

# Callout

## From ADempiere ERP Wiki

← Table of Contents | **Callout** | Callout Code ⇒

### Contents

- 1 What is a Callout?
- 2 How to create a callout
  - 2.1 Steps to create a callout
  - 2.2 What is available inside Callout
  - 2.3 How to get existing value
  - 2.4 How to change value
  - 2.5 See also
  - 2.6 Callout en Español
- 3 See Also

## What is a Callout?

Callout is a coding mechanism to modify the content of a display field in a Window. This display field need to be attached to a Column in AD Table. You can write your logic in Java Classes (need a server restart in this case), or you use a JSR223 Script Callout.

In Table and Column window, under Column tab, you will find the Callout field which will call the Callout class whenever the field undergoes input activity. Callout can change the values of other fields in the same Window in scope or in use at that time.

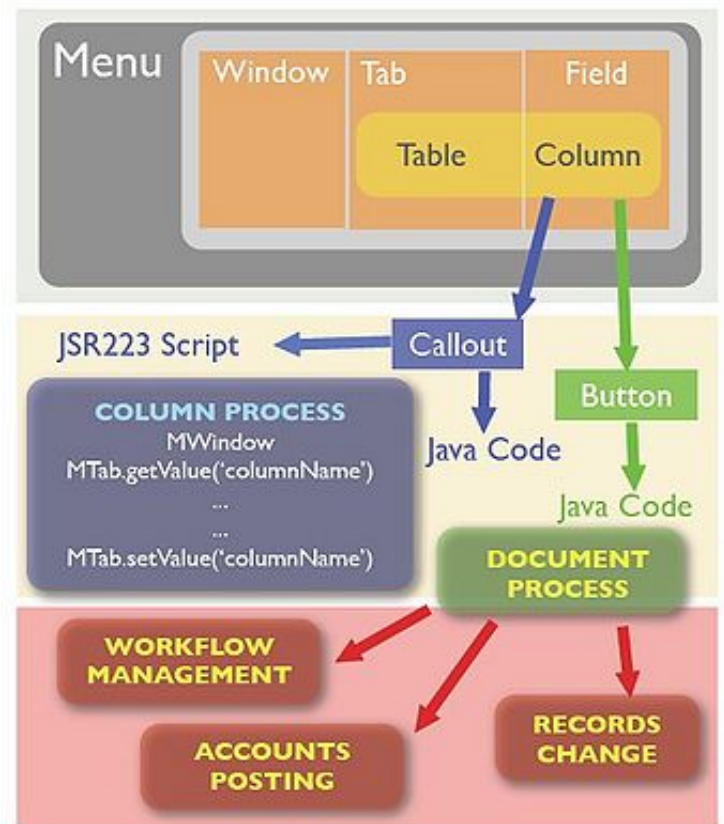
A Callout is a java method which gets executed when the field in an ADempiere window tab gets focus and onChange activity. A Callout class (extending CalloutEngine) will group such methods into a single file, usually under a document name i.e. CalloutOrder.java. A Callout is defined in the Table & Column window, under a column tab, where you may specify a list of fully qualified methods (separated by ";").

## How to create a callout

### Steps to create a callout

To write a callout you have to do the following things:

1. Write the callout function



```

package org.adempiere.callout;

import java.util.Properties;

import org.compiere.model.CalloutEngine;
import org.compiere.model.GridField;
import org.compiere.model.GridTab;
import org.compiere.util.AdempiereSystemError;
import org.compiere.util.Env;

public class SimpleCallout extends CalloutEngine {

    public String test(Properties ctx, int windowNo, GridTab mTab, GridField mField,
        Object value) throws AdempiereSystemError {

        // First get value
        Price = ((BigDecimal)mTab.getValue("Price"));

        // do some calculations
        BigDecimal Total = Qty.multiply(Price);

        // Set value back to column field
        mTab.setValue("Total", Total);

        return "this is a return string";
    }
}

```

The full qualified name of the method is `org.adempiere.callout.SimpleCallout.test` - you will need this in the next step.

2. Login as system administrator and open the 'Table&Column' window. Navigate to the table and column to which you want to add your callout by typing the full qualified method name into the 'Callout' field. (You can add more than one callout if you separate them with ';')
3. Create a jar file with your callout classes and rename it to `customization.jar`. Copy this `customization.jar` into the `Adempiere/lib` folder of your ADempiere installation and rerun the `RUN_setup.bat/sh`. Restart the ADempiere server and install the new client (or restart the client with Java WebStart).
4. Test your callout by navigating to the field where you added the callout and change it's value.

## What is available inside Callout

You can have callout functions with 5 parameters (like in the example above) or 6 parameters. The last parameter is the old value.

```

public String callout (Properties ctx, String method, int WindowNo,
    GridTab mTab, GridField mField, Object value, Object oldValue);

```

the `WindowNo` will inform the system which window is referred to. This we can understand as when the callout happens, we were in a window screen. So when the callout finishes its job, the result update will appear in the same window. `MTab` concerns the Tab (that is linked to a table & field context) that is in focus. You can hover your mouse pointer over any word and see the highlights. If you press the Ctrl key while you hover over them, and click, you may really dial in – to the class that handles the objects.

## How to get existing value

This is easily achieved as shown in the example:

```
Qty = (BigDecimal)mTab.getValue("Qty");
Price = ((BigDecimal)mTab.getValue("Price"));
BigDecimal Total = Qty.multiply(Price);
```

The getValue pattern basically obtain the value from the Window field in scope.

## How to change value

The setValue will then place a new value into the Window Field. The Total field changes as you put in a new value into either Qty or Price.

```
mTab.setValue("Total", Total);
```

## See also

- Script\_Callout
- The old famous Callout.pdf (<http://compiere.red1.org/Callout.zip>) .

## Callout en Español

Crear un callout sencillo:

Por Victor Cappugi. Venezuela

1ero. Que es un callout. Un callout en adempiere no es mas que una forma de alterar los valores de los campos en las ventanas por medio de cálculos directos o asignaciones, como ejemplo realizar la suma de dos campos. Se advierte que no se use Callout como métodos de validación de Datos, ya que ésto se hace por el diccionario. Consiste en un método que se ejecuta en adempiere cuando un campo de una ventana es modificado

Forma de Generar un callout.

1 se debe escribir en Java con su IDE favorito, la estructura del Callout. Las partes básicas serán.

```
package org.adempiere.callout; //un paquete que puede ser creado o incluir el callout en org.compiere.m
import java.util.Properties;
import org.compiere.model.CalloutEngine; //notese que pertenecen al paquete
import org.compiere.model.GridField; // org.compiere.model, si crea el callout en este
import org.compiere.model.GridTab; // paquete, estos import no son necesarios.
import org.compiere.util.AdempiereSystemError;
```

```
public class CalloutXX extends CalloutEngine () { //extension de la clase CalloutEngine el nombre Callout
nombre de su callout (siga los estandares)
public void metodo1 ( Properties ctx, int WindowNo,GridTab mTab,GridField mField,Object value)
{-----
-----}
}
```

### Funcionalidad:

- Use `getValue()` y `setValue()` para obtener y colocar valores de los campos
- Ejecute consultas de sql con la siguiente estructura:

```
String sql= "Select pa.amount from "
+"c_paymentallocate pa where"
+"C_Invoice_ID=?"; //el signo ? se sustituiria despues con una variable
PreparedStatement pstmt = null; //Variable de preparacion de Recordset
ResultSet rs = null; //Recordser donde se guardara el resultado
try //Todo dentro de un try para capturar posibles errores
{
    pstmt = DB.prepareStatement(sql, null); //prepara la ejecucion del query
    pstmt.setInt(1, S_ResourceAssignment_ID); //Sustituye el ? por la variable. ojo si hay mas de una es
//secuencial, es decir el segundo ? es 2 y asi sucesivamente
    rs = pstmt.executeQuery(); //Ejecuta el query y lo coloca en rs
    if (rs.next()) //next o registro
    {
        M_Product_ID = rs.getInt (1); //almacena en variable el 1er campo. Es secuencial
    }
}
catch (SQLException e)
{
    log.log(Level.SEVERE, "Tax", e); //envia al log el error
}
```

- Otra forma de generar consultas SQL por Jose Prince. Venezuela:

```
BigDecimal idResource = (BigDecimal) mTab.getValue(I_ProductPlanning.COLUMNNAME_ResourceForm_ID); //Aquí
String sql = "SELECT porcEficiencia FROM ResourceForm WHERE ResourceForm_ID=?"; //La condición en la ser
BigDecimal porcEficiencia = DB.getSQLValueBDEX(null, sql,idResource); //Esta línea ejecuta la consulta y
```

### Instalación del Callout

1. Genere el archivo jar con su paquete o clase (acordarse de las dependencias, tener las importaciones correctas) </li>
2. el archivo .jar debe tener nombre: customization.jar </li>
3. colocar el archivo customization.jar en la carpeta lib de su ADEMPIERE\_HOME (servidor)</li>
4. Correr el RUN\_setup o RUN\_Silentsetup</li>
5. Abrir el Sistema como System Administrator</li>
6. Ir a Tabla y Columna y buscar la tabla y el campo al cual se le quiere colocar el callout (por facilidad vaya a Ventana, Pestaña y Campo, busque la ventana, seleccione la pestaña y el campo y con el botón derecho acérquese a la tabla, ésto le permitirá asegurarse de que es la ventana correcta)</li>
7. en el campo seleccionado ir al recuadro de Llamada (callout) y colocar la ruta lógica, es decir org.adempiere.callout.Calloutprb.test por ejemplo si se creó un paquete llamado adempiere.callout y dentro tiene una clase llamada Calloutprb y un método llamado test.</li>
8. Almacene y ejecute el sistema como Usuario, abra la ventana y pruebe modificando el campo</li>

## See Also

Callout Code that details each Callout functions and location

Retrieved from "<http://www.adempiere.com/Callout>"

Categories: [Developer documentation](#) | [Features](#) | [Code snippets](#) | [Documentation](#) | [Application Dictionary](#) | [Table of Contents](#)

---

- This page was last modified on 17 January 2011, at 16:20.
- Content is available under GNU Free Documentation License 1.2.