

# RibbonX ab Office 2010 (Word 2010 und Migration VBA, Symbolleisten aus Word 2000)

---

## Inhalt

1	RibbonX Übersicht .....	2
1.1	Randbedingungen.....	2
1.2	Gut zu wissen.....	2
1.2.1	VBA-Makros .....	2
2	Word 2000 Migration.....	3
2.1	Symbolleisten entfernen .....	3
2.2	VBA – Makros anpassen .....	4
3	Word 2010 RibbonX – Programmierung .....	5
3.1	DOCxx – verhält sich wie ZIP-Datei.....	5
3.1.1	_rels .....	5
3.1.2	Eigene Registerkartenanpassungen .....	5
3.1.3	Eigene Datentypen, Eigen Contenttypen einführen [Content_Types.xml] .....	6
3.2	CustomUI.xml .....	7
3.2.1	Bearbeitungshinweise .....	7
3.2.2	Struktur der „Docx/Zip“ Datei.....	7
3.2.2.1	CustomUI – Strukturweiterung .....	8
3.2.3	CustomUI.xml.....	9
3.2.3.1	Beispiel .....	10
3.2.3.2	Befehlserklärungen zur customUI.xml .....	14
3.2.3.2.1	Hilfreiche Quellen:.....	14
3.2.3.2.2	Befehle aus dem oberen Beispiel.....	15
3.2.3.3	Eigene Abbildungen in die “DOCx/ZIP“-Zip Datei einbringen .....	16

# 1 RibbonX Übersicht

## 1.1 Randbedingungen

Nachfolgende Informationen sind unter Word 2010 ermittelt worden und in einem Word 2010 Document angewendet.

## 1.2 Gut zu wissen

In der CustomUI.xml darf nicht mit deutschen Umlauten oder Sonderzeichen wie, ä, ö, ß direkt gearbeitet werden, hier müssen die entsprechenden Entities aus der HTML bzw. XML – Sprachumgebung eingesetzt werden. Beachtet man dies nicht, meldet Word beim Start des Dokuments einen Fehler und die Registerkarte im Menüband wird nicht angezeigt.

-> Die Sonderzeichen der dt. Sprache müssen über den numerischen Code der Entities eingegeben werden.

Vgl.: <http://de.selfhtml.org/html/referenz/zeichen.htm#allgemeines>

ä = &#228 ;

ß = &#223 ;

ü = &#252 ;

Zeilenumbruch innerhalb eines „labels“

Zeilenumbruch = &#13;

### 1.2.1 VBA-Makros

Können von den Registerkarten nur aufgerufen werden, wenn die VBA-Makros wie folgt beginnen:

“Sub Makroname (control As IRibbonControl)”

## 2 Word 2000 Migration

### 2.1 Symbolleisten entfernen

Word Dokument mit einem ZIP-Programm entpacken.

Datei „attachedToolbars.bin“ im Verzeichnis „word“ löschen.

Datei „customizations.xml“ im Verzeichnis „word“ öffnen und alle Verweise auf Symbolleisten und dort hinterlegte Verweise auf Makros und Programmfunktionen löschen.

#### Beispiel:

#### Ausgangszustand:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<wne:tcg xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"><wne:keymaps><wne:
keymap wne:kcmPrimary="0443"><wne:macro
wne:macroName="PROJECT.CE_COMMANDBARS.CE_COMMANDBARS_SICHTBAR"/></wne:key
map><wne:keymap wne:kcmPrimary="0445"><wne:macro
wne:macroName="PROJECT.CE_LIFESHOW.CE_LIFESHOW"/></wne:keymap><wne:keymap
wne:kcmPrimary="0451"><wne:macro
wne:macroName="PROJECT.CE_ARBEITSHILFE.CE_ARBEITSHILFE"/></wne:keymap><wne:key
map wne:kcmPrimary="0452"><wne:macro
wne:macroName="PROJECT.CE_RISKASSESSMENTSHOW.CE_RISKASSESSMENTSHOW"/></wne:
keymap><wne:keymap wne:kcmPrimary="0457"><wne:macro
wne:macroName="PROJECT.ZURBEARBEITUNGSPOSITION.ZURBEARBEITUNGSPOSITION"/></w
ne:keymap></wne:keymaps><wne:toolbars><wne:toolbarData
r:id="rld1"/></wne:toolbars></wne:tcg>
```

#### Nach der Bearbeitung:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<wne:tcg xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"></wne:tcg>
```

Alle Verzeichnisse und Dateien, die entpackt wurden, wieder mit einem Zip-Programm packen. Dabei die Endung beachten.

Für Word-Dateien mit Makros: \*.docm

Für Word-Dateien ohne Makros: \*.docx

## 2.2 VBA – Makros anpassen

Word-Datei öffnen und eventuelle Fehlermeldungen aus Makroaufrufen über das Register „Entwicklertools“, „VBA“ und den VBA Editor lösen (auskommentieren, Code löschen oder notwendige Aufrufe umprogrammieren).

## 3 Word 2010 RibbonX – Programmierung

### 3.1 DOCxx – verhält sich wie ZIP-Datei

In der Word – Zip – Datei gibt es folgende Grundstruktur an Ordnern und Dateien:

*\_rels*

*docProps*

*word*

*[Content\_Types].xml*

#### 3.1.1 \_rels

\_rels enthält eine Datei, „.rels“, in der alle Bezüge innerhalb der Word-/Zip-Datei geregelt werden. auch für CustomUI.xml – Dateie(n), die hier eingetragen werden müssen / muss.

#### Beispielintrag in die .rels Datei für eine CutomUI.xml Datei:

```
<Relationship Id="rId4"  
Type=http://schemas.microsoft.com/office/2007/relationships/ui/extensibility  
Target="customUI/customUI.xml"/>
```

Diese Zeile gibt mit dem Attribut „Target“ den Pfad und den Namen der XML-Datei an, in der die Kundenanpassungen (custom) für die Benutzeroberfläche (UI User Interface) gemacht werden. Dabei kann der Unterordner beliebig benannt werden, er muss nicht „customUI“ heißen.

Die Angabe muss sich immer auf die oberste Ebene beziehen. Auf der Obersten Ebene der „DOC“ / „ZIP“ – Verzeichnisse liegt die Datei [Content\_Types].xml.

#### 3.1.2 Eigene Registerkartenanpassungen

Müssen, wie beschrieben in der „.rels“ bekannt gemacht werden (Unterverzeichnis in „Zip / Doc“) und werden dann in einer speziellen Datei (meist „customUI.xml“) programmiert.

### 3.1.3 Eigene Datentypen, Eigen Contenttypen einführen [Content\_Types.xml]

Will man eigene Datentypen oder Contenttypen einführen, muss man diese in der [Content\_Types.xml] tun, die auf der obersten Ebene der „DOC/ZIP“ Datei liegt.

#### Beispiel Grafikformat bekannt machen:

Um Grafiken im Format JPG innerhalb der DOC-Datei in der Ribbon-Bar verwenden zu können, muss man folgenden Eintrag in die [Content\_Types.xml] machen.

**<Default Extension="jpg" ContentType="image/.jpg"/>**

Anfang der Datei [Content\_Types.xml]:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types"><Default
Extension="bin" ContentType="application/vnd.ms-office.vbaProject"/><Default Extension="png"
ContentType="image/png"/><Default Extension="rels"
ContentType="application/vnd.openxmlformats-package.relationships+xml"/><Default
Extension="xml" ContentType="application/xml"/><Default Extension="jpg"
ContentType="image/.jpg"/><Override PartName="/word/document.xml"
ContentType="application/vnd.ms-word.document.macroEnabled.main+xml"/>
```

Nach diesem Eintrag kann man die selbst in die „DOCx/ZIP“-Zip Datei eingebrachten jpg Dateien mit dem Attribute „image“ ansprechen; Beispiele folgen bei der Erklärung zur „costumUI.xml“.

## 3.2 CustomUI.xml

Wie passe ich die Registerkarten der Benutzeroberfläche den Bedürfnissen von Benutzern an?

### 3.2.1 Bearbeitungshinweise

Die Word-Datei kann in einem Zip-Programm geöffnet werden. Falls man Änderungen an den xml-Dateien innerhalb der Word-Datei über das Zip-Programm ausführt, werden diese Änderungen nach dem Schließen der Word-Datei durch das Zip-Programm gespeichert. Die Änderungen sind somit in der Word-Datei vorhanden.

Will man allerdings enthaltene VBA-Programme anpassen, muss man die Word-Datei normal mit Word öffnen und die „Entwicklungsumgebung -> VBA“ aufrufen.

Die „DOCx/ZIP“-Zip Datei kann auch vollständig in ein Verzeichnis entpackt werden, um dort bearbeitet zu werden.

Um wieder eine „docx“ Datei zu erhalten muss man die gesamten entpackten Dateien und Verzeichnisse wieder mit einem Zip-Programm zu einer Datei zusammenfügen.

Achten muß mann auf die Extension, die man dieser Datei gibt, die Extension muss je nach Verwendungstyp der Word-Datei docx, docm, dots oder dotm lauten.

### 3.2.2 Struktur der „Docx/Zip“ Datei

Öffnet man eine docx, docm, dotx, dotm, Datei von Word 2010 mit einem ZIP-Programm, so sieht man folgende Verzeichnis- und Dateigrundstruktur:

*\_rels*

*docProps*

*word*

*[Content\_Types].xml*

Diese Dateien stellen die innere Basisstruktur einer Word 2010 Datei dar.

Um eine benutzerdefinierte Anpassung der Benutzeroberfläche (User Interface) vornehmen zu können, muss eine Datei „costumUI.xml“ angelegt werden. Wie diese in der Struktur der Word-Datei bekannt gemacht wird, ist in den vorhergehenden Kapiteln beschrieben.

### 3.2.2.1 CustomUI – Strukturweiterung

Um den Überblick wahren zu können, legt man ein neues Verzeichnis mit der Benennung „customUI“ an, in der die Datei „customUI.xml“ angelegt wird. Die innere Struktur der „Docx/Zip“ Datei sieht dann so aus:

*\_rels*

*customUI*

*docProps*

*word*

*[Content\_Types].xml*

(Von Word selbst können noch weitere Unterverzeichnisse angelegt sein, in denen z. B. die für einzelne Befehle vergebenen Tastenkombinationen abgelegt sind.)

### 3.2.3 CustomUI.xml

Die „customUI.xml“ dient der Anpassung der Registerkarten im Ribbon Menü.

Die Datei steht in unserem Beispiel im Verzeichnis customUI innerhalb der „Docx/ZIP“ Datei.

customUI/customUI.xml

Das nachfolgende Beispiel zeigt eine customUI.xml Datei.

An diesem Beispiel werden einige Befehle im Anschluss an das Beispiel erklärt.

### 3.2.3.1 Beispiel

```
<customUI xmlns="http://schemas.microsoft.com/office/2009/07/customui">
<!-- 1te Version zur Verwendung von Martin Mattusch, Gross-Bieberau, 2013-10-17 -->
<!-- "startFromScratch" bestimmt ob Word mit oder ohne Standardregisterkarten gestartet wird; oder nur die selbst erstellte Registerkarte angezeigt wird -->
<ribbon startFromScratch="false">
  <tabs>
<!-- "insertBeforeMso=TabHome" bestimmt, dass die selbst erstellten Registerkarten vor den Standardregisterkarten von Word angezeigt werden -->
<tab id="CustomTab1" label="CE-MRL" insertBeforeMso="TabHome">
  <group id="CustGrp1" label="Relevant">
    <button id="TB1" label="Gefahr relevant j / n ?" imageMso="FrontPageToggleBookmark" onAction="CE_Relevant_Show" size="large" />
  </group>

  <group id="CustGrp2" label="Lebensphase">
    <button id="TB2" label="Lebensphase" imageMso="StartAfterPrevious" onAction="CE_LifeShow" size="large" />
  </group>

  <group id="CustGrp3" label="Risikoeinsch&#228;tzung">
    <button id="TB3" label="Risikoeinsch&#228;tzung" imageMso="EquationMatrixGallery" onAction="CE_RiskAssessmentShow" size="large" />
  </group>

  <group id="CustGrp4" label="Schutzma&#223;nahmen">
    <button id="TB4" label="Schutzma&#223;nahmen" imageMso="Risks" onAction="CE_Action_Show" size="large" />
  </group>

  <group id="CustGrp5" label="Rest-Risiko">
    <button id="TB5" label="Restrisikoeinsch&#228;tzung" imageMso="EquationMatrixGallery" onAction="CE_RiskAssessmentShow" size="large" />
  </group>

  <group id="CustGrp6" label="Risikobeurteilung">
    <button id="TB6" label="Restrisiko akzeptabel (okay) ja / nein ?" imageMso="SetPertWeights" onAction="CE_RestRisk_OK_Show" size="large" />
  </group>

  <group id="CustGrp7" label="Verantwortlich f&#252;r die Umsetzung">
    <button id="TB7" label="Verantwortlich f&#252;r die Umsetzung der Schutzma&#223;nahme" imageMso="Head" onAction="CE_ResponsibleShow" size="large" />
  </group>

  <group id="CustGrp8" label="Kontrolle, ob Schutzma&#223;nahme umgesetzt wurde">
    <button id="TB8" label="Ist die Schutzma&#223;nahme umgesetzt &#13;j / n ? (Datum der Kontrolle eintragen)" imageMso="TagMarkComplete" onAction="CE_Implement_Show" size="large" />
  </group>

<!-- Kommentar -->
</tab>
</ribbon>
</customUI>
```

```
<group id="grp01" label="Vorschriften / Normen / Hilfe">
<menu id="mnu0" size="large" itemSize="normal" label="EU" image="eu-flag-yellow" >
  <button id="menBtn01" label="EG-MRL (Fundstelle im Amtsblatt - HTML)" image="eu-flag-yellow" onAction="MRL_2006_42_EG_DE" />

  <button id="EG-MRL-EU-Server-PDF" label="EG-MRL (Text als PDF)" image="eu-flag-yellow" onAction="MRL_2006_42_EG_DE_PDF" />

  <menuSeparator id="menSpt01" />

  <!-- Untermenue in einem Menue-->
  <menu id="men02" itemSize="normal" label="Weiteres zur EG-MRL" image="eu-flag-yellow" >
<button id="menBtn02" label="Harmonisierte Normen zur EG-MRL" image="eu-flag-yellow" onAction="MRL_EU_Harmonized_Standards" />
<menuSeparator id="menSpt02" />
<button id="EG-MRL-Leitfaden-EU" label="EU-Leitfaden zur EG-MRL" image="eu-flag-yellow" onAction="MRL_2006_42_EG_Leitfaden" />
  </menu>

  </menu>

  <menu id="mnu3" size="large" itemSize="normal" label="Normen" image="normen" >
  <button id="menBtn04" label="Normendatenbank SPG" image="normen" onAction="NormendatenbankSPG" />
  </menu>

  <menu id="mnu4" size="large" itemSize="normal" label="SPG" image="spg-logo" >
  <button id="menBtn05" label="GD-Nr. 33 EG-MRL" image="spg-logo" onAction="MRL_SPG_GD_33_de" />
  </menu>

</group>

</tab>
  </tabs>
</ribbon>

<!-- Kontextmenue auf Rechter-Maustaste -->
<contextMenus>
  <contextMenu idMso="ContextMenuText">
  <menuSeparator id="MySeparator" />
  <menu id="MySubMenu" label="EG-MRL" image="spg-logo" >

  <button id="ContMenueSP01" label="Gefahr relevant j / n ?" imageMso="FrontPageToggleBookmark" onAction="CE_Relevant_Show" />

  <button id="ContMenueSP02" label="Lebensphase" imageMso="StartAfterPrevious" onAction="CE_LifeShow" />

  <button id="ContMenueSP03" label="Risikoeinsch&#228;tzung" imageMso="EquationMatrixGallery" onAction="CE_RiskAssessmentShow" />

  </contextMenu>
</contextMenus>
```

```
<button id="ContMenueSP04" label="Schutzma&#223;nahmen" imageMso="Risks" onAction="CE_Action_Show" />
<button id="ContMenueSP05" label="Restrisiko&#228;tzung" imageMso="EquationMatrixGallery" onAction="CE_RiskAssessmentShow" />
<button id="ContMenueSP06" label="Restrisiko akzeptabel (okay) ja / nein ?" imageMso="SetPertWeights" onAction="CE_RestRisk_OK_Show" />
<button id="ContMenueSP07" label="Verantwortlich f&#252;r die Umsetzung der Schutzma&#223;nahme" imageMso="Head" onAction="CE_ResponsibleShow" />
<button id="ContMenueSP08" label="Ist die Schutzma&#223;nahme umgesetzt &#13;j / n ? (Datum der Kontrolle eintragen)" imageMso="TagMarkComplete" onAction="CE_Implement_Show" />
</menu>
</contextMenu>

<contextMenu idMso="ContextMenuTextTable">
<!-- <menuSeparator id="MySeparatorTable" /> -->
<menu id="MySubMenuTable" label="EG-MRL" image="spg-logo" >
    <button id="ContMenueTableSP01" label="Gefahr relevant j / n ?" imageMso="FrontPageToggleBookmark" onAction="CE_Relevant_Show" />
    <button id="ContMenueTableSP02" label="Lebensphase" imageMso="StartAfterPrevious" onAction="CE_LifeShow" />
    <button id="ContMenueTableSP03" label="Risiko&#228;tzung" imageMso="EquationMatrixGallery" onAction="CE_RiskAssessmentShow" />
    <button id="ContMenueTableSP04" label="Schutzma&#223;nahmen" imageMso="Risks" onAction="CE_Action_Show" />
    <button id="ContMenueTableSP05" label="Restrisiko&#228;tzung" imageMso="EquationMatrixGallery" onAction="CE_RiskAssessmentShow" />
    <button id="ContMenueTableSP06" label="Restrisiko akzeptabel (okay) ja / nein ?" imageMso="SetPertWeights" onAction="CE_RestRisk_OK_Show" />
    <button id="ContMenueTableSP07" label="Verantwortlich f&#252;r die Umsetzung der Schutzma&#223;nahme" imageMso="Head" onAction="CE_ResponsibleShow" />
    <button id="ContMenueTableSP08" label="Ist die Schutzma&#223;nahme umgesetzt &#13;j / n ? (Datum der Kontrolle eintragen)" imageMso="TagMarkComplete" onAction="CE_Implement_Show" />
</menu>
</contextMenu>
</contextMenus>
</customUI>
```

Die zugehörige Anpassung sieht im Ergebnis wie folgt aus; Registerkarte „CE-MRL“



### 3.2.3.2 Befehlserklärungen zur customUI.xml

#### 3.2.3.2.1 Hilfreiche Quellen:

<http://msdn.microsoft.com/en-us/library/cc313070%28v=office.12%29.aspx>

(CustomUI Specification)

<http://msdn.microsoft.com/en-us/library/dd926139%28v=office.12%29.aspx>

(CustomUI Elements)

<http://www.accessribbon.de/en/?Access - Ribbons:Ribbon XML Controls>

PDF-Dokumentationen:

<http://download.microsoft.com/download/D/3/3/D334A189-E51B-47FF-B0E8-C0479AFB0E3C/%5BMS-CUSTOMUI%5D.pdf>

<http://download.microsoft.com/download/D/3/3/D334A189-E51B-47FF-B0E8-C0479AFB0E3C/MSOFFSTAND.zip>

Grafiken aus dem Office Paket

<http://soltechs.net/CustomUI/IMG1.htm>

[www.accessribbon.de](http://www.accessribbon.de) (*Office2010\_RibbonIcons.xlsm*)

## 3.2.3.2.2 Befehle aus dem oberen Beispiel

Was	Funktion	Beispiel / Bemerkung
insertBeforeMso="TabHome"	Bestimmt, dass die selbst erstellten Registerkarten vor den Standardregisterkarten von Word angezeigt werden	<tab id="CustomTab1" label="CE-MRL" insertBeforeMso="TabHome">
imageMso="FrontPageToggleBookmark"	Bestimmt das grafische Symbol, das zum Eintrag angezeigt wird. imageMso spricht Symbole an, die mit Word bzw. Office geliefert werden.	Um selbst in der Word-Datei (ZIP Struktur) abgelegte Abbildungen ansprechen zu können, muss man das Attribut image verwenden. Siehe separate Erklärung hierzu.
onAction="CE_Relevant_Show"	Spricht hier eine VBA Makro mit dem Namen CE_Relevant_Show an.	Das Makro muss in der Definition wie folgt geschrieben sein:  Sub CE_Action_Show(control As IRibbonControl)  ...  End Sub
size="large"	Bestimmt, dass das Symbol groß über die ganze Gruppe (group) innerhalb der Registerkarte (tab) dargestellt wird.	„normal“ bewirkt, dass das Symbol in normaler Größe dargestellt wird.

### 3.2.3.3 Eigene Abbildungen in die „DOCx/ZIP“-Zip Datei einbringen

Um eigene Abbildungen einbringen zu können, muss der Dateityp in der „[Content\_Types].xml“ Datei bekannt sein oder dort bekannt gemacht werden. Wie dies geht ist weiter oben in den Kapiteln beschrieben.

Eine sinnvolle Struktur sieht wie folgt aus, dabei werden zwei Verzeichnisse neben der customUI.xml Datei angelegt.

*\_rels*

*images*

So, dass das Verzeichnis „customUI“ dann wie folgt aufgebaut ist:

*\_rels*

*images*

*customUI.xml*

Im Verzeichnis „\_rels“ wird wieder die Beziehung (Relationship) definiert, im Verzeichnis „images“ liegen die eigenen Abbildungen, die man in die Gestaltung der Registerkarten einbringen will. Die Datei, im Verzeichnis „\_rels“ heißt: „customUI.xml.rels“ und im folgenden Beispiel ist gezeigt, wie die Beziehungsdefinition funktioniert.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">

<Relationship Id="eu-flag-yellow" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
Target="images/EU-flag_yellow_low.jpg"/>

<Relationship Id="normen" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
Target="images/normen.jpg"/>

<Relationship Id="spg-logo" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
Target="images/SPG-Logo-Symbol-CSD.jpg"/>
</Relationships>
```

Die Grafikdateien werden in der customUI.xml über die „ID“ (Bsp. oben „spg-logo“) angesprochen.