

5 Simple SOLIDWORKS Macros Every Design Engineer Should Know

📅 Date	April 21, 2026
≡ Topic	SolidWorks macros for design engineers

Copy these, run them today. No programming degree required.

By Prasath | ScriptedCAD | <https://scriptedcad.com>



Want to go beyond macros and build real SolidWorks automation tools? I teach the full SolidWorks API path at ScriptedCAD. scriptedcad.com

Every design engineer I know has spent hours doing the same thing in SolidWorks over and over. Export this drawing. Update that title block. Rebuild and save before the release. Again. And again.

Here is the part nobody told you: SolidWorks has a built-in macro recorder and a full API underneath. You can automate almost any repetitive action. Some macros need no code at all. The ones that do need code are 10 to 20 lines of VBA you can copy, paste, and run in under 5 minutes.

These are the 5 macros I wish someone had handed me on day one.



A macro is a recorded or written script that tells SolidWorks what to do. You run it once. It does the work. SolidWorks supports VBA macros out of the box with no extra software needed.

Macro 1: Record and Replay Any Action

This is the entry point. No code. No API knowledge. Just record what you do and play it back.

How to do it:

1. Go to Tools > Macro > Record
2. Perform any action in SolidWorks (apply a view, insert a note, change a display state)
3. Click Stop and save the file as a .swp macro
4. Run it again with Tools > Macro > Run

Real scenario: You apply the same isometric view with shaded-with-edges display every time you send a screenshot to a customer. That is 6 clicks every time. Record it once. Now it is 1 click.

You can assign macros to a toolbar button or keyboard shortcut in Tools > Customize > Toolbars. That turns any recorded sequence into a single key press.



The recorder does not capture every UI click. It works best for operations on geometry or model data. Test your recorded macro on a separate file before relying on it.

Macro 2: Save the Active Drawing as PDF

Ask any design engineer what they do 10 to 20 times a day. Exporting drawings to PDF is near the top. This macro saves the active drawing as a PDF in the same folder. No dialog boxes. No clicking through the save menu.

```
Sub SaveAsPDF()  
    Dim swApp As SldWorks.SldWorks  
    Dim swModel As SldWorks.ModelDoc2  
    Dim sFilePath As String  
    Dim errors As Long  
    Dim warnings As Long  
  
    Set swApp = Application.SldWorks  
    Set swModel = swApp.ActiveDoc
```

```

If swModel Is Nothing Then
    MsgBox "No active document found."
    Exit Sub
End If

sFilePath = Left(swModel.GetPathName, Len(swModel.GetPa
thName) - 7) & ".pdf"
swModel.Extension.SaveAs sFilePath, 0, 0, Nothing, erro
rs, warnings

MsgBox "Saved: " & sFilePath
End Sub

```

To use this: go to Tools > Macro > New, paste the code, save it. Then assign it to a keyboard shortcut in Tools > Customize. One key press and the PDF is done.



The macro builds the PDF path by replacing the .SLDDRW extension with .pdf. The output lands in the same folder as the drawing file with no extra setup.

Macro 3: Update Custom Properties on a Part

Title blocks pull their data from custom properties: part number, description, revision, material. If you are filling these in manually on every file you open, you are losing 5 to 10 minutes per part.

This macro sets the three most common properties on the active document in one run. Change the values in the code to match your part before running.

```

Sub UpdateCustomProperties()
    Dim swApp As SldWorks.SldWorks
    Dim swModel As SldWorks.ModelDoc2
    Dim swCustPropMgr As SldWorks.CustomPropertyManager

```

```

Set swApp = Application.SldWorks
Set swModel = swApp.ActiveDoc
Set swCustPropMgr = swModel.Extension.CustomPropertyManager("")

swCustPropMgr.Add3 "PartNo", swCustomInfoText, "PN-001", swCustomPropertyReplaceValue
swCustPropMgr.Add3 "Description", swCustomInfoText, "Bracket Assembly", swCustomPropertyReplaceValue
swCustPropMgr.Add3 "Revision", swCustomInfoText, "A", swCustomPropertyReplaceValue

MsgBox "Custom properties updated."
End Sub

```



Once custom properties are set, your title block updates automatically. If your company uses a standard set of properties across all files, you can build this macro into a template workflow that runs every time a new part is created.

Macro 4: Batch Export All Drawings in a Folder to PDF

The single-file PDF macro is useful. But what happens when you need to export 50 drawings before a review meeting that starts in 30 minutes?

This macro opens every .SLDDRW file in a folder you specify, exports it as PDF, and closes it. What used to take 30 to 40 minutes of clicking now runs in the background while you prepare your slides.

```

Sub BatchExportToPDF()
    Dim swApp As SldWorks.SldWorks
    Dim swModel As SldWorks.ModelDoc2
    Dim sFolder As String
    Dim sFile As String

```

```

Dim sPDFPath As String
Dim errors As Long
Dim warnings As Long

Set swApp = Application.SldWorks
sFolder = "C:\Drawings\" ' Change this to your drawing
s folder path

sFile = Dir(sFolder & "*.SLDDRW")

Do While sFile <> ""
    Set swModel = swApp.OpenDoc6(sFolder & sFile, swDoc
DRAWING, swOpenDocOptions_Silent, "", errors, warnings)
    sPDFPath = sFolder & Left(sFile, Len(sFile) - 7) &
".pdf"
    swModel.Extension.SaveAs sPDFPath, 0, 0, Nothing, e
rrors, warnings
    swApp.CloseDoc swModel.GetPathName
    sFile = Dir
Loop

MsgBox "Batch export complete."
End Sub

```



Change the sFolder path to your actual folder location. Always test on 3 to 5 files first before running across a full project folder. Make sure no files in the folder are open in SolidWorks before you start.

Macro 5: Rebuild All Open Documents and Save

Before releasing any files, every open document should be force-rebuilt and saved. Most engineers skip this or forget. Stale models cause problems downstream: wrong dimensions in drawings, incorrect BOMs, failed exports, errors in assemblies.

This macro loops through every open document in SolidWorks, forces a full rebuild, and saves it.

```
Sub RebuildAndSaveAll()  
    Dim swApp As SldWorks.SldWorks  
    Dim swModel As SldWorks.ModelDoc2  
    Dim errors As Long  
    Dim warnings As Long  
  
    Set swApp = Application.SldWorks  
    Set swModel = swApp.GetFirstDocument  
  
    Do While Not swModel Is Nothing  
        swModel.ForceRebuild3 False  
        swModel.Save3 swSaveAsCurrentVersion, errors, warni  
ngs  
        Set swModel = swModel.GetNext  
    Loop  
  
    MsgBox "All documents rebuilt and saved."  
End Sub
```



Run this before any file release, drawing package submission, or PDM check-in. It takes under a minute and prevents the rebuild warnings that slow down the engineers receiving your files.

What These 5 Macros Actually Prove

Task automation is the entry level. Repetitive actions become one click. Hours become minutes. That is real value and you can start delivering it today.

But the same API that powers these macros also powers things at a completely different scale:

- Product configurators that generate complete models and drawing packages from customer inputs

- Automated quoting systems that replace 3 days of manual work with a 20-minute process
- Drawing QA checkers that audit every sheet before release and flag errors automatically
- Design validation tools that check manufacturing constraints before the file ever leaves CAD

The engineers who build those tools are not senior developers. They are design engineers who started exactly where you are right now: with a simple macro they copied and ran for the first time.



The SolidWorks API is not a developer tool. It is a design engineer tool. The engineer who learns it does not just save time. They become the person who builds the tools everyone else depends on.

What To Do Next

Pick one macro from this guide. Open SolidWorks. Go to Tools > Macro > New. Paste the code. Run it on a test file.

That first run is the moment everything changes. You will see SolidWorks do something you used to do by hand, and you will understand why engineers who learn this never go back.



Start with Macro 2. Open any drawing. Paste the Save as PDF code. Run it. Your first automation is done in under 5 minutes.



Ready to go from macros to building real SolidWorks automation tools? I teach the full API path at ScriptedCAD. scriptedcad.com