

Dokumentation Ribbon-Admin 2010

Dokumentation zum Produkt **Ribbon-Admin 2010**.

Stand: 16. Januar 2012

© 2007-2012 André Minhorst
Borkhofer Straße 17
47137 Duisburg
<http://www.ribbon-admin.de>
info@ribbon-admin.de

1 Vorbereitung und Voraussetzungen

Der Ribbon-Admin 2010 arbeitet nur mit Access 2010 (32-bit) zusammen, nicht mit anderen Versionen – aber vor allem nicht mit der 64-bit-Version!

Access hat die eigentümliche Eigenschaft, dass es Fehler in der Ribbon-Definition standardmäßig nicht meldet. Das heißt, das ein fehlerhaftes benutzerdefiniertes Ribbon einfach nicht angezeigt wird. Ändern können Sie dies, indem Sie die Access-Option **Fehler des Benutzeroberflächen-Add-Ins anzeigen** in den Access-Optionen aktivieren. Diese Option sollten Sie zumindest während der Entwicklung und der Testphase unbedingt aktivieren!

2 Installation

Die Installation des **Ribbon-Admin 2010** erfolgt mithilfe des Add-In-Managers.

Führen Sie die folgenden Schritte durch:

- » Entpacken Sie die im Zip-Archiv enthaltene Datei RibbonAdmin2010.acdda in ein beliebiges Verzeichnis.
- » Starten Sie **Access 2010**.
- » Wählen Sie den Ribbon-Eintrag **Datenbank-Tools|Add-Ins|Add-In-Manager** aus.
- » Klicken Sie auf **Neues hinzufügen ...** und wählen Sie im Auswahldialog die soeben entpackte Datei **Ribbon-Admin2010.acdda** aus.
- » Schließen Sie den Dialog.
- » Betätigen Sie den Ribbon-Befehl **Datenbank-Tools|Add-Ins**. Dort finden Sie nun den Eintrag **RibbonAdmin2010** vor, mit dem Sie den **Ribbon-Admin 2010** starten.

2.1 Installation eines Updates

Wenn Sie den **Ribbon-Admin 2010** bereits installiert haben und eine neue Version installieren möchten, gehen Sie wie oben vor. Eventuell bereits erstellte Ribbon-Definitionen bleiben erhalten, da diese in der Backend-Datenbank **RibbonAdmin2010_Data.accdb** gespeichert wurden.

3 Start des Ribbon-Admins

Ribbon-Admin 2010 starten Sie über den Ribbon-Eintrag **Datenbank-Tools|Add-Ins|Ribbon-Admin 2010**. Voraussetzung für den Start ist, dass Sie die Anwendung, der Sie Ribbon-Definitionen hinzufügen möchten, geöffnet haben. Ist keine Anwendung geöffnet, wird der Start abgebrochen.

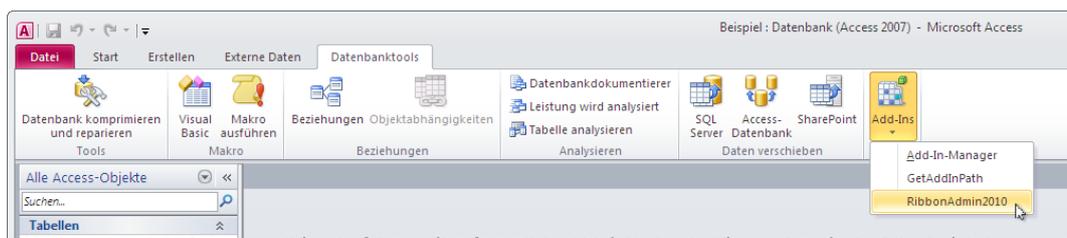


Bild 1: Start des **Ribbon-Admin 2010**

Ribbon-Admin 2010 zeigt bei der Erstinstallation zunächst den Dialog zum Eingeben der Registrierungsdaten an (siehe **Bild 2**). Diese Daten erhalten Sie mit der E-Mail, die auch den Downloadlink enthält. Nach der Eingabe der Daten erscheint der eigentliche Startbildschirm und schließlich das Formular des **Ribbon-Admin 2010**.

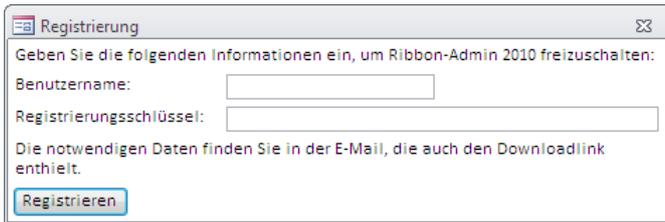


Bild 2: Eingabe der Registrierungsdaten

4 Erstellen von Ribbon-Definitionen

Nach dem Start beginnen Sie gleich mit dem Anlegen der ersten Ribbon-Definition.

4.1 Anlegen einer Anwendung

Das erste Formular sieht beim ersten Start noch recht spartanisch aus – siehe **Bild 3**. Dementsprechend gibt es für den ersten Schritt auch kaum Alternativen, denn Sie müssen zunächst eine Anwendung festlegen. Eine Anwendung fasst eine oder mehrere Ribbon-Definitionen zusammen. Dazu haben Sie zwei Möglichkeiten: Entweder Sie betätigen die einzige aktuell sichtbare Ribbon-Schaltfläche **Neue Anwendung** oder den gleichnamigen Kontextmenü-Eintrag des einzigen Elements des TreeView-Steuerelements.

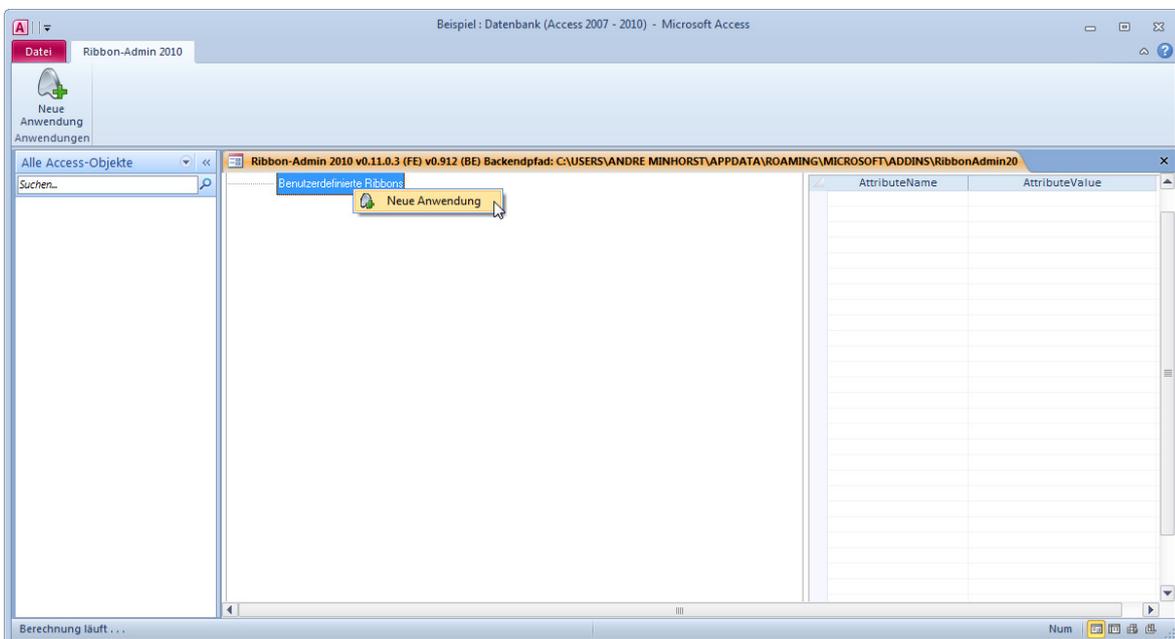


Bild 3: Der **Ribbon-Admin 2010** vor dem Anlegen der ersten **customUI**-Definition

Nach dem Betätigen dieses Befehls erscheint ein neuer Eintrag im TreeView-Steuerelement. Sie können gleich den Namen der Anwendung festlegen (siehe **Bild 4**).

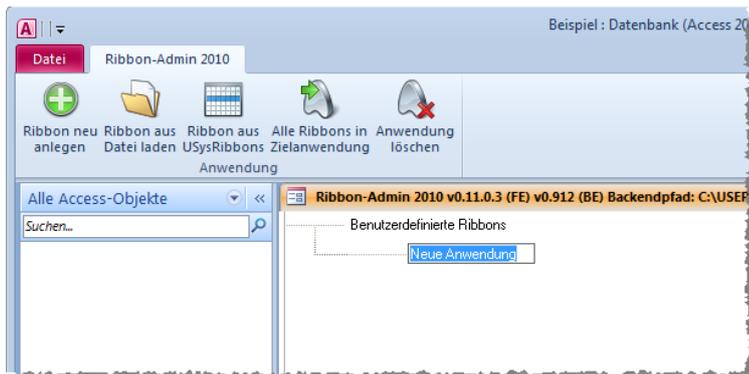


Bild 4: Anlegen des ersten Elements als Repräsentation einer Anwendung

Das Ribbon und auch das Kontextmenü zeigt für diesen Anwendungs-Eintrag die folgenden Optionen an:

- » **Ribbon neu anlegen:** Legt ein Root-Element für ein neues Ribbon an. Dieses enthält noch keine Informationen zur eigentlichen Definition des Ribbons.
- » **Ribbon aus Datei laden:** Lädt ein Ribbon aus einer XML-Datei, das Sie gegebenenfalls von Hand erstellt oder mit einem anderen Tool erzeugt haben und zeigt es im **Ribbon-Admin 2010** an.
- » **Ribbon aus USysRibbons:** Lädt ein Ribbon aus der Tabelle USysRibbons, soweit vorhanden. Diese Tabelle speichert die in der Anwendung verwendeten Ribbon-Definitionen. Wenn Sie diese mit dem Ribbon-Admin 2010 bearbeiten wollen, können Sie diesen Vorgang hiermit beginnen.
- » **Alle Ribbons in Zielanwendung:** Überträgt alle für diese Anwendung erstellten Ribbon-Definitionen in die Tabelle USysRibbons der aktuell geöffneten Datenbank. Eventuell bestehende Einträge können je nach Wunsch überschrieben oder beibehalten werden.
- » **Anwendung löschen:** Löscht diesen Anwendungsknoten und alle enthaltenen Ribbon-Definitionen.

4.2 Anlegen einer Ribbon-Definition

Das Root-Element legen Sie mit einem Klick auf den Ribbon-Eintrag **Ribbon neu anlegen** an. Mit dem neuen Eintrag erscheinen gleich auch eine ganze Reihe weiterer Optionen im Ribbon (siehe **Bild 5**).

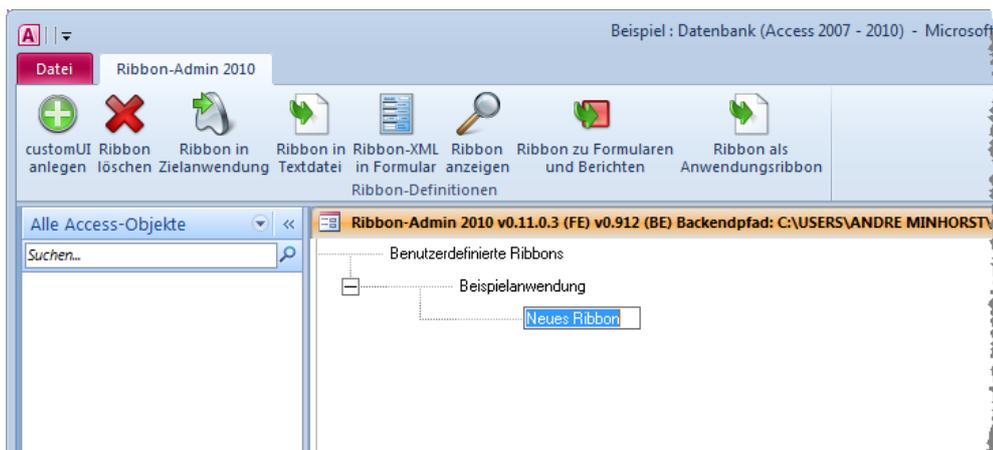


Bild 5: Nach dem Anlegen des Root-Elements der ersten Ribbon-Definition erscheinen neue Befehle im Ribbon und im Kontextmenü.

Sie haben nun die folgenden Möglichkeiten:

- » **customUI anlegen**: Legt das erste Element der Ribbon-Definition an.
- » **Ribbon löschen**: Löscht die komplette Ribbon-Definition samt aller enthaltenen Elemente.
- » **Ribbon in Zielanwendung**: Schreibt die Ribbon-Definition in die Tabelle USysRibbons der aktuell geöffneten Datenbank.
- » **Ribbon in Textdatei**: Schreibt die aktuelle Ribbon-Definition im XML-Format in eine Textdatei.
- » **Ribbon-XML in Formular**: Zeigt die aktuelle Ribbon-Definition im XML-Format in einem Formular an.
- » **Ribbon anzeigen**: Zeigt den aktuellen Zustand des Ribbons entsprechend der Definition im Ribbon der aktuellen Anwendung an.
- » **Ribbon zu Formularen und Berichten**: Öffnet einen Dialog, mit dem Sie die in der Tabelle **USysRibbons** gespeicherten Ribbon-Definitionen zu Formularen und Berichten der aktuell geöffneten Datenbank zuweisen können.
- » **Ribbon als Anwendungsribbon**: Legt das aktuelle Ribbon als Anwendungsribbon fest. Das heißt, dass diese Ribbon-Definition beim nächsten Start umgesetzt wird.

Achtung: Backstage-Anpassungen, also Anpassungen des Bereichs, der sich hinter dem **Datei**-Registerreiter des Ribbons verbergen, können leider nicht live angezeigt werden. Diese müssen Sie mit dem Befehl **Ribbon in Zielanwendung** erst in die aktuell geöffnete Datenbank übertragen und diese dann neu starten, um die Veränderungen im Backstage-Bereich zu begutachten.

Die folgenden Abschnitte beschreiben nun zunächst das Anlegen der eigentlichen Ribbon-Definition. Danach werden die übrigen hier aufgeführten Funktionen vorgestellt.

4.3 customUI-Element anlegen

Mit einem Klick auf den Ribbon-Eintrag **customUI** anlegen beziehungsweise des gleichnamigen Eintrags im Kontextmenü fügen Sie der Ribbon-Definition das eigentliche Hauptelement der XML-Definition des Ribbons hinzu (siehe **Bild 6**).

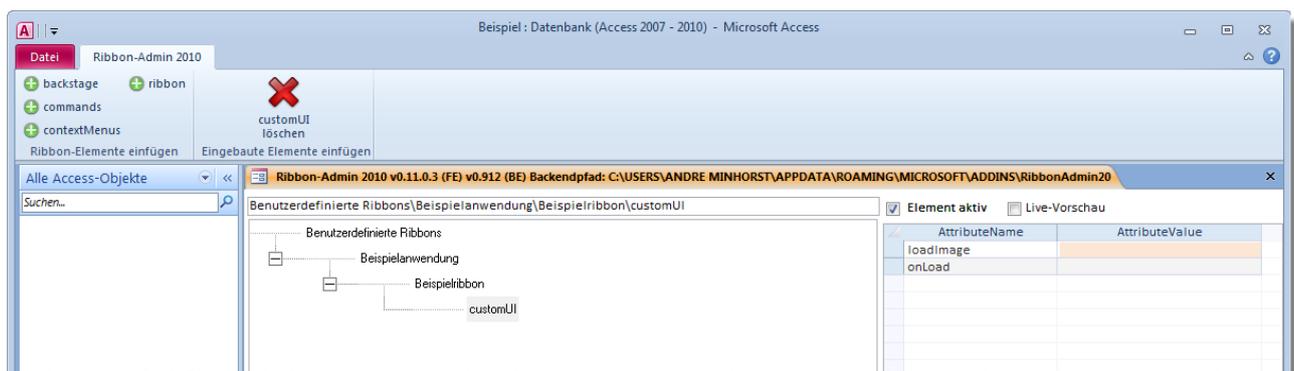


Bild 6: Ribbon mit dem **customUI**-Element

Mit dem **customUI**-Element erscheinen im rechten Bereich auch erstmals einige Attribute: **loadImage** und **onLoad**. In diesem Fall handelt es sich um zwei Callback-Attribute, denen Sie die Namen von VBA-Prozeduren mit entsprechender Syntax zuweisen – mehr dazu weiter unten.

Vorerst sind die neuen Befehle im Ribbon und im Kontextmenü für das **customUI**-Element interessant: Sie können unterhalb dieses Elements nun die folgenden Elemente anordnen:

- » **backstage**
- » **commands**
- » **contextualMenus**
- » **ribbon**

Ein weiterer Befehl erlaubt das Löschen des **customUI**-Elements.

4.4 ribbon-Element anlegen

Wenn Sie eine Änderung des Aussehens des Ribbons herbeiführen möchten, legen Sie ein **ribbon**-Element an, das weitere Elemente aufnehmen kann. Wenn Sie das **ribbon**-Element anklicken, erscheinen wiederum die passenden Befehle im Ribbon und im Kontextmenü. Außerdem finden Sie in der Attributliste den Eintrag **startFromScratch** vor (siehe **Bild 7**).



Bild 7: Der **Ribbon-Admin 2010** mit einem frisch angelegten **ribbon**-Element.

Den Wert dieses Attributs können Sie von Hand eintragen (**true/false**) oder mit dem Kontextmenü hinzufügen. Dem **ribbon**-Element können Sie weitere Elemente unterordnen, und zwar die Elemente **contextualTabs**, **gat** und **tabs**. Der Weg für die Anpassung der Ribbon-Leiste selbst erfordert das Hinzufügen eines **tabs**-Elements mit dem entsprechenden Befehl.

Das **ribbon**-Element können Sie im Gegensatz zu den bisher angelegten Elementen nicht nur löschen, sondern auch kopieren und ausschneiden. Sie können dieses Element somit auch in eine andere Ribbon-Definition verschieben oder dorthin kopieren.

4.5 tabs-Element anlegen

Wenn Sie nun ein **tabs**-Element hinzufügen, stellen Sie fest, dass dieses gar keine Attribute aufweist. Außerdem können Sie diesem lediglich eines oder mehrere **tab**-Elemente unterordnen. Sie können aber auch eingebaute Tabs einfügen, das **tabs**-Element kopieren oder ausschneiden oder dieses löschen (siehe **Bild 8**).

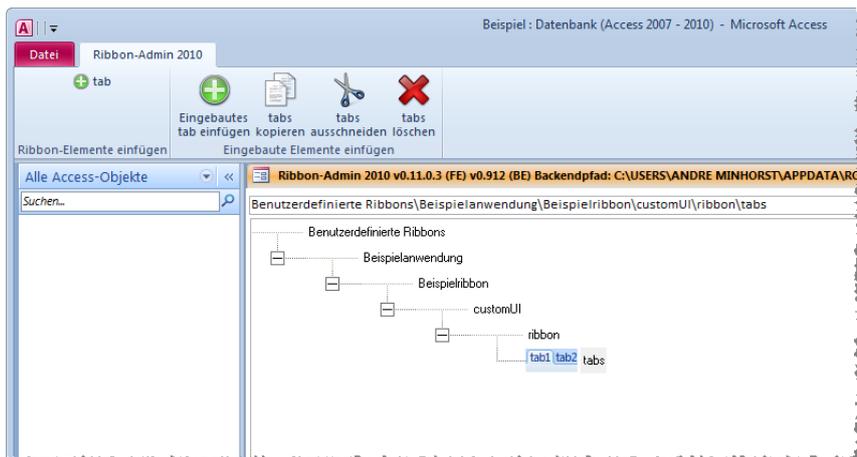


Bild 8: Optionen des **tabs**-Elements

4.6 tab-Element anlegen

Ein **tab**-Element legen Sie durch einen Klick auf den entsprechenden Eintrag im Ribbon oder Kontextmenü des **tabs**-Elements an (siehe Bild 9).

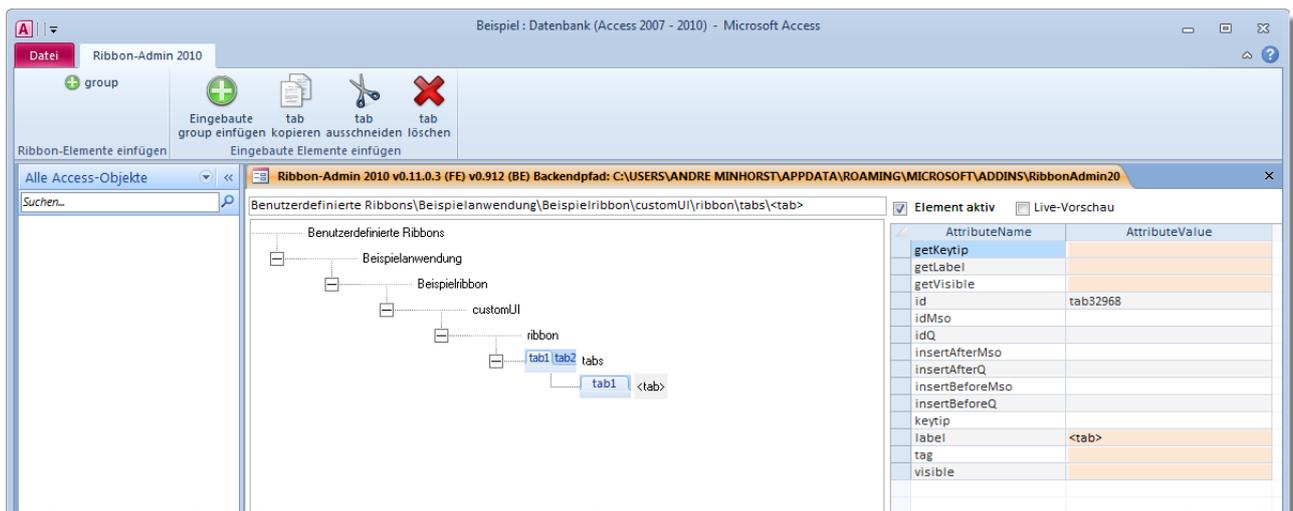


Bild 9: Hinzufügen eines **tab**-Elements

Das **tab**-Element enthält Eigenschaften, die das Aussehen des Ribbons beeinflussen. Diese passen Sie im Attributbereich rechts an. Stellen Sie dort zumindest den Wert des Attributs **label** auf einen Text ein, der im Tab-Reiter erscheinen soll.

Dem **tab**-Element können Sie eines oder mehrere **group**-Element unterordnen. Außerdem können Sie eingebaute **group**-Elemente einfügen.

4.7 Eingebautes tab-Element anlegen

Wenn Sie den Befehl **Eingebautes tab einfügen** betätigen, erscheint der Dialog aus Bild 10 zur Auswahl der gewünschten **tab**-Definition. Sie wählen dort zunächst das **tabSet** aus, in dem sich das **tab**-Element befindet. Wenn Sie beispielsweise das Seitenansicht-Tab in eine eigene Ribbon-Definition schreiben möchten, die beim Anzeigen der Berichtsvorschau erscheint, wählen Sie zunächst **None (Core Tab)** und dann **Seitenansicht (Ta-**

bPrintPreviewAccess) aus. Diese Anforderung kommt beispielsweise auf Sie zu, wenn Sie der Benutzer der Anwendung die Runtime-Version einsetzt – diese zeigt das Seitenansicht-Ribbon nicht an.

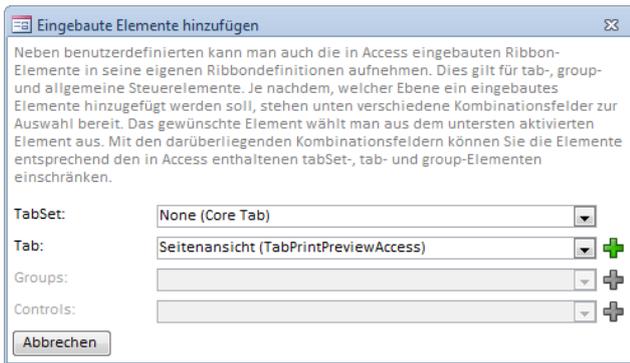


Bild 10: Auswählen eines eingebauten **tab**-Elements zum Einfügen unterhalb des **tabs**-Elements.

Genaugenommen ist das Einfügen kompletter eingebauter Tabs nur mit einem Trick möglich: Ein eingebautes Tab besteht nämlich aus einer Reihe von **group**-Elementen, die statt eines einfachen eingebauten **tab**-Elements zusammen mit dem neuen **tab**-Element hinzugefügt werden (siehe **Bild 11**). Diese group-Elemente zeichnen sich wie auch eingebaute Steuerelemente dadurch aus, dass Sie lediglich einen Wert für das Attribut **idMso** angeben.

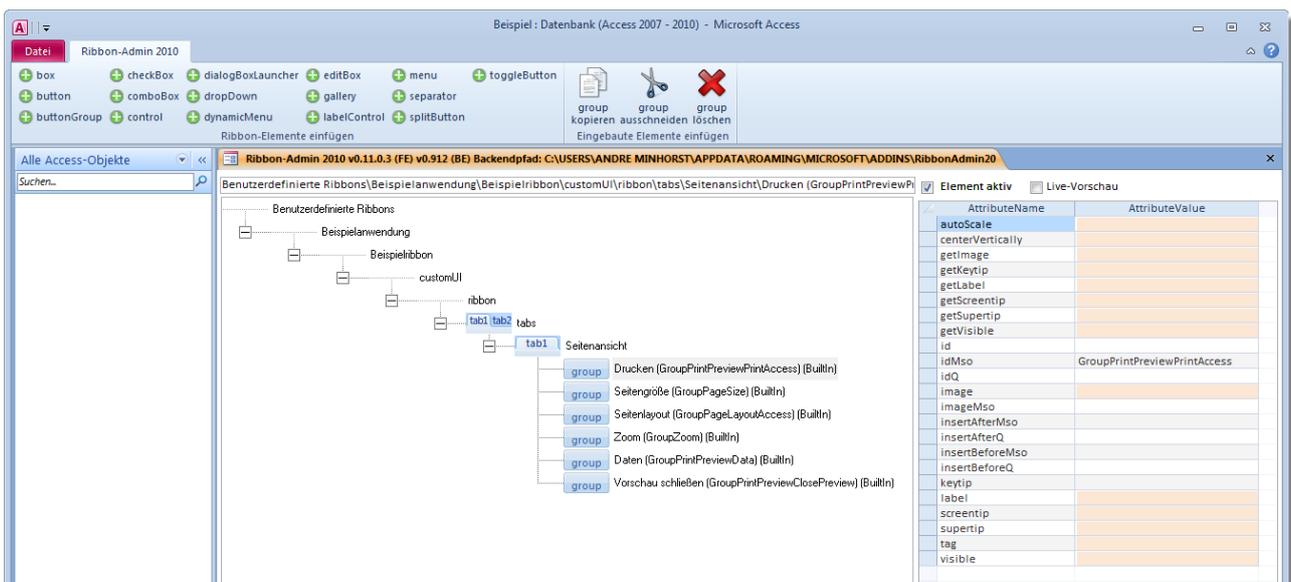


Bild 11: Ein durch einige **group**-Elemente abgebildetes eingebautes **tab**-Element

4.8 group-Element anlegen

Das **tab**-Element hält unter anderem einen Befehl zum Einfügen einer benutzerdefinierten Gruppe bereit. Nach dem Einfügen eines solchen Elements erscheint diese wie erwartet unterhalb des **tab**-Element. Im Ribbon und im Kontextmenü finden Sie nun die Möglichkeit zum Einfügen der eigentlichen Steuerelemente vor (siehe **Bild 12**).

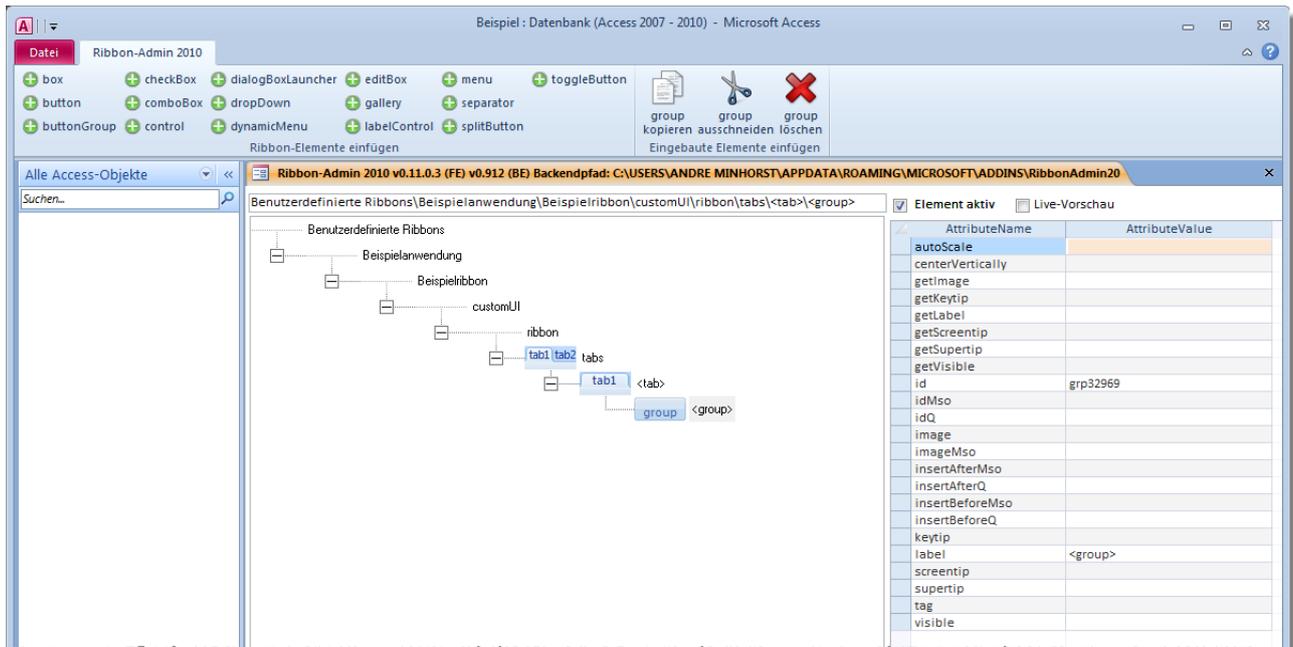


Bild 12: Unterhalb des **group**-Elements lassen sich eine ganze Reihe Steuerelemente einfügen.

4.9 Eingebautes group-Element einfügen

Genau wie bei den **tab**-Elementen können Sie auch eingebaute **group**-Elemente in die Ribbon-Definition einfügen. Dazu brauchen Sie nur den Befehl **Eingebaute group einfügen** des Ribbons oder des Kontextmenüs des jeweiligen **tab**-Elements aufzurufen. Im nun erscheinenden Dialog wählen Sie nacheinander das TabSet, das Tab und die Group aus (siehe Bild 13). Der **Ribbon-Admin 2010** legt dann ein neues **group**-Element an und trägt den englischen Namen der eingebauten Gruppe für das Attribut **idMso** ein.

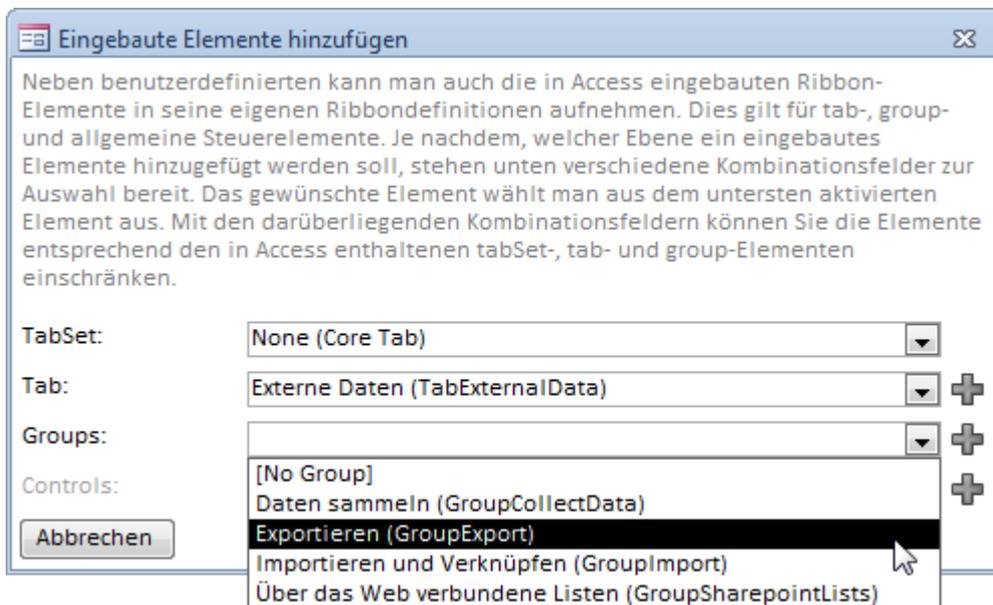


Bild 13: Hinzufügen eines eingebauten **group**-Elements

4.10 button-Element anlegen

Das wohl wichtigste Steuerelement im Ribbon ist das **button**-Element. Sie legen es über den entsprechenden Eintrag des Ribbons oder Kontextmenüs etwa des **group**-Elements an (es gibt noch weitere Elemente, denen Sie **button**-Elemente unterordnen können). Das **button**-Element erscheint wie in **Bild 14**.

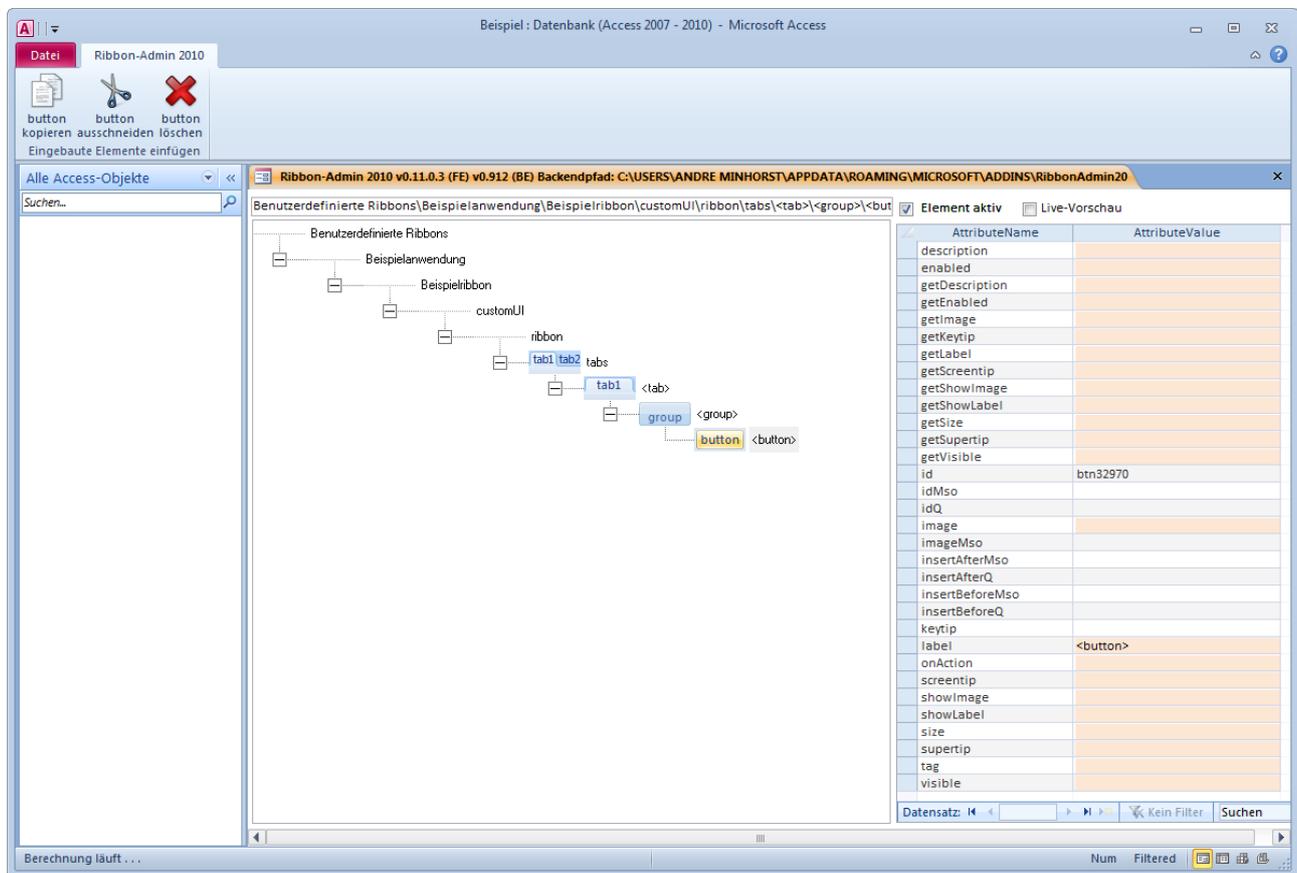


Bild 14: Der **Ribbon-Admin 2010** nach dem Anlegen eines **button**-Elements

Die Beschriftung des **button**-Elements können Sie direkt im TreeView ändern oder über das Attribut **label**. Die übrigen Attribute stellen Sie rechts im Attributbereich ein – einige Beispiele folgen.

4.10.1 VBA-Funktion zu **button**-Element zuweisen

Schaltflächen sollen VBA-Funktionen anstoßen. Damit ein mit dem **Ribbon-Admin 2010** erstelltes **button**-Element dies tut, tragen Sie für das Attribut **onAction** des Elements den Namen der VBA-Prozedur ein und legen diese entsprechend an. Dies erledigen Sie mit dem **Ribbon-Admin 2010** wie folgt:

- » Wählen Sie den Eintrag **Callbackfunktion anlegen** des Kontextmenüs des Attributs **onAction** des **button**-Elements an.
- » Gegebenenfalls wurde noch kein Modul namens **mdlRibbons** zum Speichern der Callback-Funktionen der Ribbons dieser Anwendung angelegt. In diesem Fall bestätigen Sie die Meldung aus **Bild 15** mit **Ja**.
- » Bestätigen Sie die nun erscheinende Meldung aus **Bild 16** mit der Schaltfläche **OK**.

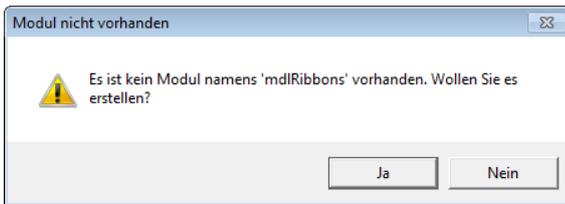


Bild 15: Falls das Modul **mdlRibbons** noch nicht vorliegt, wird es nun angelegt.

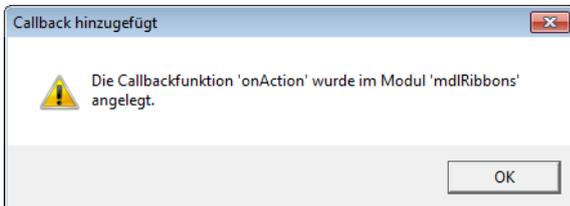


Bild 16: Die Callback-Funktion wurde erfolgreich angelegt.

Das **onAction**-Attribut enthält nun den Wert **onAction**. Im VBA-Editor finden Sie im Modul **mdlRibbons** eine entsprechende leere Prozedur vor (siehe **Bild 17**).

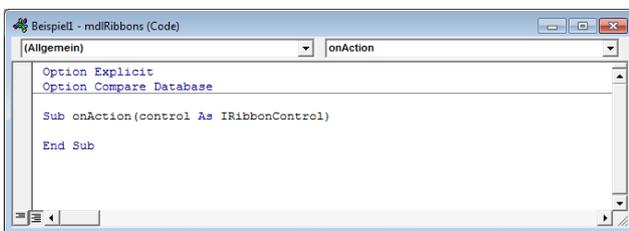


Bild 17: Die leere Prozedur **onAction** wird beim Anklicken eines **button**-Elements ausgelöst.

Ergänzen Sie diese Prozedur zu Testzwecken etwa wie folgt:

```
Sub onAction(control As IRibbonControl)
    MsgBox control.ID
End Sub
```

Ein Klick auf das **button**-Element sollte nun den Namen des Elements ausgeben (also den Wert des **id**-Attributs).

Nun ist ein passender Zeitpunkt gekommen, um sich das Aussehen des aktuellen Zustands des Ribbons einmal anzusehen und die Funktion zu testen. Wie dies funktioniert, erfahren Sie weiter hinten unter **Kapitel 7, Ribbon-Vorschau**.

4.10.2 button-Element mit einem eingebauten Bild versehen

Das **button**-Element können Sie mit einem eingebauten oder einem benutzerdefinierten Bild versehen. Für beides liefert der **Ribbon-Admin 2010** entsprechende Funktionen. Zum Einfügen eines eingebauten Bildes klicken Sie mit der rechten Maustaste auf das Attribut **imageMso**. Es erscheint der Dialog aus **Bild 18**. Mit dem Textfeld oben filtern Sie die Namen der angezeigten Bilder. Klicken Sie auf ein Bild und dann auf die Schaltfläche **OK**, um dieses für das aktuelle Steuerelement zu übernehmen. Der Name des Bildes wird dann für das Attribut eingetragen.

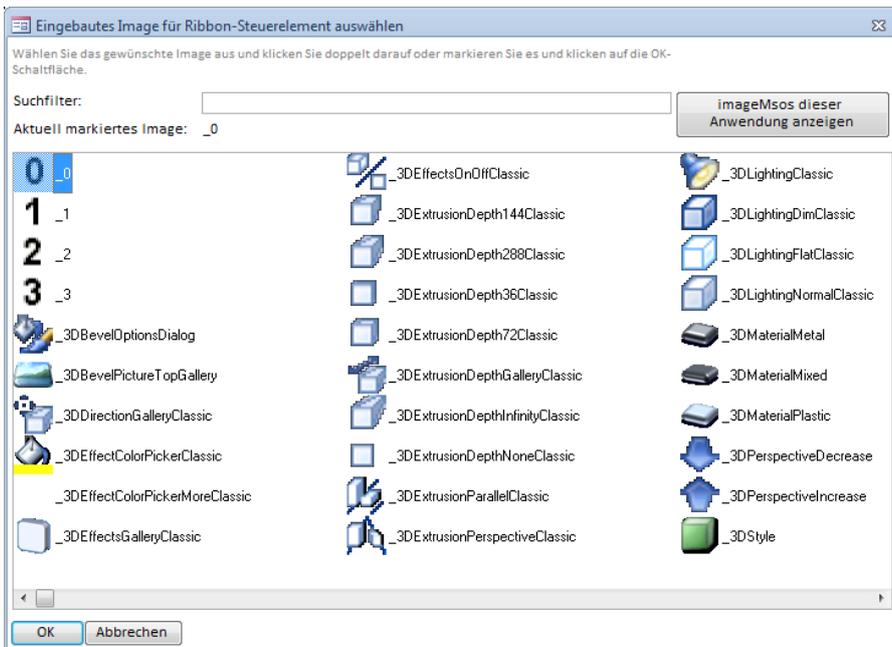


Bild 18: Auswählen eines eingebauten Bildes

Prüfen Sie die Einstellung in der Vorschau (siehe [Kapitel 7, Ribbon-Vorschau](#)).

4.10.3 button-Element mit einem benutzerdefinierten Bild versehen

Bei benutzerdefinierten Bildern ist der Aufwand etwas höher, aber nur für den **Ribbon-Admin 2010** – Sie brauchen ebenfalls nur ein paar Mausklicks.

Zunächst klicken Sie mit der rechten Maustaste auf das Attribut **image**. Wählen Sie den einzigen Eintrag des Kontextmenüs aus, um den Dialog aus **Bild 19** anzuzeigen.

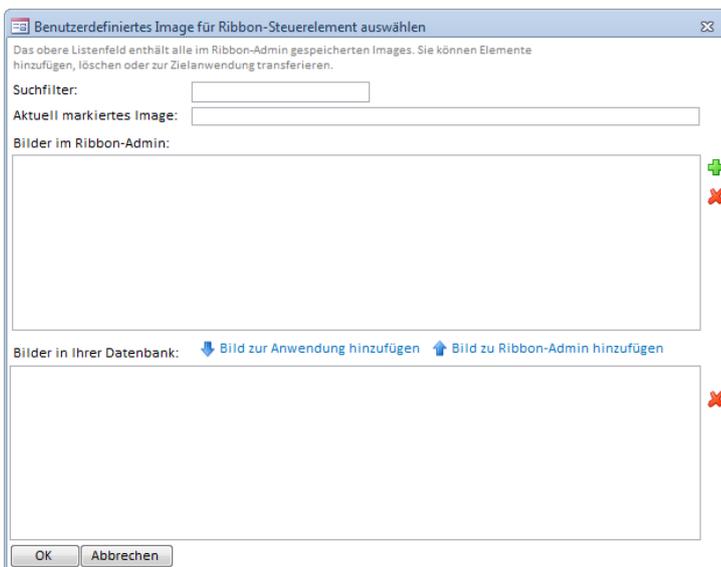


Bild 19: Verwaltung benutzerdefinierter Bilder im Ausgangszustand

Das Verfahren ist dreistufig:

- » Als erstes fügen Sie Bilder zum **Ribbon-Admin 2010** hinzu. Dazu klicken Sie auf das Plus-Zeichen rechts neben der oberen Liste und wählen die gewünschten Bilder aus.
- » Im zweiten Schritt übertragen Sie dann die gewünschten Bilder in die Zielanwendung. Dazu wählen Sie im oberen Bereich ein Bild aus und klicken auf den Befehl **Bild zur Anwendung hinzufügen**.
- » Schließlich wählen Sie eines der zur Anwendung hinzugefügten Bilder aus und weisen es mit einem Klick auf **OK** dem aktuell markierten Steuerelement zu.

Auf diese Weise sammelt sich im **Ribbon-Admin 2010** schnell eine Menge oft benötigter Bilder an, die Sie erstens schnell in Ihre Anwendungen einfügen und zweitens den Steuerelementen zuweisen können.

Als erstes lesen Sie also die Bilder ein. Diese werden in einer Tabelle im **Ribbon-Admin 2010** gespeichert, damit Sie unabhängig von der Quelle zur Verfügung stehen. Dazu klicken Sie auf die **Plus**-Schaltfläche rechts neben der oberen Liste. Es erscheint ein Dateiauswahl-Dialog, mit dem Sie eines oder mehrere Bilddateien auswählen können (siehe **Bild 20**).

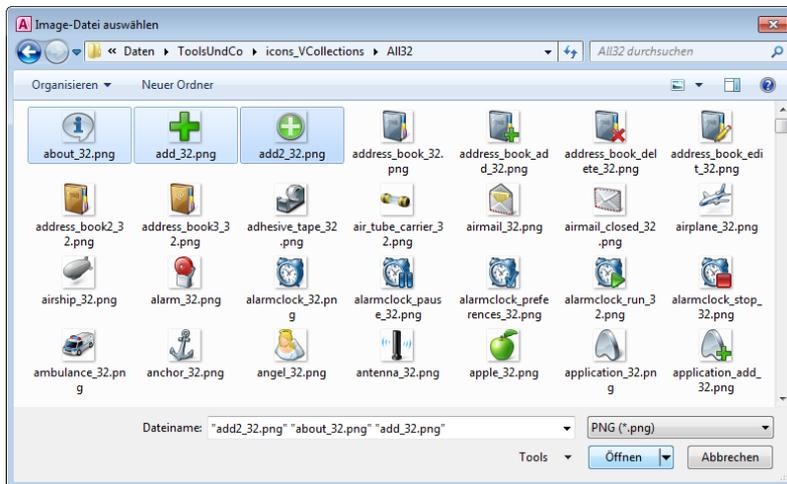


Bild 20: Auswahl der zu importierenden Bilder

Die ausgewählten Bilder erscheinen daraufhin in der oberen Liste des Dialogs aus **Bild 21**. Markieren Sie nun ein Bild und klicken auf den Befehl **Bild zur Anwendung hinzufügen**, erscheint dieses auch in der unteren Liste. Gleichzeitig wird es intern in der Systemtabelle **MSysResources** der Zielanwendung gespeichert. Das aktuell in der unteren Liste ausgewählte Bild wird oben im Textfeld angezeigt und beim Anklicken der **OK**-Schaltfläche für das Attribut **image** des Steuerelements eingetragen.

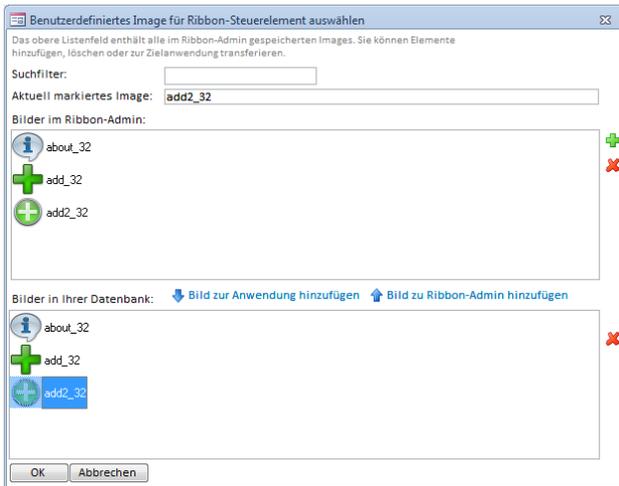


Bild 21: Hinzufügen von Bildern zur aktuell geöffneten Anwendung

Wenn Sie nun zu einem anderen Attribut wechseln und somit die Eingabe des Wertes für das Attribut **image** bestätigen, erscheint eine Meldung, die Sie gegebenenfalls auf noch durchzuführende Schritte hinweist (siehe **Bild 22**).

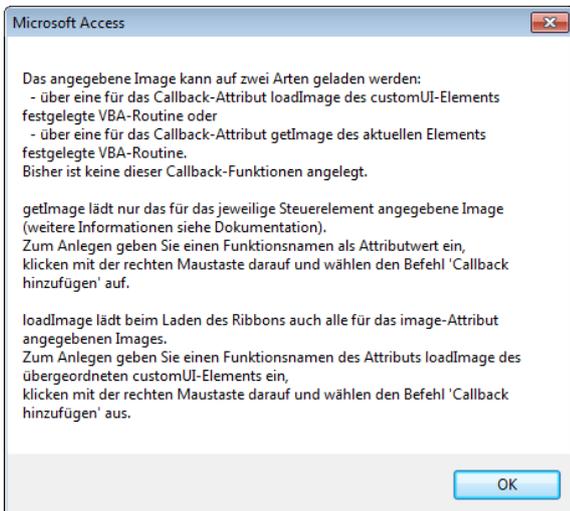


Bild 22: Meldung mit Hinweisen auf weitere Schritte zur Anzeige von Bildern im Ribbon

4.10.4 Callback zum Anzeigen von Bildern hinzufügen

Damit ein Bild im Ribbon angezeigt wird, benötigen Sie nämlich noch ein paar Zeilen VBA-Code. Diese fügt der **Ribbon-Admin 2010** für Sie hinzu. Dies sind die notwendigen Schritte, die Sie selbst durchführen müssen:

- » Klicken Sie auf das **customUI**-Element der aktuellen Ribbon-Definition.
- » Wählen Sie den Eintrag **Callback hinzufügen** des Kontextmenüs des Attributs **loadImage** aus (siehe **Bild 23**).

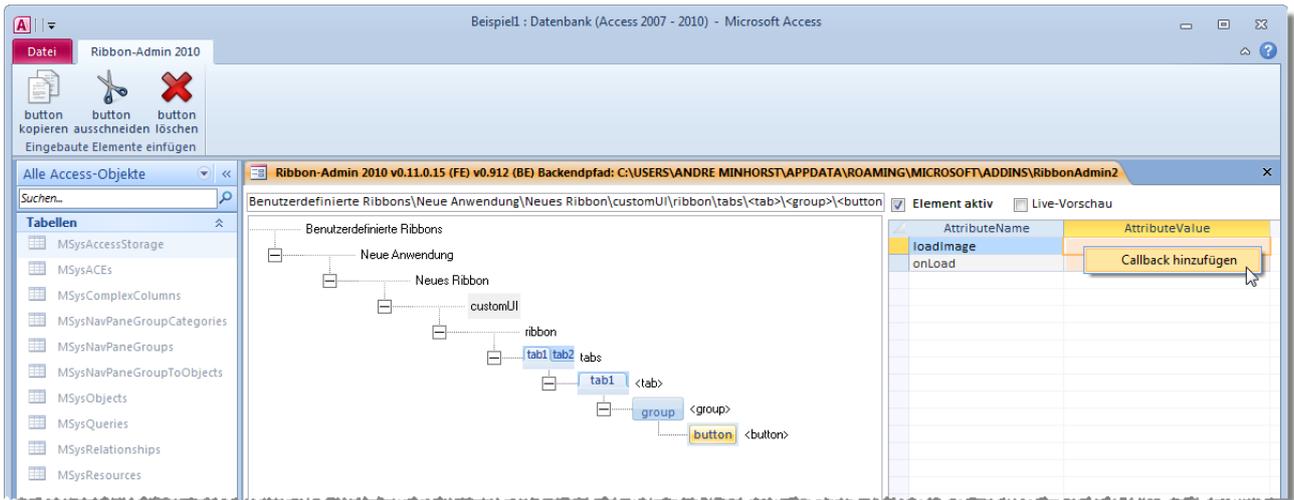


Bild 23: Hinzufügen einer Callback-Funktion für das Attribut **loadImage**.

Gegebenenfalls erscheint eine Meldung, die darauf hinweist, dass für die Anzeige von Bildern im Ribbon ein Verweis auf die Bibliothek **Microsoft Office 14.0 Object Library** nötig ist (siehe **Bild 24**).

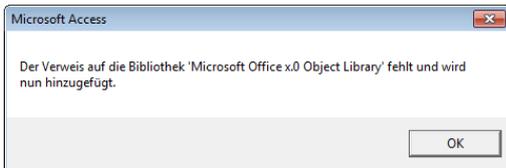


Bild 24: Die Office-Library wird automatisch hinzugefügt.

Außerdem legt der **Ribbon-Admin 2010** ein Modul namens **mdlRibbons** an, das alle durch den **Ribbon-Admin 2010** erstellten Callback-Funktionen und Variablen aufnimmt (siehe **Bild 25**).

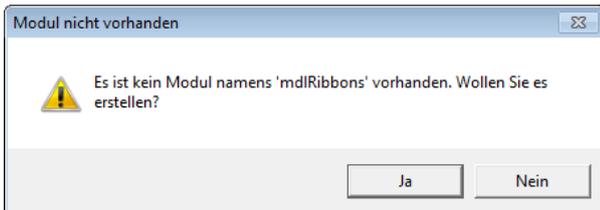


Bild 25: Anlegen des Moduls **mdlRibbons**

Danach vermeldet der **Ribbon-Admin 2010** noch das erfolgreiche Anlegen der Callbackfunktion **loadImage** im Modul **mdlRibbons** (siehe **Bild 26**).

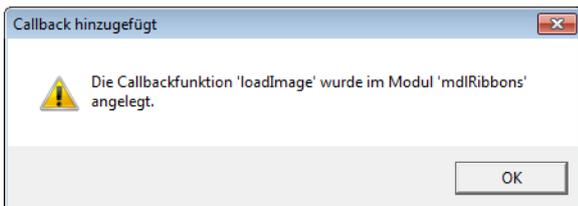


Bild 26: Hinzufügen der Callback-Funktion **loadImage**

Die Callbackfunktion **loadImage** ruft ihrerseits eine weitere Prozedur auf, die der **Ribbon-Admin 2010** in einem eigenen Modul namens **mdlRibbonImages** speichert (siehe **Bild 27**).



Bild 27: Anlegen des Moduls **mdlRibbonImages** mit Funktionen zum Anzeigen von Bildern im Ribbon

Stellen Sie noch das Attribut **size** des **button**-Elements auf den Wert **large** ein und betrachten Sie dann das Ribbon in der Vorschau (siehe **Kapitel 7, Ribbon-Vorschau**). Das Ergebnis sollte wie in **Bild 28** aussehen.

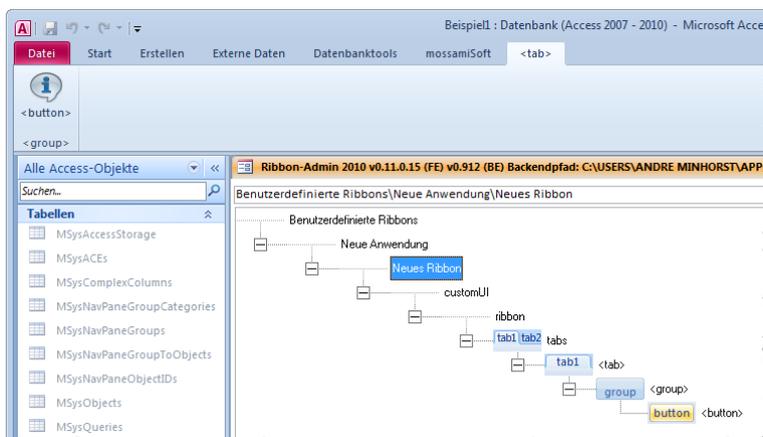


Bild 28: Ein **button**-Element mit Bild in der Vorschau.

4.11 Weitere Steuerelemente hinzufügen

Der **Ribbon-Admin 2010** zeigt die jeweils verfügbaren Steuerelemente im Ribbon beziehungsweise im Kontextmenü an. Sie fügen diese Steuerelemente genau wie das **button**-Steuerelement über die entsprechende Schaltfläche zur Ribbon-Definition hinzu und stellen ihre Attribute ein.

4.12 Eingebaute Steuerelemente hinzufügen

Genau wie bei den **tab**- oder **group**-Elementen können Sie mit dem **Ribbon-Admin 2010** auch eingebaute Steuerelemente zu Ihrer Ribbon-Definition hinzufügen. Dazu verwenden Sie den Eintrag **Eingebautes control einfügen** des Ribbons oder des Kontextmenüs des jeweiligen **group**-Elements oder weiterer Steuerelemente, die das Unterordnen von Elementen erlauben. Es erscheint der bereits bekannte Dialog aus **Bild 29**, der diesmal allerdings vier Schritte bis zur Auswahl des gewünschten Steuerelements erfordert.

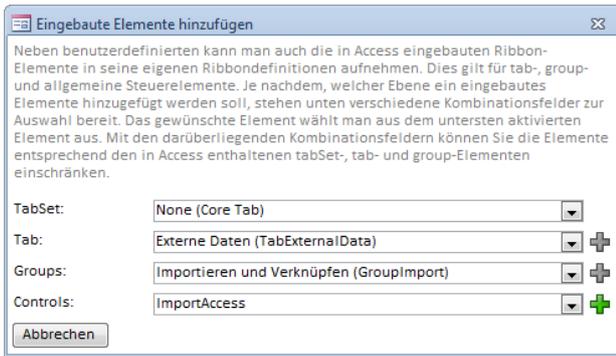


Bild 29: Hinzufügen eines eingebauten Steuerelements

Alternativ können Sie auch zuerst ein Steuerelement anlegen und dann den Eintrag **Eingebaute idMso hinzufügen** des Kontextmenüs des Attributs **idMso** aufrufen.

4.13 Elemente aktivieren und deaktivieren

Nicht immer möchten Sie Elemente gleich löschen, wenn Sie diese einmal nicht benötigen oder etwas daran nicht funktioniert. Selektieren Sie einfach das betroffene Element und deaktivieren Sie die Option **Element aktiv** über dem Attributbereich. Das Element wird dann zwar weiterhin im **Ribbon-Admin 2010** angezeigt, aber nicht mehr in die XML-Definition des Ribbons integriert.

4.14 Reihenfolge von Elementen ändern

Die Reihenfolge von Elementen des gleichen übergeordneten Element ändern Sie mit folgenden Tastenkombinationen:

- » **Strg + Nach oben:** Vertauscht das Element mit dem darüber liegenden Element.
- » **Strg + Nach unten:** Vertauscht das Element mit dem darunter liegenden Element.

Achtung: Das Vertauschen ist nicht immer möglich. Einige Elemente müssen in einer bestimmten Reihenfolge angegeben werden.

5 Ribbon in die Zielanwendung schreiben

Wenn Sie die Ribbon-Definition fertiggestellt haben oder diese in der Zielanwendung testen möchten, wählen Sie das Root-Element der jeweiligen Definition aus und betätigen Sie den Befehl **Ribbon in Zielanwendung** aus dem Ribbon oder dem Kontextmenü zu diesem Element.

Der **Ribbon-Admin 2010** speichert die Ribbon-Definitionen in der Tabelle **USysRibbons** der Zieldatenbank. Diese muss gegebenenfalls noch angelegt werden, was sich durch die Meldung aus **Bild 30** äußert.

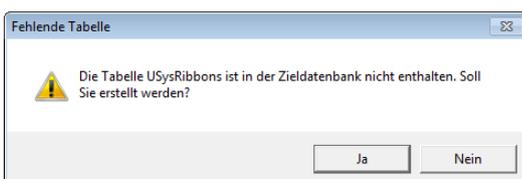


Bild 30: Wenn noch keine Tabelle namens **USysRibbons** vorhanden ist, wird diese nun angelegt.

Diese Tabelle sehen Sie im Navigationsbereich nur, wenn Sie die Anzeige von Systemobjekten aktiviert haben (Rechtsklick auf die Titelleiste des Navigationsbereichs, Eintrag **Navigationsoptionen...**, dort die Option **Systemobjekte anzeigen** aktivieren). Diese Tabelle enthält je ein Feld für den Namen der Ribbon-Definition sowie für den resultierenden XML-Ausdruck (siehe **Bild 31**).

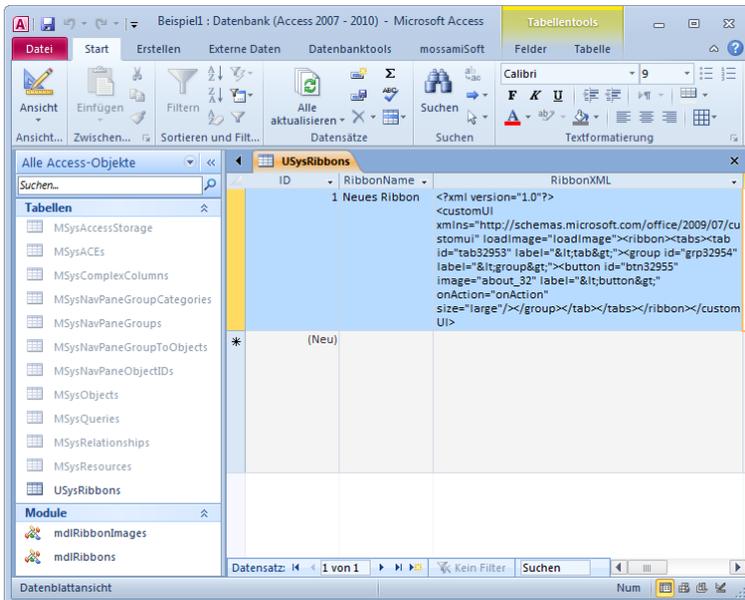


Bild 31: Eine Ribbon-Definition in der Tabelle **USysRibbons** der Zieldatenbank

6 Ribbon zur Anzeige festlegen

Ribbons werden aus zwei Anlässen angezeigt:

- » wenn Sie als Anwendungsribbon festgelegt sind oder
- » wenn Sie als Ribbon eines Formulars oder Berichts festgelegt sind.

6.1 Ribbon als Anwendungsribbon festlegen

Wenn Sie ein Ribbon als Anwendungsribbon definieren möchten, aktivieren Sie das Root-Element der Ribbondefinition im **Ribbon-Admin 2010** und wählen Sie den Befehl **Ribbon als Anwendungsribbon** aus (siehe **Bild 32**).

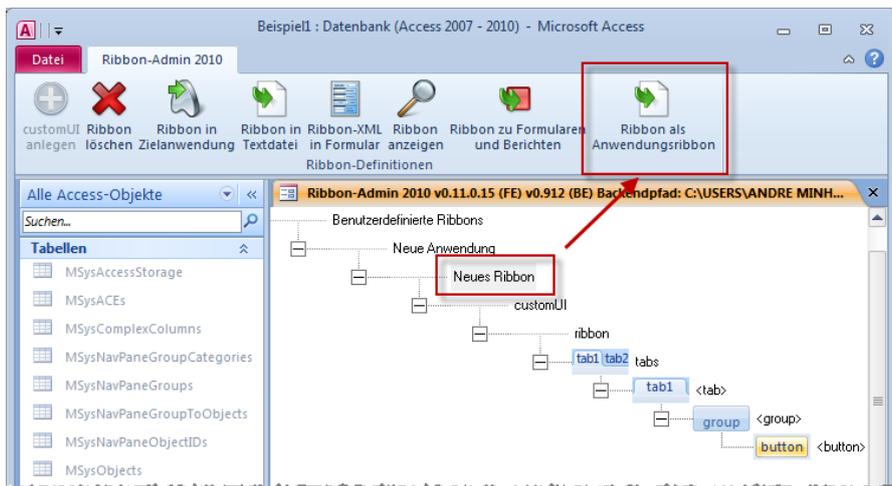


Bild 32: Ribbon als Anwendungsribbon festlegen

Das Ribbon wird nun in die Eigenschaft **Name des Menübandes** in den Access-Optionen eingetragen. Diese Option können Sie auch selbst einstellen oder kontrollieren (siehe **Bild 33**).

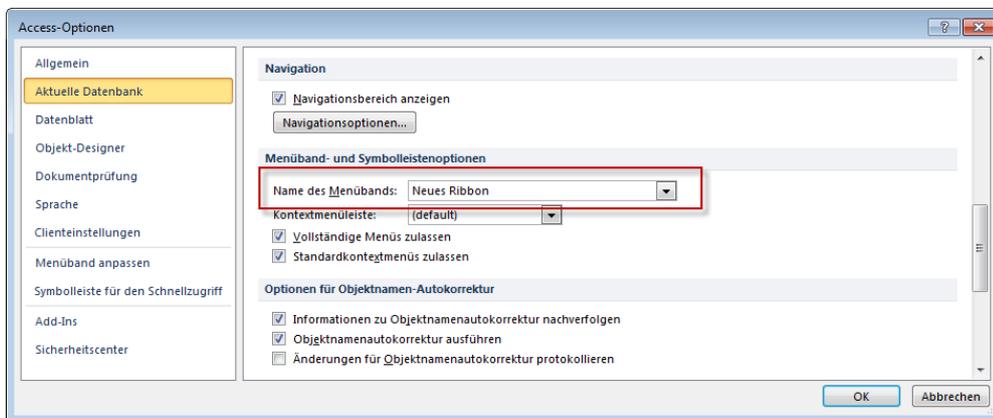


Bild 33: Diese Einstellung nimmt der **Ribbon-Admin 2010** für Sie vor.

Eine so festgelegte Ribbon-Definition wird beim nächsten Start der Anwendung angezeigt.

6.2 Ribbon zu Formular oder Bericht zuweisen

Sie können auch dafür sorgen, dass Ribbon-Definitionen in Zusammenhang mit Formularen oder Berichten angezeigt werden. Dazu tragen Sie den Namen der Ribbon-Definition aus der Tabelle **USysRibbons** für die Eigenschaft **Name des Menübands** des Formulars und gegebenenfalls auch des Unterformulars ein.

Diese Funktion wird in einer späteren Version des **Ribbon-Admin 2010** implementiert werden.

7 Ribbon-Vorschau

Die Ribbon-Vorschau können Sie auf zwei Arten hervorrufen:

- » Eine Live-Vorschau, die alle Änderungen gleich abbildet, erhalten Sie, wenn Sie die Option **Live-Vorschau** über dem Attributbereich aktivieren. Achtung: Die Ribbon-Befehle zum Bearbeiten der Ribbon-Definition werden dann nicht mehr angezeigt. Das Bearbeiten muss somit mit den entsprechenden Kontextmenü-Befehlen erfolgen.

- » Eine aktuelle Vorschau findet statt, wenn Sie den Befehl **Ribbon anzeigen** des Root-Elements der Ribbon-Definition ausführen (siehe **Bild 34**).

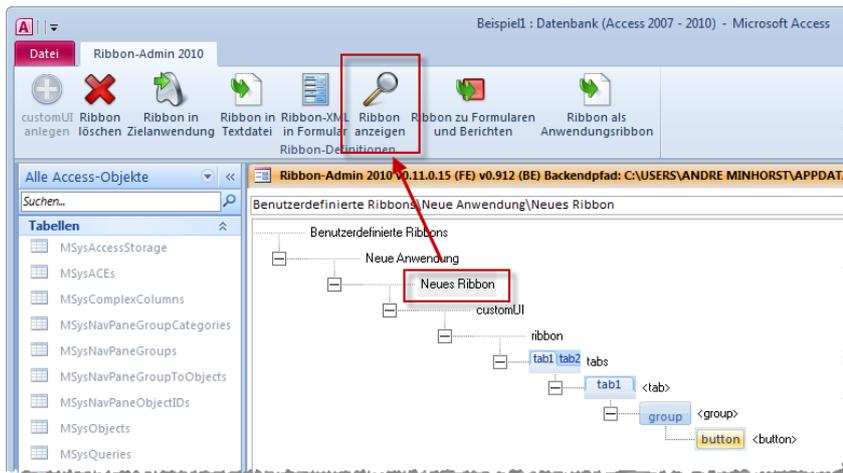


Bild 34: Der Aufruf dieses Befehls zeigt das aktuelle Aussehen des Ribbons an.

7.1 Fehler bei der Vorschau

Bei der Vorschau kommt es zur Anzeige von Fehlermeldung, wenn beispielsweise Pflichtelemente oder -attribute nicht vorhanden oder ungültig sind. In diesem Fall ändern beziehungsweise ergänzen Sie die Ribbon-Definition entsprechend und zeigen die Vorschau erneut an.