

Samples for the SAP IBP Ribbon Extension

You can extend the **SAP IBP** ribbon of the SAP IBP, add-in for Microsoft Excel (Excel add-in) with additional VBA functions by using an .xml file and VBA coding included in a .xlam file.

This document provides detailed information about the steps that need to be done and additional information related to the sample files provided.

MAIN STEPS

1. Create a file called **RibbonXML.xml** and save it at location **%localappdata%\SAP_IBP_AddIn**. The .xml file contains details about the elements to be added and names the VBA functions that are to be triggered. For details see SAP Help Portal at [Contents of the RibbonXML.xml](#).
2. Create a Microsoft Excel .xlam file containing the implementation of the VBA functions that are named in the .xml file and save it at location **%appdata%\Microsoft\AddIns** to make it available as Excel add-in. You can decide how to name the file. In our examples we are using **IBPExtension.xlam**.
3. Enable the add-in by choosing **Microsoft Excel > File > Options > Add-ins > Manage: Excel Add-ins > Go...** and selecting its checkbox. This way the add-in is loaded each time you open a Microsoft Excel workbook.

Please note: This needs to be done on each PC where the SAP IBP, add-in for Microsoft Excel should be extended.

4. The **SAP IBP** ribbon is automatically updated with your additional VBA functions. The functions are added to the right part of the **SAP IBP** ribbon.

SAMPLES

1.1 Sample to explain the content of the .xml and .xlam file

This sample provides a .xml file that includes all different kind of components. It is meant to provide an overview about how the content could look like and how to model the different components. There's no concrete business use case behind. The same applies to the .xlam file. Instead of message dialogs showing up you would need to implement your VBA functions that fit to your business use case.

Please note: To be able to use the <OnActionParameters> specified in the sample .xml file you need to have at least version 2302.2.0 of the Excel add-in installed.

1.2 Sample to add one basic function

This is an easy example how to extend the **SAP IBP** ribbon by one VBA function. With the sample files you are able to add a basic VBA function to the **SAP IBP** ribbon that is triggering one of the existing [SAP IBP APIs](#). The function allows you to log on to a specific connection (system and planning area) by using the **SAP IBP API** "Connect".

Please note: The connection name you are passing as parameter when calling the API "Connect" needs to be an existing name of a connection.

1.3 Sample for 2302 and newer

This is a more complex example that builds on the previous sample. With the sample files you are able to add several functions to the **SAP IBP** ribbon, using the **SAP IBP APIs** "Connect" and "OpenFavorite" and <OnActionParameters> that are available with Excel add-in version 2302.2.0 and newer. Additionally, the functions are getting enabled or disabled dependent on whether you are connected to an IBP system or not.

Please note: The connection name you are passing as parameter when calling the API “Connect” needs to be an existing name of a connection. Same applies for the parameter you are passing when calling the API “OpenFavorite”, the name of the planning view favorite needs to exist in your system and planning area.

CONCLUSION

This sample files are meant to show you the possibilities of the **SAP IBP Ribbon Extension** functionality. Even if you do not see an added value in the additional functions provided, we encourage you to think about your own business use cases and how you could implement those along with the **SAP IBP APIs** we are providing.

Follow us



www.sap.com/contactsap

© 2023 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See www.sap.com/trademark for additional trademark information and notices.