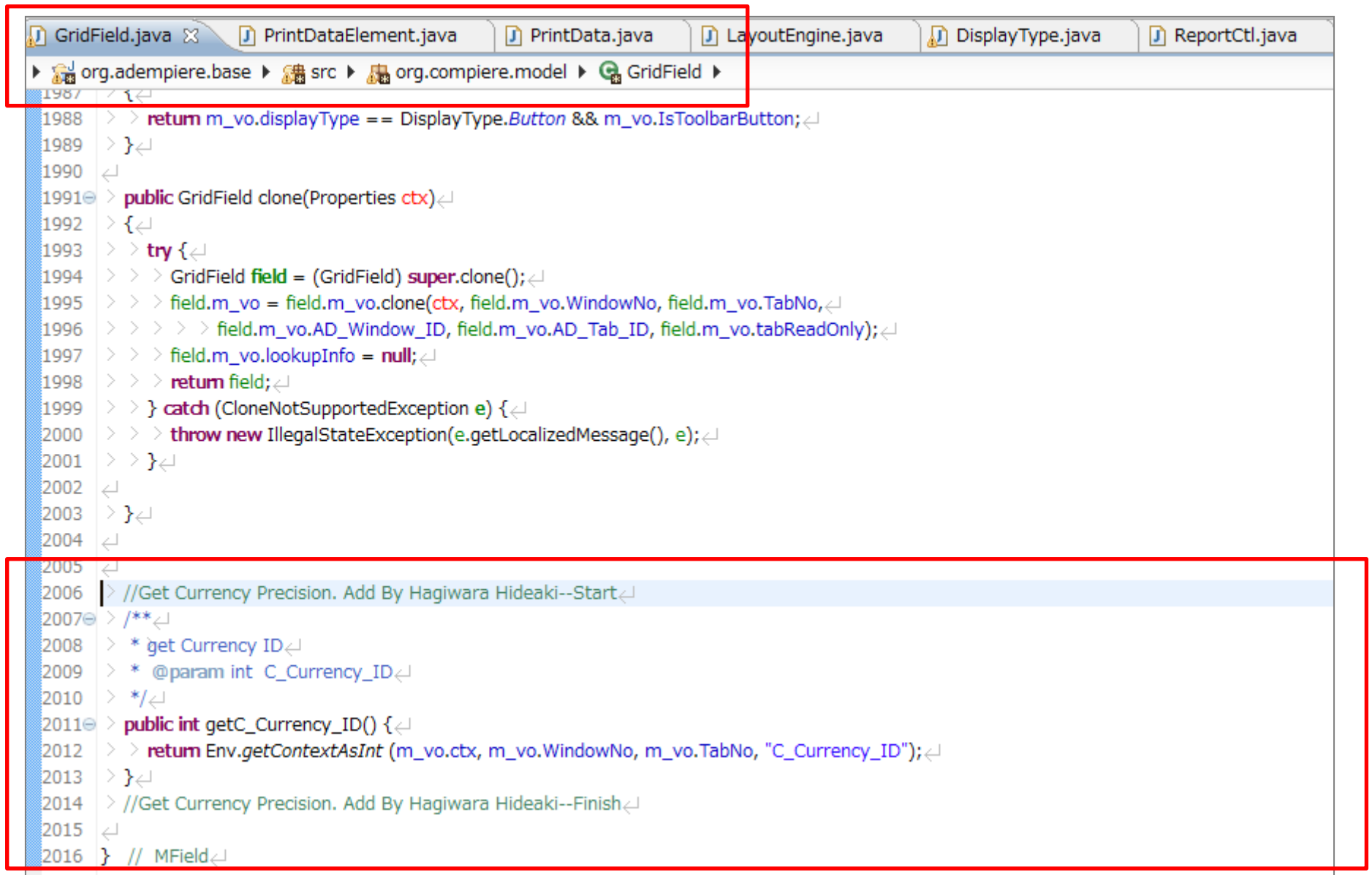
- Control to display of after the decimal point When Display Type is “Amount” and “CostPrice”, by using the C\_Currency\_ID of “StdPrecision” and “CostingPrecision”

```
/*Return Format for numeric DisplayType
 * @param displayType Display Type (default Number)
 * @param language Language
 * @param pattern Java Number Format pattern e.g. "#,##0.00"
 * @return number format
 */
public static DecimalFormat getNumberFormat(int displayType, Language language, String pattern)
{
    return getNumberFormat(displayType, language, pattern, 0);
}
```

- Modify “public static DecimalFormat getNumberFormat(int displayType, Language language, String pattern)” method.

## (Step1)Basic modification : Add New Method to GridField Class

- Add new method `getC_Currency_ID()` to GridField Class



The screenshot shows an IDE window with several tabs: GridField.java, PrintDataElement.java, PrintData.java, LayoutEngine.java, DisplayType.java, and ReportCtl.java. The GridField.java tab is active, showing the following code:

```
1987 < < <
1988 > > return m_vo.displayType == DisplayType.Button && m_vo.IsToolBarButton;
1989 > }
1990 <
1991 public GridField clone(Properties ctx)
1992 > {
1993 > > try {
1994 > > > GridField field = (GridField) super.clone();
1995 > > > field.m_vo = field.m_vo.clone(ctx, field.m_vo.WindowNo, field.m_vo.TabNo,
1996 > > > > field.m_vo.AD_Window_ID, field.m_vo.AD_Tab_ID, field.m_vo.tabReadOnly);
1997 > > > field.m_vo.lookupInfo = null;
1998 > > > return field;
1999 > > } catch (CloneNotSupportedException e) {
2000 > > > throw new IllegalStateException(e.getLocalizedName(), e);
2001 > > }
2002 <
2003 > }
2004 <
2005 <
2006 > //Get Currency Precision. Add By Hagiwara Hideaki--Start
2007 > /**
2008 > * get Currency ID
2009 > * @param int C_Currency_ID
2010 > */
2011 public int getC_Currency_ID() {
2012 > > return Env.getContextAsInt (m_vo.ctx, m_vo.WindowNo, m_vo.TabNo, "C_Currency_ID");
2013 > }
2014 > //Get Currency Precision. Add By Hagiwara Hideaki--Finish
2015 <
2016 } // MField
```

- For Copy

```
//Get Currency ID
/**
 *get Currency ID
 * @param int C_Currency_ID
 */
public int getC_Currency_ID() {
    return Env.getContextAsInt (m_vo.ctx, m_vo.WindowNo, m_vo.TabNo, "C_Currency_ID");
}
```

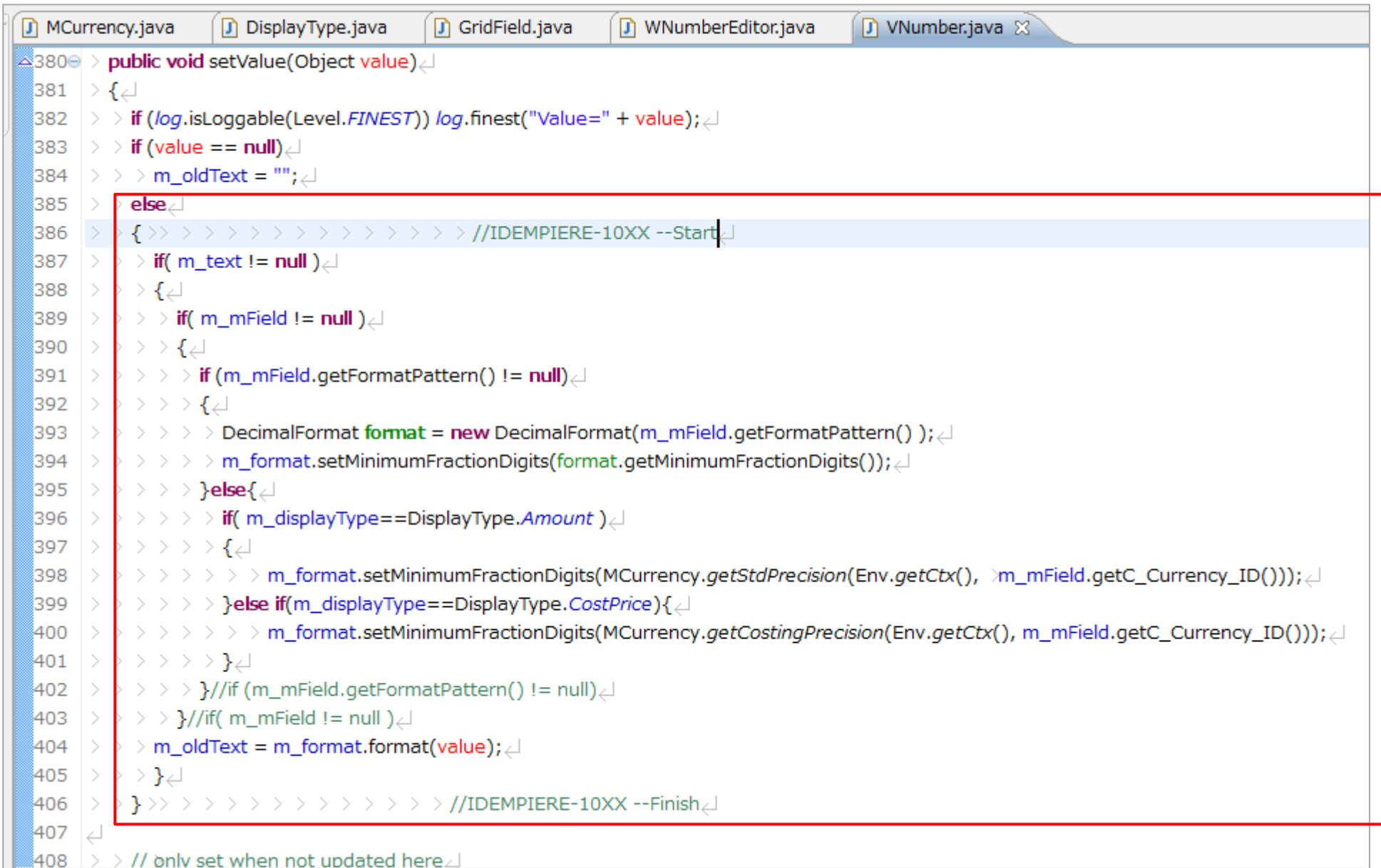




## (Step3)Apply currency precision For Swing-UI : Modify VNumber Class

### ➤ Modify setValue() Method of VNumber Class

```
import org.compiere.model.MCurrency;
```



```
380 > public void setValue(Object value) {
381 > {
382 > > if (log.isLoggable(Level.FINEST)) log.finest("Value=" + value);
383 > > if (value == null)
384 > > > m_oldText = "";
385 > > else
386 > > > { //IDEMPIERE-10XX --Start
387 > > > > if (m_text != null)
388 > > > > {
389 > > > > > if (m_mField != null)
390 > > > > > {
391 > > > > > > if (m_mField.getFormatPattern() != null)
392 > > > > > > {
393 > > > > > > > DecimalFormat format = new DecimalFormat(m_mField.getFormatPattern());
394 > > > > > > > m_format.setMinimumFractionDigits(format.getMinimumFractionDigits());
395 > > > > > > }else{
396 > > > > > > > if (m_displayType==DisplayType.Amount)
397 > > > > > > > {
398 > > > > > > > > m_format.setMinimumFractionDigits(MCurrency.getStdPrecision(Env.getCtx(), m_mField.getC_Currency_ID()));
399 > > > > > > > }else if(m_displayType==DisplayType.CostPrice){
400 > > > > > > > > m_format.setMinimumFractionDigits(MCurrency.getCostingPrecision(Env.getCtx(), m_mField.getC_Currency_ID()));
401 > > > > > > > }
402 > > > > > > } //if (m_mField.getFormatPattern() != null)
403 > > > > > > } //if (m_mField != null)
404 > > > > > > m_oldText = m_format.format(value);
405 > > > > > > }
406 > > > > > > } //IDEMPIERE-10XX --Finish
407 > >
408 > > // only set when not updated here
```

### ➤ For Copy

```
{ //IDEMPIERE-10XX --Start
if (m_text != null)
{
if (m_mField != null)
{
if (m_mField.getFormatPattern() != null)
{
DecimalFormat format = new DecimalFormat(m_mField.getFormatPattern());
m_format.setMinimumFractionDigits(format.getMinimumFractionDigits());
}else{
if (m_displayType==DisplayType.Amount)
{
m_format.setMinimumFractionDigits(MCurrency.getStdPrecision(Env.getCtx(), m_mField.getC_Currency_ID()));
}else if(m_displayType==DisplayType.CostPrice){
m_format.setMinimumFractionDigits(MCurrency.getCostingPrecision(Env.getCtx(), m_mField.getC_Currency_ID()));
}
} //if (m_mField.getFormatPattern() != null)
} //if (m_mField != null)
m_oldText = m_format.format(value);
}
} //IDEMPIERE-10XX --Finish
```



## (Step4)Apply currency precision For Reports : Add New Method to PrintData Class

Add this method to PrintData Class to apply currency precision for display

```
/******  
*get C_Currency_ID of current row  
*  
* @return int C_Currency_ID or 0  
*/  
public int getC_Currency_ID()  
{  
    int indexOfCurrency = this.getIndex("C_Currency_ID");  
    int C_Currency_ID = 0;  
  
    if( indexOfCurrency >= 0 ) // when currencyIndex < 0 , PrintData instance doesn't have PrintDataElement of C_Currency_ID  
    {  
        Object obj = m_matrix.getRowData().get(indexOfCurrency);  
        if( obj instanceof PrintDataElement)  
        {  
            PrintDataElement data = (PrintDataElement)obj;  
            String value = data.getValueKey();  
            try  
            {  
                C_Currency_ID = Integer.parseInt(value);  
            }catch (Exception e){  
                log.info("Value="+value.toString());  
            }  
        }  
    }  
    return C_Currency_ID;  
}
```

## (Step4)Apply currency precision For Reports : Modify PrintDataElement Class

```
/**  
 * Get Node Value Display  
 * @param language optional language - if null numbers/dates are not formatted  
 * @return display value optionally formatted  
 */
```

```
public String getValueDisplay (Language language)  
{  
    return getValueDisplay (language, 0);  
}
```

Modify "public String getValueDisplay (Language language)" method.

```
public String getValueDisplay (Language language, int C_Currency_ID)
```

```
{  
    if (m_value == null)  
        return "";  
    String retValue = m_value.toString();  
    if (m_displayType == DisplayType.Location)  
        return getValueDisplay_Location();  
    // ID columns should be printed as ID numbers - teo_sarca [ 1673363 ]  
    else if (DisplayType.ID == m_displayType && m_value instanceof KeyNamePair)  
        return ((KeyNamePair)m_value).getID();  
    else if (m_columnName.equals("C_BPartner_Location_ID") || m_columnName.equals("Bill_Location_ID"))  
        return getValueDisplay_BPLocation();  
    else if (m_displayType == 0 || m_value instanceof String || m_value instanceof NamePair)  
        ;  
    else if (language != null)//Optional formatting of Numbers and Dates  
    {  
        if (DisplayType.isNumeric(m_displayType))  
            retValue = DisplayType.getNumberFormat(m_displayType, language, m_formatPattern, C_Currency_ID).format(m_value);  
        else if (DisplayType.isDate(m_displayType))  
            retValue = DisplayType.getDateFormat(m_displayType, language, m_formatPattern).format(m_value);  
    }  
    return retValue;  
} //getValueDisplay
```

Add logic to apply currency precision for display

# (Step4)Apply currency precision For Reports : Modify LayoutEngine Class(1)

## ➤ Modify “private PrintElement createFieldElement()” Method

```
LayoutEngine.java | MCurrency.java | DisplayType.java | GridField.java | WNumberEditor.java | VNumber.java | PrintData.java
org.adempiere.base | src | org.compiere.print.layout | LayoutEngine | createFieldElement(MPrintFormatItem, int, String, boolean) : PrintElement
1333 > private PrintElement createFieldElement (MPrintFormatItem item, int maxWidth,
1334 > > String FieldAlignmentType, boolean isForm)
1335 > {
1336 > > // Get Data
1337 > > Object obj = m_data.getNode(new Integer(item.getAD_Column_ID()));
1338 > > if (obj == null)
1339 > > > return null;
1340 > > else if (obj instanceof PrintDataElement)
1341 > > > ;
1342 > > else
1343 > > {
1344 > > > log.log(Level.SEVERE, "Element not PrintDataElement " + obj.getClass());
1345 > > > return null;
1346 > > }
1347 <
1348 > > // Convert DataElement to String
1349 > > PrintDataElement data = (PrintDataElement)obj;
1350 > > if (data.isNull() && item.isSuppressNull())
1351 > > > return null;
1352 <
1353 > > //IDEMPIERE-10XX -- Start
1354 > > String stringContent = null;
1355 > > if(data.getDisplayType()==DisplayType.Amount || data.getDisplayType()==DisplayType.CostPrice)
1356 > > > stringContent = data.getValueDisplay (m_format.getLanguage(), m_data.getC_Currency_ID());
1357 > > else
1358 > > > stringContent = data.getValueDisplay (m_format.getLanguage());
1359 > > //IDEMPIERE-10XX -- Finish
1360 <
```

Add logic to apply currency precision for display

## ➤ For Copy

```
//IDEMPIERE-10XX -- Start
String stringContent = null;
if(data.getDisplayType()==DisplayType.Amount || data.getDisplayType()==DisplayType.CostPrice)
    stringContent = data.getValueDisplay (m_format.getLanguage(), m_data.getC_Currency_ID());
else
    stringContent = data.getValueDisplay (m_format.getLanguage());
//IDEMPIERE-10XX -- Finish
```

## (Step4)Apply currency precision For Reports : Modify LayoutEngine Class(2)

### ➤ Modify “private PrintElement layoutTable ()” Method

```
LayoutEngine.java | MCurrency.java | DisplayType.java | GridField.java | WNumberEditor.java | VNumber.java | PrintData.java
org.adempiere.base | src | org.compiere.print.layout | LayoutEngine | layoutTable(MPrintFormat, PrintData, int) : PrintElement
1718 >>>>>>> if (item.getAD_Column_ID() > 0) // teo_sarca, [ 1073342 ]
1719 >>>>>>> obj = printData.getNode(new Integer(item.getAD_Column_ID()));
1720 >>>>>>> if (obj == null)
1721 >>>>>>> ;
1722 >>>>>>> else if (obj instanceof PrintDataElement)
1723 >>>>>>> {
1724 >>>>>>>> PrintDataElement pde = (PrintDataElement)obj;
1725 >>>>>>>> if (pde.isID() || pde.isYesNo())
1726 >>>>>>>>> dataElement = (Serializable) pde.getValue();
1727 >>>>>>>>> else
1728 >>>>>>>>> { //IDEMPIERE-10XX --Start
1729 >>>>>>>>>> if(pde.getDisplayType()==DisplayType.Amount || pde.getDisplayType()==DisplayType.CostPrice)
1730 >>>>>>>>>> {
1731 >>>>>>>>>>>> int C_Currency_ID = printData.getC_Currency_ID();
1732 >>>>>>>>>>>> if(C_Currency_ID==0 && m_data != null)
1733 >>>>>>>>>>>>> dataElement = pde.getValueDisplay(format.getLanguage(), m_data.getC_Currency_ID()); //use currency of parent Data.
1734 >>>>>>>>>>>>> else
1735 >>>>>>>>>>>>>> dataElement = pde.getValueDisplay(format.getLanguage(), C_Currency_ID);
1736 >>>>>>>>>>>>>> }else{
1737 >>>>>>>>>>>>>>> dataElement = pde.getValueDisplay(format.getLanguage());
1738 >>>>>>>>>>>>>>> }
1739 >>>>>>>>>>>>> } //IDEMPIERE-10XX --finish
1740 >>>>>>>>>>>>> }
1741 >>>>>>>>>>>>> else
1742 >>>>>>>>>>>>>> log.log(Level.SEVERE, "Element not PrintDataElement " + obj.getClass());
1743 >>>>>>>>>>>>>> columnElement = dataElement;
1744 >>>>>>>>>>>>> }
```

Add logic to apply currency precision for display

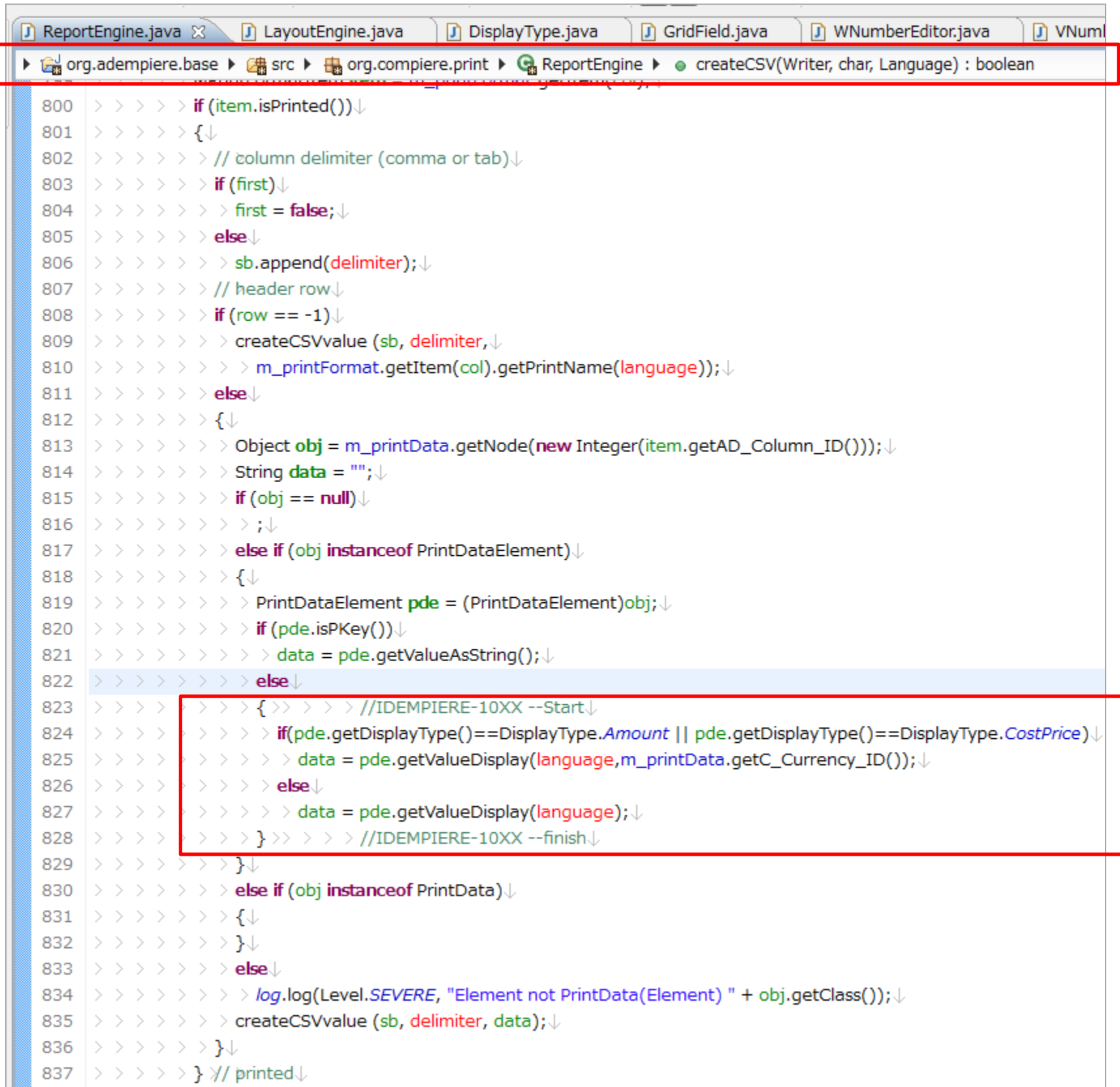
### ➤ For Copy

```
{ //IDEMPIERE-10XX --Start
if(pde.getDisplayType()==DisplayType.Amount || pde.getDisplayType()==DisplayType.CostPrice)
{
int C_Currency_ID = printData.getC_Currency_ID();
if(C_Currency_ID==0 && m_data != null)
dataElement = pde.getValueDisplay(format.getLanguage(), m_data.getC_Currency_ID()); //use currency of parent Data.
else
dataElement = pde.getValueDisplay(format.getLanguage(), C_Currency_ID);
}else{
dataElement = pde.getValueDisplay(format.getLanguage());
}
} //IDEMPIERE-10XX --finish
```



## (Step4)Apply currency precision For Reports : Modify ReportEngine Class(2)

### ➤ Modify "createCSV()" Method



```
ReportEngine.java | LayoutEngine.java | DisplayType.java | GridField.java | WNumberEditor.java | VNum
org.adempiere.base > src > org.compiere.print > ReportEngine > createCSV(Writer, char, Language) : boolean
800 >>>> if (item.isPrinted())↓
801 >>>> {↓
802 >>>> > // column delimiter (comma or tab)↓
803 >>>> > if (first)↓
804 >>>> > > first = false;↓
805 >>>> > > else↓
806 >>>> > > > sb.append(delimiter);↓
807 >>>> > > > // header row↓
808 >>>> > > > if (row == -1)↓
809 >>>> > > > > createCSVvalue (sb, delimiter,↓
810 >>>> > > > > > m_printFormat.getItem(col).getPrintName(language));↓
811 >>>> > > > > > else↓
812 >>>> > > > > > {↓
813 >>>> > > > > > > Object obj = m_printData.getNode(new Integer(item.getAD_Column_ID()));↓
814 >>>> > > > > > > String data = "";↓
815 >>>> > > > > > > if (obj == null)↓
816 >>>> > > > > > > > ;↓
817 >>>> > > > > > > > else if (obj instanceof PrintDataElement)↓
818 >>>> > > > > > > > {↓
819 >>>> > > > > > > > > PrintDataElement pde = (PrintDataElement)obj;↓
820 >>>> > > > > > > > > if (pde.isPKey())↓
821 >>>> > > > > > > > > > data = pde.getValueAsString();↓
822 >>>> > > > > > > > > > else↓
823 >>>> > > > > > > > > > { >>>> //IDEMPIERE-10XX --Start↓
824 >>>> > > > > > > > > > > if(pde.getDisplayType()==DisplayType.Amount || pde.getDisplayType()==DisplayType.CostPrice)↓
825 >>>> > > > > > > > > > > > data = pde.getValueDisplay(language,m_printData.getC_Currency_ID());↓
826 >>>> > > > > > > > > > > > else↓
827 >>>> > > > > > > > > > > > > data = pde.getValueDisplay(language);↓
828 >>>> > > > > > > > > > > > } >>>> //IDEMPIERE-10XX --finish↓
829 >>>> > > > > > > > > > }↓
830 >>>> > > > > > > > > > > else if (obj instanceof PrintData)↓
831 >>>> > >>>> {↓
832 >>>> > >>>> }↓
833 >>>> > >>>> else↓
834 >>>> > >>>> > > log.log(Level.SEVERE, "Element not PrintData(Element) " + obj.getClass());↓
835 >>>> > >>>> > > createCSVvalue (sb, delimiter, data);↓
836 >>>> > >>>> }↓
837 >>>> } // printed↓
```

### ➤ For Copy

```
{
    //IDEMPIERE-10XX --Start
    if(pde.getDisplayType()==DisplayType.Amount || pde.getDisplayType()==DisplayType.CostPrice)
        data = pde.getValueDisplay(language,m_printData.getC_Currency_ID());
    else
        data = pde.getValueDisplay(language);
}
//IDEMPIERE-10XX --finish
```