

```

public void resetGUIElementsForCommandTypeChange(int IndexOldGUIElement, int IndexNewGUIElement)
{
    boolean ArrOldGUIElement[] = new boolean [(ArrShowElements.length)];
    boolean EnableBtnSave = true;
    for (int i = 0; i < ArrShowElements.length; i++)
    {
        ArrOldGUIElement[i] = ArrShowElements[IndexOldGUIElement][i];
        // wenn das neu darzustellende GUI Element noch nicht dargestellt wurde
        if ( (ArrOldGUIElement[i] == false) && (ArrShowElements[IndexNewGUIElement][i] == true) )
        {
            if ( ArrActiveGUIComponents[i] instanceof JTextField )
            {
                ((JTextField)ArrActiveGUIComponents[i]).setText("");
                EnableBtnSave = false;
            }
        }
    }
}

```

```

public void displayBasicParameters()
{
    int [] BasicParameters = VMECommand.getBasicParameters();
    // CommandType
    ((JComboBox)ArrActiveGUIComponents[cVMEUtil.k_ComboBoxCommandType]).
    setSelectedItem( (VecListForCommandTypes.elementAt(BasicParameters[cVMEUtil.k_ComboBoxCommandType]
    )));
    IndexForOldComboBoxForEventHandlingSelection = BasicParameters[cVMEUtil.k_ComboBoxCommandType];
    // Address
    if (VMECommand.getCommandType() != cVMEUtil.k_VMECommandGotoIndex)
    {
        ((JTextField)ArrActiveGUIComponents[cVMEUtil.k_TextFieldAddress]).
        setText ( ( new Integer(BasicParameters[cVMEUtil.k_TextFieldAddress])).toString() );
    }
    // AM
    ((JComboBox)ArrActiveGUIComponents[cVMEUtil.k_ComboBoxAM]).
    setSelectedItem( (VecListForAM.elementAt(BasicParameters[cVMEUtil.k_ComboBoxAM] )));
    // DL
    ((JComboBox)ArrActiveGUIComponents[cVMEUtil.k_ComboBoxDL]).
    setSelectedItem( (VecListForDL.elementAt(BasicParameters[cVMEUtil.k_ComboBoxDL] )));
    // DoPrint
    boolean isSelected;
    if (BasicParameters[cVMEUtil.k_CheckBoxDoPrint] == 0)
    {
        isSelected = false;
    }
    else
    {
        isSelected = true;
    }
    ((JCheckBox)ArrActiveGUIComponents[cVMEUtil.k_CheckBoxDoPrint]).
    setSelected( isSelected);
    repaint();
}

```

```
public class MyActionListenerForButtonPanel implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        System.out.println("actionPerformed");
        if (e.getActionCommand() == cVMEUtil.k_ResetVMECommand)
        {
            resetParameter();
        }
        else if (e.getActionCommand() == cVMEUtil.k_SaveVMECommand)
        {
            if (checkInputState() == true)
            {
                try
                {
                    saveParameter();
                }
                catch (java.lang.Exception exc)
                {
                    System.out.println("error exception " + exc.getMessage());
                }
            }
        }
        else if (e.getActionCommand() == cVMEUtil.k_StartCommand)
        {
            if (EventHandler.getRunStatus() != true)
            {
                EventHandler.start();
            }
            EventHandler.sendAllVMECommands();
        }
    }
}
```