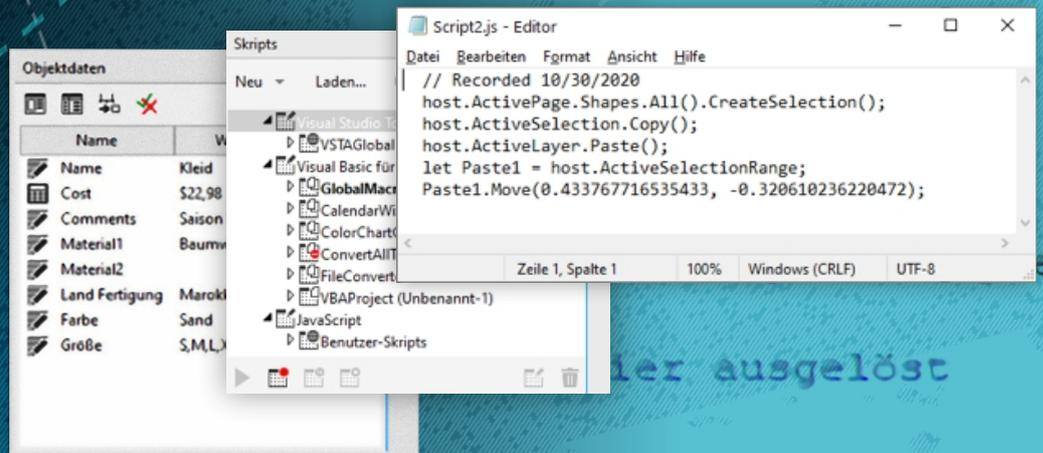


Shortcuts mit Skripts

Workflows automatisieren
mit CorelDRAW



Über den Trainer

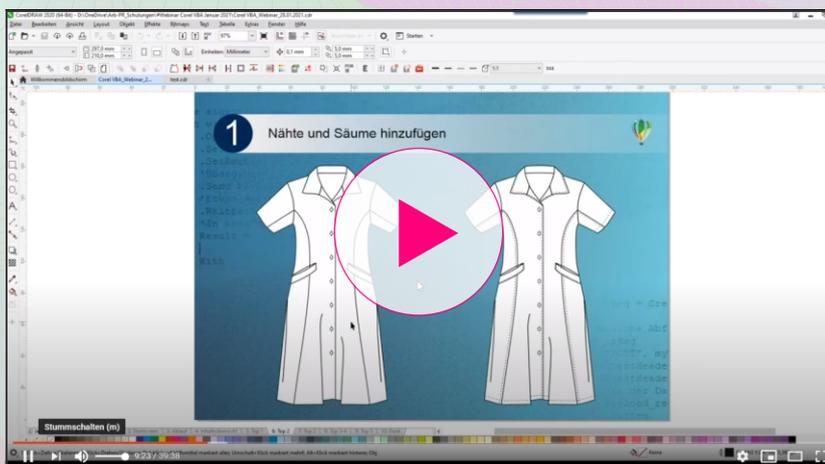
Onno Strenge

Onno Strenge ist CorelDRAW-Plug-in-Entwickler und Qualitätsmanager. Als langjähriger Mitinhaber einer Entwicklerfirma ist Onno Strenge mit Corel Script und der Skriptsprache VBA auf Du-und-Du. Seit Jahrzehnten teilt er sein Wissen in Lehrgängen und Webinaren. Sein Anliegen ist es, auch „Endnutzern“ die Benutzerfreundlichkeit der fantastischen technischen Möglichkeiten von Skripten (= Makros) aufzuzeigen und ihnen deren Nutzung für die tägliche Arbeit zu ermöglichen.



Shortcuts mit Skripts. Workflows automatisieren mit CorelDRAW

Videotraining hier direkt laden.



Weitere
Videotrainings
und E-Books zum
Herunterladen
finden Sie unter

<https://www.coreldraw.com/de/learn/webinars/>

Inhalt

01	Warum Makros/Skripts?	4
02	Makro-/Skript-Programmierung und Programmiersprachen	5
03	Praktische Anwendungen für Makros	6
04	Makros aufzeichnen	7
05	Einbinden des Makros in die Oberfläche	9
06	Makros und Mehr	10
	• Arbeiten im Scripteditor	10
	• Bearbeiten von Makros	11
	• Hilfe-Funktionen	11
07	Vorbereitungen in CorelDRAW & Sicherheit	12
08	Beispiel eines komplexeren Makros: Das Forschungsprojekt „PreQA“	13
09	Fragen & Antworten	14

Impressum

V. i. S. d. P.: Jörg Rieger, Corel GmbH
Redaktion: Peter Knoll, Katrin Richthofer
Layout und Satz: Michaela Maginot
Bildnachweis/Grafiken: Onno Strenge, Ulli Strenge,
Peter Lütke-Wissing (Titelseite)

01 | Warum Makros/Skripts?

In diesem E-Book bekommen Sie eine Kurzeinführung in die VBA-Nutzung am Beispiel von CorelDRAW 2020 für Windows. Sie lernen Möglichkeiten der Automatisierung in Corel-Programmen kennen und bauen Berührungspunkte ab. Denn auch SIE können schnell und einfach ein eigenes Makro (= Skript) erstellen, das Ihnen hilft, CorelDRAW an Ihre ganz spezifischen Arbeitsabläufe anzupassen und diese zu automatisieren.

Effizienzsteigerung ist DAS Stichwort dieses E-Books. Sie können nahezu alle Funktionen des Programms mit den Makros/Skripts verwenden und diese anschließend in Ihren Arbeitsbereich integrieren, ja sogar mit einem eigenen Tastaturkürzel versehen und so das Makro mit einem Klick starten.

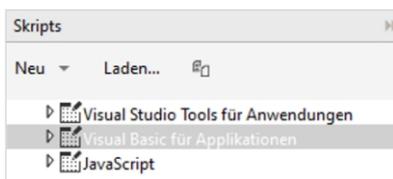
Es ist nicht nur möglich, eine Funktion innerhalb einer geöffneten Zeichnung zu automatisieren. Sie können mit Makros auch automatisiert Dateien öffnen, bearbeiten und diese – wenn gewünscht – auch in einem anderen Dateiformat speichern. Das ermöglicht weitere Möglichkeiten, Änderungen an einer Reihe von Verzeichnissen vorzunehmen – als Mode-Designer können Sie beispielsweise alle CDR-Dateien in einem Saison-Verzeichnis und den Unterverzeichnissen mit einem Klick von der Winterkollektion 2020 ins Design der Frühjahrskollektion 2021 überführen, einen bestimmten Text suchen und durch neuen Text ersetzen (neue Adresse oder neue Saison) oder z. B. alle Linien von 1 Punkt Strichdicke auf 2 Punkt anpassen.

02 | Makro-/Skript-Programmierung und Programmiersprachen

Die Möglichkeit, Makros/Skripts aufzuzeichnen und zu programmieren, steht Ihnen in einer Reihe von Corel-Applikationen zur Verfügung: In CorelDRAW, der CorelDRAW Technical Suite (inkl. DESIGNER) und in Corel PHOTO-PAINT. Es ist möglich, die einmal erstellten Makros in anderen Corel-Applikationen zu verwenden – solange die ausgeführten Werkzeuge im anderen Programm auch verfügbar sind. Ein Import aus der CorelDRAW Technical Suite in die CorelDRAW Graphics Suite ist daher immer problemlos möglich, während sie umgekehrt eventuell keinen Zugriff auf die erweiterten Werkzeuge der CorelDRAW Technical Suite haben.

Auch Office-Produkte wie Word und Excel sind mit VBA programmierbar. Das eröffnet viele weitere Möglichkeiten, nicht nur Daten, sondern auch Teile der VBA-Makros zwischen den CorelDRAW-Anwendungen und den Office-Anwendungen auszutauschen. Das trifft für die Dialoge sowie für den VBA-Code zu, solange es keine Corel-spezifischen Funktionen sind.

Um die verschiedenen Skriptsprachen zu sehen, die Ihnen CorelDRAW unter Windows oder macOS zur Verfügung stellt, öffnen Sie das **Andock-/Informationsfenster > Skripts**.



Unter Windows stehen Ihnen mit VST, VBA und JavaScript drei Skriptsprachen zur Verfügung, unter macOS nur JavaScript.

VST: Visual Studio Editor (ALT + Umschalt + F12)

Visual Studio wendet sich an fortgeschrittene Programmierer. Hier ist die Installation von Microsoft Visual Studio 2019 mindestens in der Version „Community“ erforderlich. Das Programm benötigt mindestens 15 GB Platz auf der Festplatte. Es läuft nur unter Windows.

VBA: Visual Basic for Application (ALT + F11)

Zur Nutzung dieser Option ist keine spezielle zusätzliche Installation notwendig, weswegen wir uns auf diese Option konzentrieren werden. VBA läuft nur unter Windows, ist aber JavaScript so ähnlich, dass auch macOS-Nutzer viele Informationen 1:1 übertragen können.

Tipp:

In CorelDRAW, Corel PHOTO-PAINT und im Corel DESIGNER ist derselbe VBA-Editor wie in den Microsoft Office-Applikationen integriert. Haben Sie bereits Makros für Excel oder Word programmiert, werden Sie sich daher sehr schnell in dieser Umgebung zurechtfinden. Die Struktur aller Makros ist sich sehr ähnlich – es lohnt sich, das Konzept einmal zu verstehen, dann können Sie es auf allen Plattformen anwenden.

JavaScript

JavaScript läuft unter Windows und unter macOS. Zur Nutzung dieser Option ist keine spezielle zusätzliche Installation notwendig.

Corel liefert für die Windows-Version bereits ein paar Makros mit aus. Der Speicherort ist **C:\Program Files\Corel\CorelDRAW Graphics Suite 2020\Draw\GMS**.

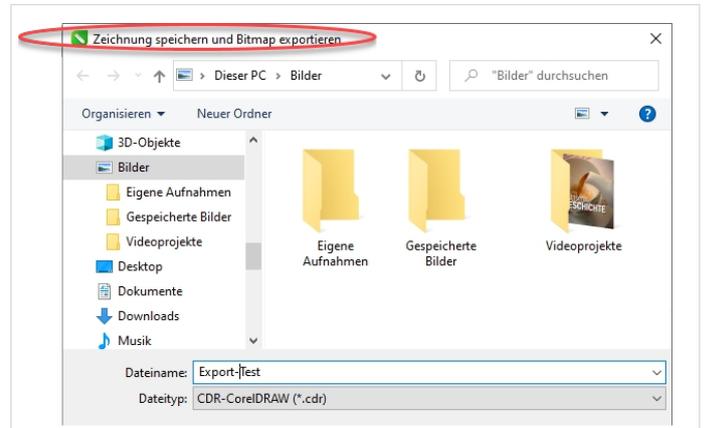
Zur Kommunikation mit externen Servern nutzen Sie WinHttpRequest.

Das bietet Ihnen nahezu unbegrenzte Möglichkeiten eines Makros, der Daten-Im- und Export wird im Kapitel 8 dieses E-Books kurz angerissen.

03 | Praktische Anwendungen für Makros

Ein oft verwendetes, praktisches Beispiel aus dem Plug-in „Corel for Fashion“ ist die PDM-Funktion: Wenn Sie eine Zeichnung als CorelDRAW-Original und als JPEG-Datei – eventuell sogar noch in verschiedenen Auflösungen und Größen für verschiedene Social Media Plattformen – speichern wollen, müssen Sie viele Schritte nacheinander ausführen: **Speichern unter > Eingabe neuer Name > Modifikationen abspeichern**, dann Exportieren in das gewünschte Format und Verzeichnis, mit dem erneuten Anklicken aller Optionen. Stattdessen können Sie all diese Schritte für den Export eines Bitmaps in PDM- (Produkt-Daten-Management), PLM- (Produkt-Lifecycle-Management) oder ERP- (Enterprise-Resource-Planning) Systemen einmal aufzeichnen und als Makro mit einem Mausklick aufrufen: Die Zeichnung wird sofort wie voreingestellt exportiert.

Wenn Sie nicht exakt die identische Aktion immer wieder ausführen lassen wollen, können Sie mit Komfort-Abfragen Ihr Makro flexibler gestalten. So definieren Sie z. B. eine Dialogbox, in die Sie beim Aktivieren Daten interaktiv eingeben können – erst mit der Bestätigung der Eingabe wird die eigentliche Aktion ausgeführt.



In einem Rutsch: Mit einem Makro/Skript geben Sie Ihre Datei via Knopfdruck in unterschiedlichen Dateiformaten aus.

04 | Makros aufzeichnen

Ein neues Makro können Sie mit Klick der rechten Maustaste auf die jeweilige Programm-Umgebung erstellen.

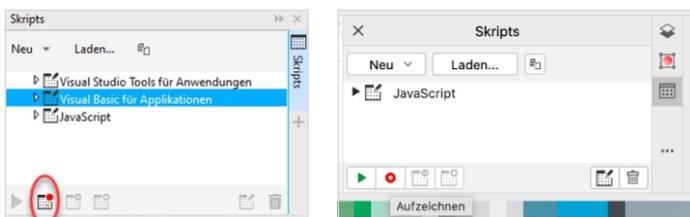
Achtung: Die Datei wird standardmäßig in den persönlichen Ordner abgelegt: **C:\Users\
AnwenderName\AppData\Roaming\
Corel\CorelDRAW Graphics Suite
2020\Draw\GMS**. Dieses Verzeichnis ist für gewöhnlich versteckt und nicht sichtbar, was ein Problem sein kann, wenn Sie mit mehreren Menschen einen Computer nutzen oder in verschiedenen Verzeichnissen arbeiten.

Tipp:

Speichern Sie Ihre Makros/Skripts – alternativ zur standardmäßigen Speicherung im persönlichen Ordner – in **C:\Program Files\Corel\CorelDRAW Graphics Suite 2020\Draw\GMS**, sie sind dann auch für andere User sichtbar.

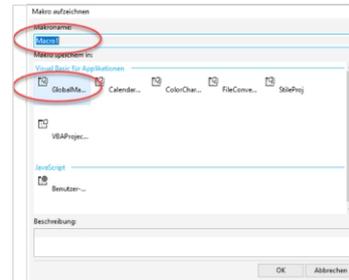
Öffnen Sie zunächst Ihre Datei. Um mit VBA und Java ein kleines Makro aufzuzeichnen, starten Sie die Aufzeichnung mit **Extras > Skripts > Aufzeichnung** (roter Knopf unten im Fenster) oder mit **Fenster > Andockfenster/Informationsfenster > Skripts** (Klick auf die entsprechende Schaltfläche).

Unten rechts im Fenster sehen Sie verschiedene Symbole, die aktiv werden, sobald Sie ein Makro gestartet haben.



Das **Skripts**-Fenster mit dem roten Aufnahme-Button unter Windows und macOS im Vergleich

Über das rote Schaltfeld **Aufzeichnen** starten Sie die Aufnahme eines Skripts.



Im Arbeitsfenster können Sie einen **Makronamen** vergeben und (unter Windows) unter verschiedenen **Skript-Optionen** wählen.

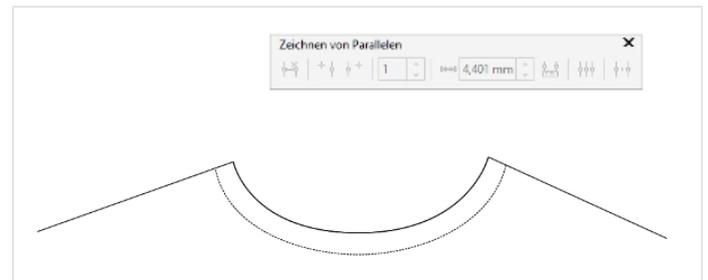


Mit den Schaltfeldern unten können Sie die **Aufzeichnung anhalten...**



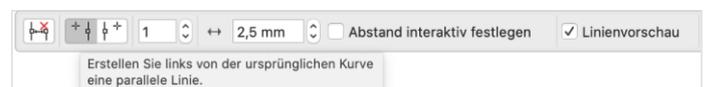
... oder die **Aufzeichnung stoppen**.

Eine konkrete, in verschiedenen Bereichen oft verwendete Funktion ist das Einfügen einer zweiten Linie parallel zu bestehenden Linien: Sie stellt in der Textilindustrie verschiedene Nähte dar, im Landschaftsbau Raseneinfassungen, in der Straßenverkehrssicherung Sicherheitsstreifen etc.



In der Schaltfläche **Zeichnen von Parallelen** können Sie verschiedene Parameter festlegen.

Parallelen können Sie aktivieren, wenn Sie eines der Zeichenwerkzeuge ausgewählt haben – also z. B. das Hilfsmittel **Stift**. Klicken Sie rechts oben auf die Schaltfläche **Zeichnen von Parallelen** um die **Eigenschaftsleiste fürs Zeichnen von Parallelen** zu aktivieren.



All die Einstellungen fürs Parallelenzeichnen können Sie in einem Makro automatisieren – oder gezielt die veränderlichen Werte in einem Abfragefeld des Makros abfragen.

Unter Windows klicken Sie alternativ mit der rechten Maustaste in die Symbolleiste und aktivieren Sie im Dropdownmenü **Zeichnen von Parallelen**.

Markieren Sie ihr Objekt. Mit einem Doppelklick auf das Objekt oder alternativ unter Windows über **F10** öffnen Sie das Hilfsmittel **Form**. Markieren Sie alle Knotenpunkte, die zur Linie gehören.

Starten Sie jetzt erst die Aufzeichnung des Makros!

Über **Strg/Befehl + C**, **Strg/Befehl + V** kopieren Sie die Punkte in den Zwischenspeicher und bestätigen Sie mit der **Leertaste** (= Auswahlwerkzeug). Nun legen Sie in der Eigenschaftsleiste fürs **Zeichnen von Parallelen** die gewünschte Parallele fest, also z. B. **Parallele Linie rechts**, Anzahl: **1**, Abstand: **1,5** cm. Mit markierter Linie ändern Sie – wenn gewünscht – im Bearbeitungsfenster oben die Eigenschaft der neu gezeichneten Parallele, z. B. **Linienstärke** und/oder **Linienstil**. So haben Sie Ihre Linie erzeugt und ihr einen Stil gegeben. Beenden Sie nun die Makro-Aufnahme.



Wenn Sie alle Nähte eines Kleides oder alle Blumenbeete mit einer identischen parallelen Linie doppeln wollen, lohnt sich ein Aufzeichnen aller Auswahlkriterien in einem Makro.

05 | Einbinden des Makros in die Oberfläche

Nicht nur die Hilfe bei einer Routinetätigkeit, sondern auch das einfache Aufrufen dieser Hilfe ist wichtig. Wie so oft gibt es auch hier viele Wege, das neu erstellte Makro in den Arbeitsbereich zu integrieren – insbesondere, wenn Sie Windows-UserIn sind.

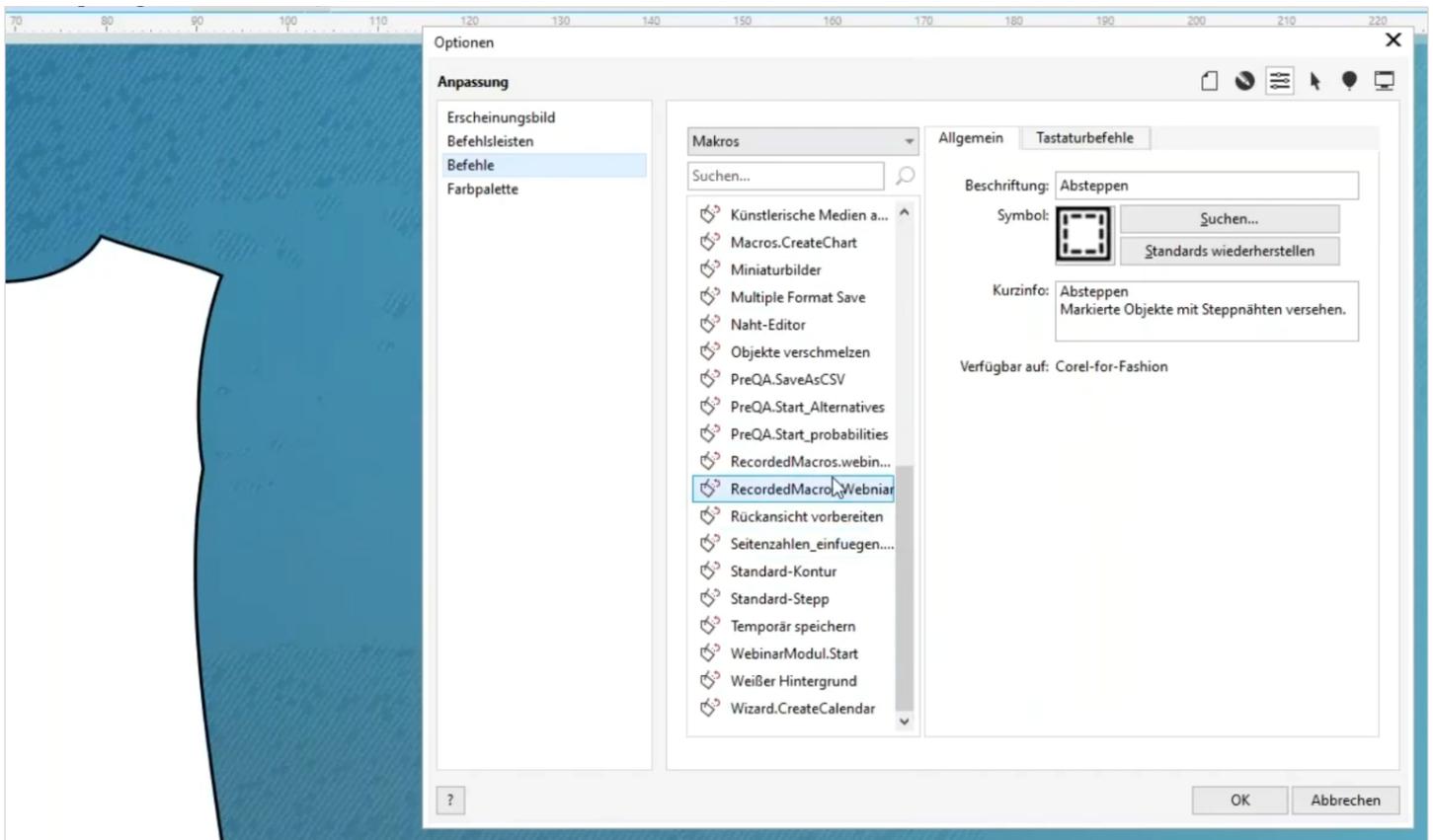
Sie können beispielsweise ein Icon in die Oberfläche integrieren. Symbole müssen das Format *.ico haben. Vorbereitend können Sie ein kleines Symbol in Corel PHOTO-PAINT erstellen, das thematisch zu dem neu erzeugten Makro passt – hier also z. B. einfach zwei parallele Linien, in der Größe 16 x 16 oder 32 x 32 Pixel. Speichern Sie Ihr eigenes Symbol im Format *.ico.

Klicken Sie mit der **rechten Maustaste** in die gewünschte Symbolleiste, um sie anzupassen. Einen neuen Befehl ergänzen Sie unter Windows über **Anpassen > Eigenschaftsleiste-Symbolleiste >**

Neuen Befehl hinzufügen oder über **Extras > Optionen > Anpassen > Befehle**.

In diesem Fenster können Sie zunächst im Drop-Down-Menü zu **Recorded Makros scrollen**, diese auswählen und das gewünschte Makro markieren. Editieren Sie es nach Wunsch und legen Sie ein **Symbol** für diesen Befehl fest, indem Sie über **Suchen** Ihr vorbereitetes Symbol anwählen.

Sie können Ihrem Makro auch einen **Neuen Tastaturbefehl** zuordnen: Wenn Sie den Reiter **Tastaturbefehle** wählen und nun einen freien Befehl – z. B. **Strg + Shift + W** – eingeben, ist diese Kombination als Kürzel für dieses Makro festgelegt. Sollte der Befehl schon vergeben sein, informiert Sie das Programm. Über **Drag & Drop** können Sie Ihren neu erstellten Befehl in die Oberfläche einfügen.



Unter den **Optionen** der Eigenschaftsleiste definieren Sie unter Windows **Befehle** und **Symbole**.

06 | Makros und Mehr

Nun haben Sie, ohne ein einziges Wort Code geschrieben zu haben, eine Funktion programmiert. Auf diese Weise können Sie beliebig viele wiederkehrende Arbeitsschritte aufzeichnen und sich viele lästige Klicks ersparen. Wenn Sie immer exakt diese zweite parallele Linie brauchen oder nur andere ganz klar definierte Aufgaben automatisieren wollen, reicht das vollkommen aus.

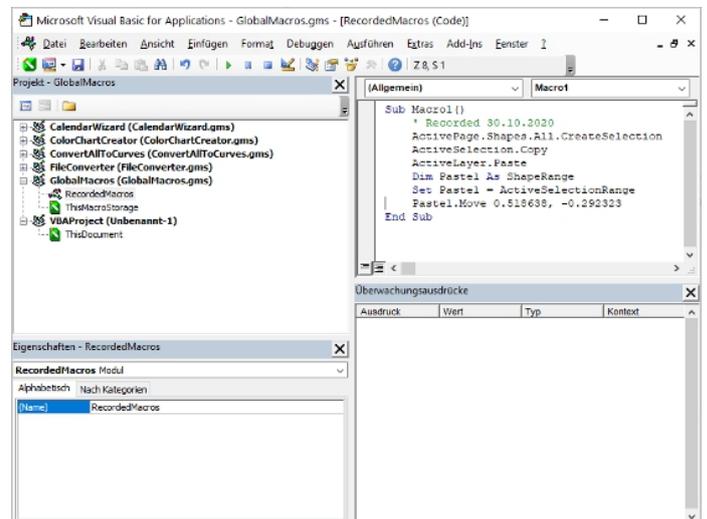
In unserem Makro haben wir die Annahme getroffen, dass das richtige Segment vor dem Starten des Makros markiert wurde – hier könnte Sie das Programm, statt an irgendeiner zufällig noch markierten Stelle des Dokuments eine Parallellinie einzufügen, mit einer Nachrichtenbox informieren, falls kein Objekt im aktiven Fenster markiert ist. Über eine **Message-Box** können Sie mit dem Anwender kommunizieren – und so häufige Fehler vermeiden. Sie haben mit den **Form-Werkzeugen** die Möglichkeit, **User Forms** hinzuzufügen und eigene Dialog-Boxen zu erstellen, mit denen Sie mit dem Anwender kommunizieren können. Über das **Beschriftungs-Feld** können Sie einfache Informationen eingeben, unten links sehen Sie die verschiedenen Optionen der Steueroptionen:

- **Caption** ist der Text, der angezeigt wird.
- Sie können weitere Optionen hinzufügen und zwischen Code und Objekt hin- und herschalten.
- Im **Textfeld** können Sie Text und Zahlen abfragen.
- **Checkfelder** sind Kontrollkästchen.
- In **Optionfeldern** können Sie zwischen verschiedenen Optionen wählen.
- Die **Befehls-Schaltfläche** ist eine Schaltfläche, die der Benutzer wählen kann, um einen Befehl auszuführen – typischerweise „OK“ oder „Abbrechen“.

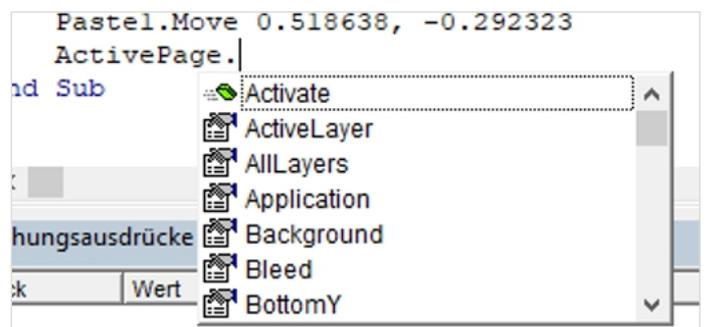
Dies ist eine Sammlung von Möglichkeiten und Optionen bei der Arbeit mit Makros – wie geht das nun in der Praxis?

Arbeiten im Scripteditor

Um mit Makros noch etwas mehr machen zu können, beispielsweise Komfortabfragen und Interaktion, müssen Sie nun im nächsten Schritt in den Code gehen. Unter **Extras > Skript > Scripteditor** finden Sie unter Windows links im **Editorfenster** eine Übersicht über alle Projekte, rechts den eigentlichen Editor und die verschiedenen Makros. Unten im Fenster haben Sie die Möglichkeit der Ausgabe von Zahlen und Daten.



Der **VBA-Scripteditor** bietet links neben dem eigentlichen Editor eine Übersicht über alle Projekte.



Im Editor werden die Elemente der Klassen automatisch angezeigt.

Bearbeiten von Makros

Im **Scripteditor** können Sie sich das aufgenommene Skript anzeigen lassen und es auch erweitern. Um das aufgezeichnete Makro zu erweitern, markieren Sie im VBA-Makro z. B. **ActiveSelectionRange** und verwenden Sie ggf. **F1**, um so in die **API-Dokumentation** zu gehen und sich anzeigen zu lassen, welche Optionen Sie mit diesem Befehl haben.

```
' Recorded 28.01.2021
Dim OrigSelection As ShapeRange
Set OrigSelection = ActiveSelectionRange
OrigSelection.Copy
Dim pasteopt As StructPasteOptions
Set pasteopt = CreateStructPasteOptions
With pasteopt.ColorConversionOptions
    .SourceColorProfileList = "sRGB IEC61966-2.1,ISO
    .TargetColorProfileList = "sRGB IEC61966-2.1,ISO
End With
Dim Pastel As ShapeRange
Set Pastel = ActiveLayer.PasteEx(pasteopt)
Pastel.ApplyNoFill
Pastel.CreateParallelCurves 1, -0.059055
Pastel(1).Style.StringAssign ("fill":{"overprint":
"5","rateAcceleration":"0","yScale":
ActiveLayer.Shapes(1).Style.StringAssign ("fill":{"
"5","rateAcceleration":"0","yScale":
```

Auf den ersten Blick erscheint der Editor etwas verwirrend. Im Vergleich zu Chinesisch ist diese Sprache aber ein Klacks zum Erlernen, versprochen! **ActiveSelectionRange** ist z. B. ein Corel-eigener Befehl.

Tip:

Die **Hilfe** Funktion (in VBA **F1**) ist unter Windows nicht nur beim Einstieg in den Scripteditor eine große Hilfe für alle des Englischen Mächtigen – benutzen Sie diese Option ruhig oft: Ihre Frage zu einer Funktion wurde sicher schon einmal gestellt. Unter den **SDK** und **VBA-Beispielen** finden Sie ganz konkrete Beispiele, die Sie mit **Strg + C / Strg + V** herauskopieren und im eigenen Makro weiterverwenden können.

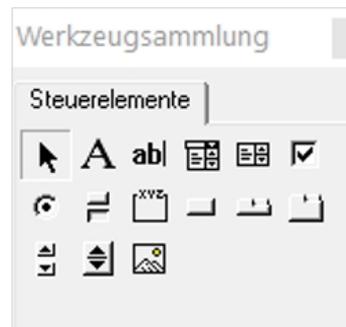
Bisher liegen die Makros im Scripteditor in **RecordedMakros** bzw. in **Scripts**. Hier können Sie diese Makros auch mit einem Mausklick starten.

Nun wollen Sie Ihr aufgezeichnetes Makro eventuell erweitern, indem Sie z. B. ein paar Komfort-Abfragen einbauen.

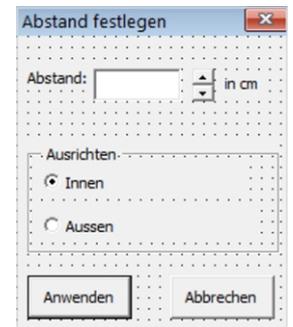
```
'Zunächst einige Sicherheitsabfragen
If Application.Documents.Count = 0 Then
    'Ist überhaupt eine Datei geöffnet?
    MsgBox sMessageKeineDateiGeoeffnet, vbOKOnly, sTitelKeineDateiGeoeffnet
Exit Sub
End If
```

Programmcode für einfache Dialogboxen – Message Boxen (**MsgBox**)

Auch verschiedene Dialogboxen mit Eingabemöglichkeiten können Sie einfügen, Textboxen erstellen, Texte als solche definieren, Textfarben festlegen etc. Das Programm liefert dazu im **Scripteditor** über **Einfügen > Userform** eine einfache **Werkzeugsammlung** zur Erstellung von aktiven Elementen wie z. B. Dialogboxen.



Die **Werkzeugsammlung** bietet bereits eine vordefinierte Auswahl an interaktiven Elementen.



Die Parameter einer **Dialogbox** können Sie flexibel festlegen.

Hilfe-Funktionen

Die **CoreIDRAW Community Developer Area** ist eine hilfreiche Quelle für Fragen zur Makro-Programmierung. Hier können Sie sich das Skript anzeigen lassen und viele Textblöcke direkt übernehmen – sehr hilfreich!

API Documentation > CoreIDRAW > 2020 > Application > IVGApplication VBA

Application.ActiveSelectionRange property
Gets the active selection range

Syntax:
Property Get ActiveSelectionRange() As ShapeRange

Remarks:
The ActiveSelectionRange property returns the active ShapeRange object in the active document.

Examples:
The following VBA example displays messages about selected objects in the active document, depending on the number of objects selected.

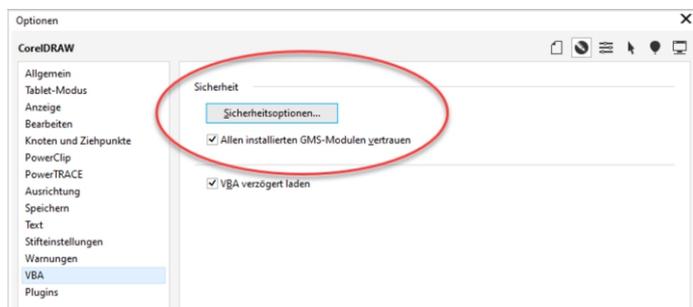
```
Sub SelectionRangeActive()
    If ActiveSelectionRange.Count = 1 Then
        MsgBox "There is only one shape selected in the shape range."
    ElseIf ActiveSelectionRange.Count = 0 Then
        MsgBox "There are no shapes selected in the shape range."
    Else
        MsgBox "There are " & ActiveSelectionRange.Count & " shapes in the selected shape range."
    End If
End Sub
```

In der **API Dokumentation** finden Sie ausführliche Informationen, teilweise mit kurzen Code-Beispielen.

07 | Vorbereitungen in CoreIDRAW & Sicherheit

Um mit Makros arbeiten zu können, braucht es nicht viel:

- Sie sollten idealerweise beim Start des Programmes NICHT – wie vielleicht sonst gewohnt – **Makros deaktivieren** anklicken.
- Als weitere Vorab-Aktion können Sie – wenn auf Ihrem Rechner schon viel mit Makros gearbeitet wurde, die Sie nie nutzen – Ihr GMS-Verzeichnis leeren. Öffnen Sie dazu das **Andock-/Informationsfenster > Skripts**, klicken Sie mit der rechten Maustaste auf das Makro, das Sie entfernen möchten und klicken Sie dann auf **Makroprojekt entfernen**. Belassen Sie nur die Build-In Makros im Verzeichnis.
- Um die Sicherheit bei der Verwendung von Makros zu gewährleisten, rufen Sie über **Extras > Optionen > VBA** die Sicherheitsoptionen auf. Hier können Sie die Einstellungen gemäß Ihrer Firmenpolitik anpassen.



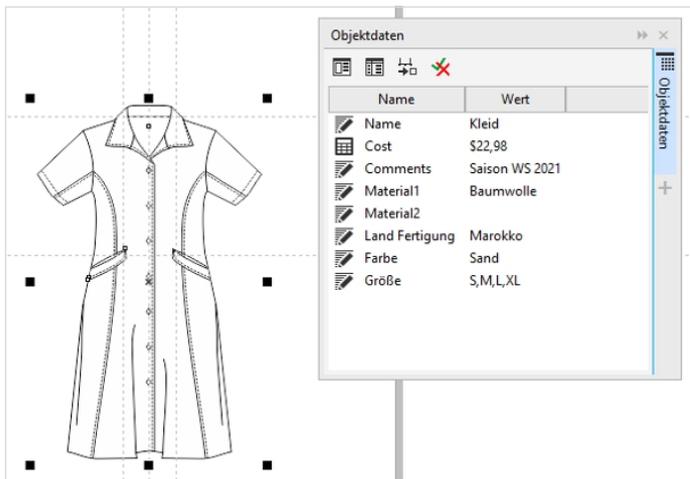
Achten Sie darauf, die Sicherheitsoptionen entsprechend Ihrer Firmenpolitik anzupassen.

Wenn Sie eine CoreIDRAW-Datei bekommen, kann diese schon Makros enthalten. Als Standardeinstellung werden Sie gefragt, ob Sie diese Makros aktivieren wollen. Entscheiden Sie selbst, wie vertrauenswürdig die Quelle ist, aus der Sie das Dokument bekommen.

08 | Beispiel eines komplexeren Makros: Das Forschungsprojekt „PreQA“

Sie können nun eigene Makros aufnehmen, editieren, und interaktive Elemente einfügen und definieren. Wo kann die Reise weiter hingehen?

Sehen wir uns ein komplexes Beispiel an: Die Firma CDS ist an einem vom Bundesforschungsministerium finanzierten internationalen Forschungsprojekt beteiligt. „Preventive Quality assurance – Vorsorgende Qualitätssicherung“ ist das Thema: Je früher ein potentieller Fehler erkannt wird, desto besser lassen sich Folgekosten vermeiden. Im „Preventive QA Forschungsvorhaben“ wird von einem interdisziplinären Konsortium aus Forschungs- und Industriepartnern ein System zur präventiven Qualitätssicherung im kreativen Produktdesign der Modebranche entwickelt. Potentielle Produktfehler sollen bereits beim Design detektiert werden. Methoden des maschinellen Lernens analysieren dafür die Retouren-Daten der Produktion – also beispielsweise Fehler, die gehäuft auftreten, wenn Nähte zu nah nebeneinander liegen.



In CorelDRAW können Sie ganz einfach **Objektdaten** auslesen und an andere Systeme exportieren.

Die Produktmerkmale lassen sich aus Entwürfen über die Objektdaten (die sie im **Andockfenster/ Informationsfenster Objektdaten** finden) und den Objektdaten-Manager in CorelDRAW auslesen und an einen externen Server leiten.

```
Set winHttpRequest = CreateObject("WinHttp.WinHttpRequest.5.1")

'Die eigentliche Abfrage wird hier ausgelöst
With winHttpRequest
  .Open "POST", myURL, True
  .SetRequestHeader "Content-Type", "application/json"
  .SetRequestHeader "Connection", "keep-alive"
  'Übergabe der Daten
  .Send Payload_request
  'Etwas warten
  .WaitForResponse 4000
  'In Results werden die Rückmeldungen des Assistenz-System gespeichert
  Result = .ResponseText
End With
```

Über **Skripte** kann CorelDRAW mit externen Servern kommunizieren.

Über Excel oder ein EAP-System ermittelt das Assistenzsystem der Kollegen aus den Forschungszentren die Wahrscheinlichkeiten von Fehlerkosten: Die Daten werden dazu ausgelesen, ein externer Server vergleicht die Objektdaten dieses Designs mit denen anderer, offenbar fehleranfälliger Designs und warnt den Anwender, wenn erforderlich. So lassen sich die Informationen aus den Retourendaten bereits ins neue Design übertragen, potenzielle Produktfehler frühzeitig entdecken und auftretende Folgekosten vermeiden.

09 | Fragen & Antworten

Kann man programmübergreifend zwischen verschiedenen Corel-Programmen aufzeichnen? Konkretes Fallbeispiel z. B. Auswahl einer Datei in CorelDRAW > Export als PNG-Datei > Kopieren in PHOTO-PAINT > dann Exportieren mit einer Transparenz?

Skripte können leider nicht programmübergreifend aufgezeichnet werden: Man kann aus CorelDRAW direkt Corel PHOTO-PAINT aufrufen, aber dann verlässt man die Umgebung und muss an dieser Stelle mit einem neuen Makro ansetzen.

Welche Funktionen lassen sich in CorelDRAW nicht aufzeichnen!?

Geschätzt 90 Prozent der Funktionen lassen sich aufzeichnen, die anderen 10 Prozent muss man händisch programmieren. Nicht als Makro/Skript aufzeichnen können Sie insbesondere:

- Filterfunktionen, die über die Plug-in-Schnittstelle eingebunden sind
- Objektdaten lassen sich nicht automatisch auslesen, dies kann über die Programmierung geschehen.

Ist VBA integraler Bestandteil von CorelDRAW?

VBA und VSTA sind in der CorelDRAW Graphics Suite Windows-Version automatisiert enthalten und lassen sich über **F11** aufrufen. In der macOS-Version und in CorelDRAW Essential und Standard sind diese Funktionen nicht enthalten.

Wie stelle ich sicher, dass meine Makros bei einem Abo-Update in die neue Version mitgenommen werden?

Im CorelDRAW-GMS-Verzeichnis sind alle Makros enthalten – diese können Sie über Copy & Paste in die neue Version übernehmen. Genauso können Sie das Verzeichnis auch an die Rechner von Kollegen übertragen, wenn Sie die entsprechenden Schreibrechte haben. Speichern Sie gemeinsam genutzte Makros in einen Shared Folder, vor dem Start von CorelDRAW sollten andere Nutzer die Makros in ihr eigenes GMS-Verzeichnis übertragen. Die Übernahme von Veränderungen der Symbolleiste funktionieren über das Exportieren der Arbeitsbereiche.

Fragen zu praktischen Arbeitsbeispielen:

Kann ich einen in einem Makro programmierten Abstand nachträglich ändern?

Ja, Sie können das resultierende Ergebnis einer mit einem Makro erzeugten Datei frei bearbeiten. Alternativ können Sie beim Ausführen des Makros über ein Textfeld die gewünschten Werte jeweils abfragen.

Funktioniert das proportionale Füllen eines Bitmaps in einem Powerclip über Makros?

Aktuell unterstützt Corel dies noch nicht. Sie erhalten die Fehlermeldung: "Recording of this command is not supported: PowerclipContentOperation".

Kann ich für eine Druckvorlage für die möglichst platzsparende Anordnung verschiedener Entwürfe Makros verwenden?

Für das Programmieren von Makros hilft immer die Frage: Wie würde ich das händisch machen? Diese Vorgehensweise können Sie dann aufzeichnen und programmieren. Dafür ist das Nutzen und Umschreiben bereits bestehender Makros oft zielführend. Zum Automatisieren sollten Sie in diesem Fall zunächst den einzelnen Bereichen Namen und Eigenschaften zuweisen, in Ihren Entwürfen automatisiert die Namen finden und Sequenzen hineinkopieren. In CorelDRAW gibt es unter Windows ein voreingestelltes Makro zum Erstellen von Kalendern, dieses Makro lässt sich für diese Anwendung eventuell abwandeln.

Wie funktioniert das Importieren und Exportieren vorhandener Makros konkret?

Sie können neue Makros auf der Programmebene einzufügen: Alle Programme liegen unter Windows in **C:\Program Files\Corel\CorelDRAW Graphics Suite 2020\Draw\GMS**. Alles, was dort liegt, wird nach einem Neustart von CorelDRAW im Programm angezeigt und kann dort durch Doppelklick gestartet werden.

Wie wurde die PDM-Funktion in „Corel for Fashion“ fürs Speichern und Exportieren erstellt?

Dieses Makro übernimmt zwei Speicheraufgaben in einem Rutsch. Es wurde erstellt über **Datei > Exportieren**, dann wurde das Makro angehalten, im Editor geschaut, der vergebene explizite Dateiname gekennzeichnet und das Verzeichnis programmiert.



Die **PDM-Funktion** ist eines der vielen **Corel for Fashion** Makros.

Über **F1** bekommen Sie unter Windows viele Code-Parameter erklärt, sehr oft hilft es auch, Bestandteile aus bestehendem Code zu verwenden und abzuwandeln. Neben den in CorelDRAW enthaltenen Makros können dafür auch Makro-Plug-ins eine reiche Quelle sein. „Corel for Fashion“ bietet z. B. nicht nur ModedesignerInnen eine Programmierung vieler immer wiederkehrender Arbeitsschritte, auch für die Makro-Programmierung in anderen Bereichen ist es eine Quelle durchdacht programmierter Makros für verschiedenste Anwendungen.

Hier der Code der Export-Funktion aus der CorelDRAW Hilfe:

```
Sub Test()
  Dim d As Document
  Dim s As Shape
  Dim Filter As ExportFilter
  Set d = CreateDocument
  Set s = d.ActiveLayer.CreateEllipse2(4, 5, 2)
```

Hier wird eine Ellipse gezeichnet, damit man etwas exportieren kann.

```
s.Fill.ApplyFountainFill CreateRGBColor(255, 0, 0),
CreateRGBColor(0, 0, 0)
```

Diese Ellipse wird hier gefüllt.

```
Set Filter = d.ExportBitmap("C:\Temp\Doc.jpeg",
cdrJPEG, cdrCurrentPage, cdrRGBColorImage, 0, 0,
72, 72)
With Filter
  .Compression = 80
  .Optimized = True
  .Smoothing = 50
  .SubFormat = 1
  .Progressive = False
  .Finish
End With
End Sub
```

Kontakt

Weitere Informationen und Downloads
aller Testversionen unter:

www.coreldraw.com/business

Fachliche Fragen:

Peter Knoll, corel@4edition.com

