

2021 - All about



# Macro Dialog feature

By Henry Castro

Software Engineer  Collabora  
Online

[hcastro@collabora.com](mailto:hcastro@collabora.com)

# Overview

- Run VBA
- Javascript Dialog

# Running Macros

- UNO Command “.uno:RunMacro”
- Macro Selector Dialog
- Run Script “SfxObjectShell::CallXScript”

# UNO Command “.uno:RunMacro”

sfx2/source/appl/appserv.cxx

```
case SID_RUNMACRO:
{
    SfxAbstractDialogFactory* pFact = SfxAbstractDialogFactory::Create();
    SAL_INFO("sfx.appl", "SfxApplication::OfaExec_Impl: case ScriptOrg");

    Reference <XFrame> xFrame(GetRequestFrame(rReq));
    if ( !xFrame.is() )
    {
        const SfxViewFrame* pViewFrame = SfxViewFrame::Current();
        if ( pViewFrame )
            xFrame = pViewFrame->GetFrame().GetFrameInterface();
    }
}
```

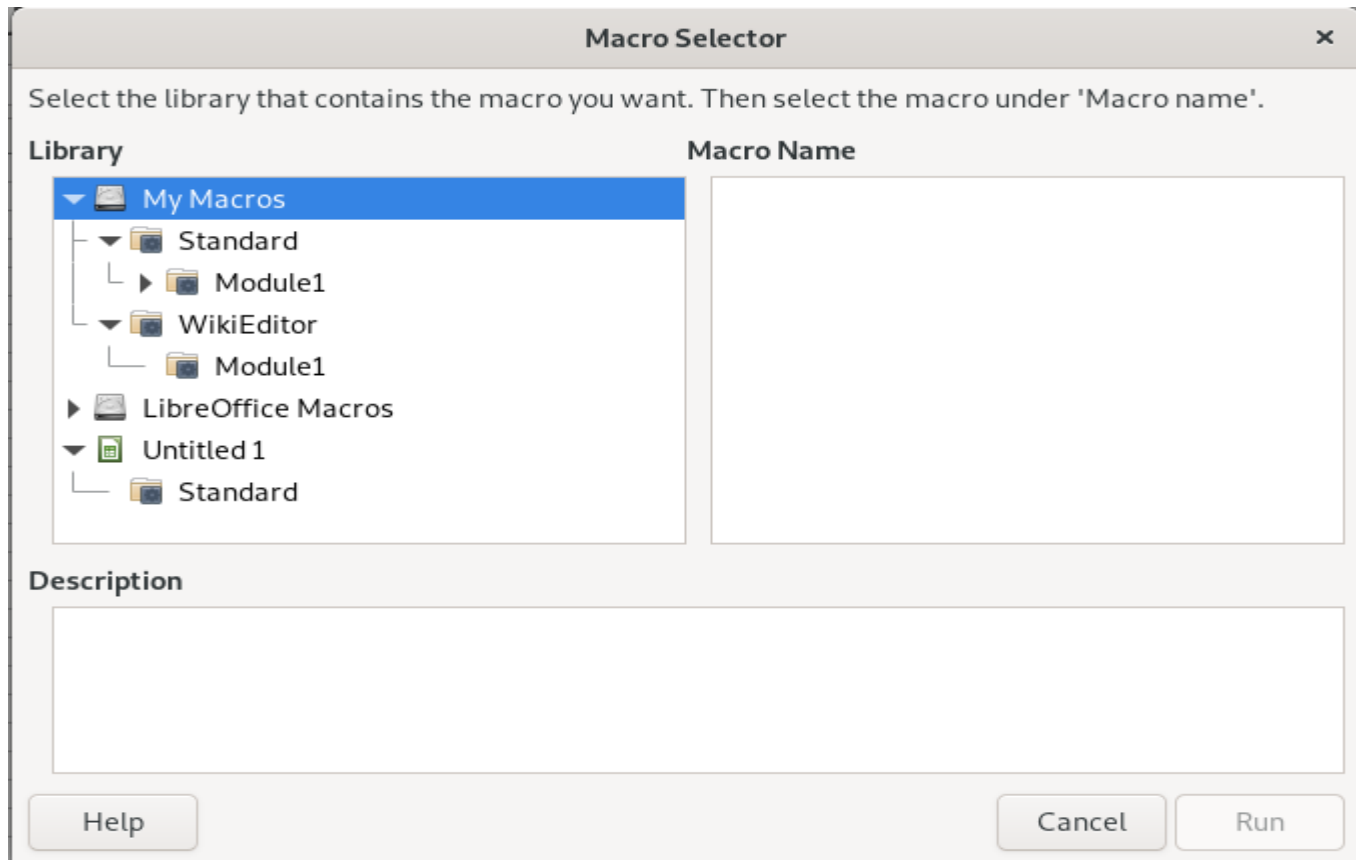
# Macro Selector Dialog

sfx2/source/appl/appserv.cxx

```
do // artificial loop for flow control
{
    VclPtr<AbstractScriptSelectorDialog> pDlg(
        pFact->CreateScriptSelectorDialog(
            lcl_getDialogParent(xFrame), xFrame));
    OSL_ENSURE( pDlg, "SfxApplication::OfaExec_Impl( SID_RUNMACRO ): no c
    if ( !pDlg )
        break;
    pDlg->SetRunLabel();

    pDlg->StartExecuteAsync([pDlg, xFrame](sal_Int32 nDialogResult) {
        if ( !nDialogResult )
        {
            pDlg->disposeOnce();
            return;
        }
    })
}
```

# Macro Selector Dialog

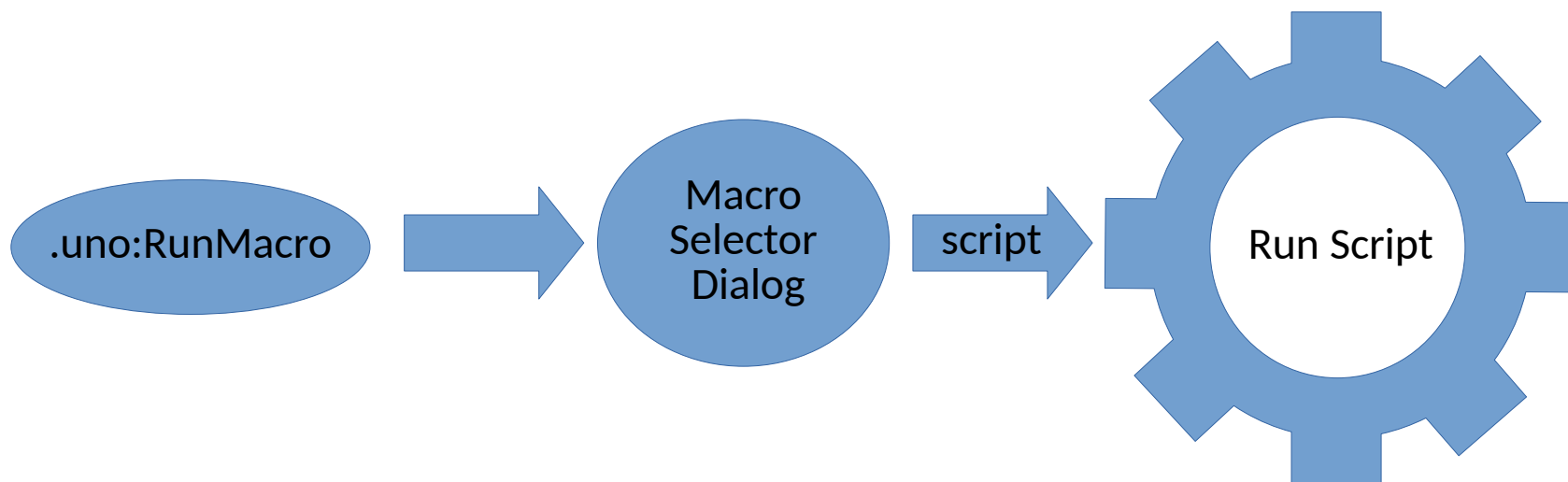


# Run Script

sfx2/source/appl/appserv.cxx

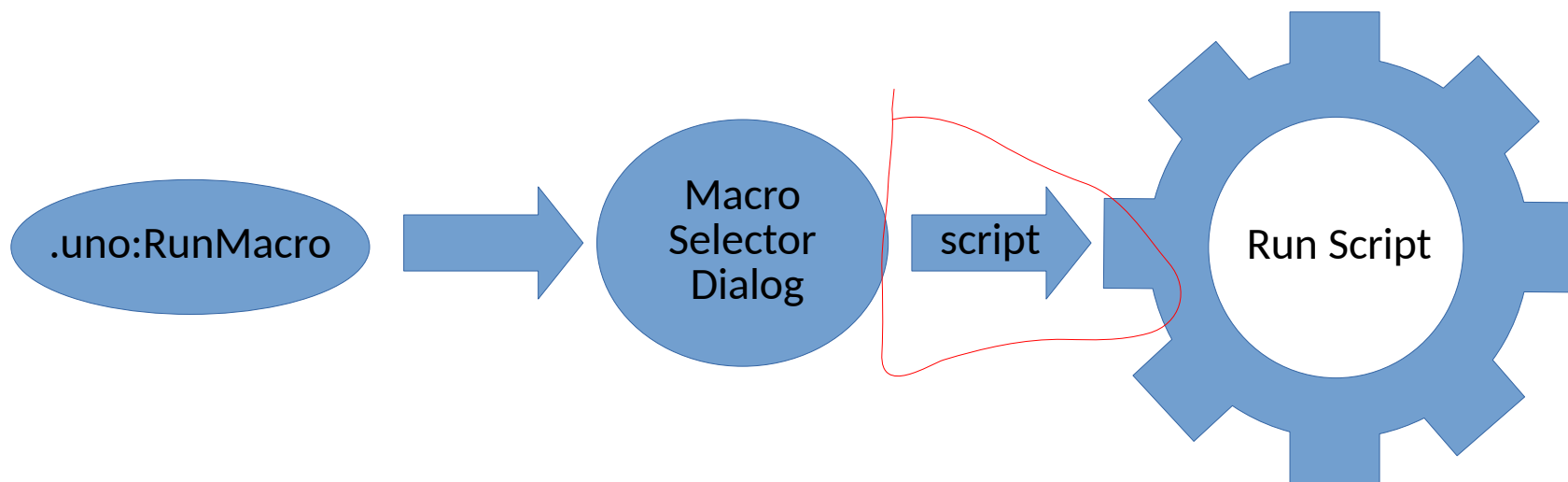
```
SfxObjectShell::CallXScript( xScriptContext,  
                             pDlg->GetScriptURL(),  
                             args,  
                             ret,  
                             outIndex,  
                             outArgs );  
  
    pDlg->disposeOnce();  
});  
}  
while ( false );  
rReq.Done();
```

# Run Script

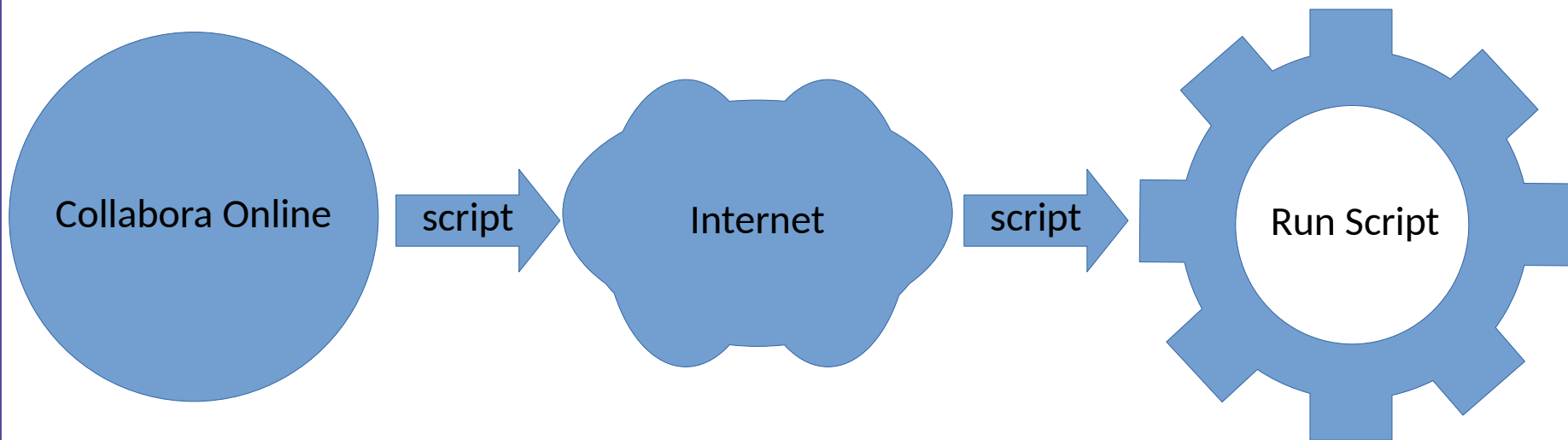




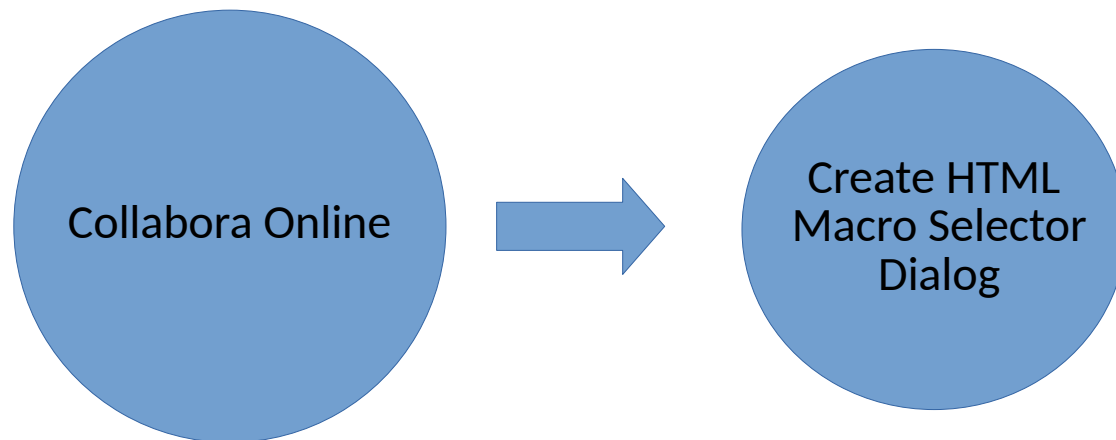
# Run Script



# Run Script



# Macro Selector Dialog



# Macro Selector Dialog

- Static Dialogs
- Tunneling Dialogs
- Dynamic Javascript Dialogs

# Macro Selector Dialog

- Static Dialogs
- Tunneling Dialogs
- Dynamic Javascript Dialogs

# Macro Selector Dialog

cui/source/customize/cfgutil.cxx

```

/*
 * Implementation of SvxScriptSelectorDialog
 *
 * This dialog is used for selecting Slot API commands
 * and Scripting Framework Scripts.
 */

SvxScriptSelectorDialog::SvxScriptSelectorDialog(
    weld::Window* pParent, const css::uno::Reference< css::frame:
    : GenericDialogController(pParent, "cui/ui/macroselectordialc
    , m_xDialogDescription(m_xBuilder->weld_label("helpmacro"))
    , m_xCategories(new CuiConfigGroupListBox(m_xBuilder->weld_tr
    , m_xCommands(new CuiConfigFunctionListBox(m_xBuilder->weld_t
    , m_xLibraryFT(m_xBuilder->weld_label("libraryft"))
    , m_xMacronameFT(m_xBuilder->weld_label("macronameft"))
    , m_xOKButton(m_xBuilder->weld_button("ok"))

```

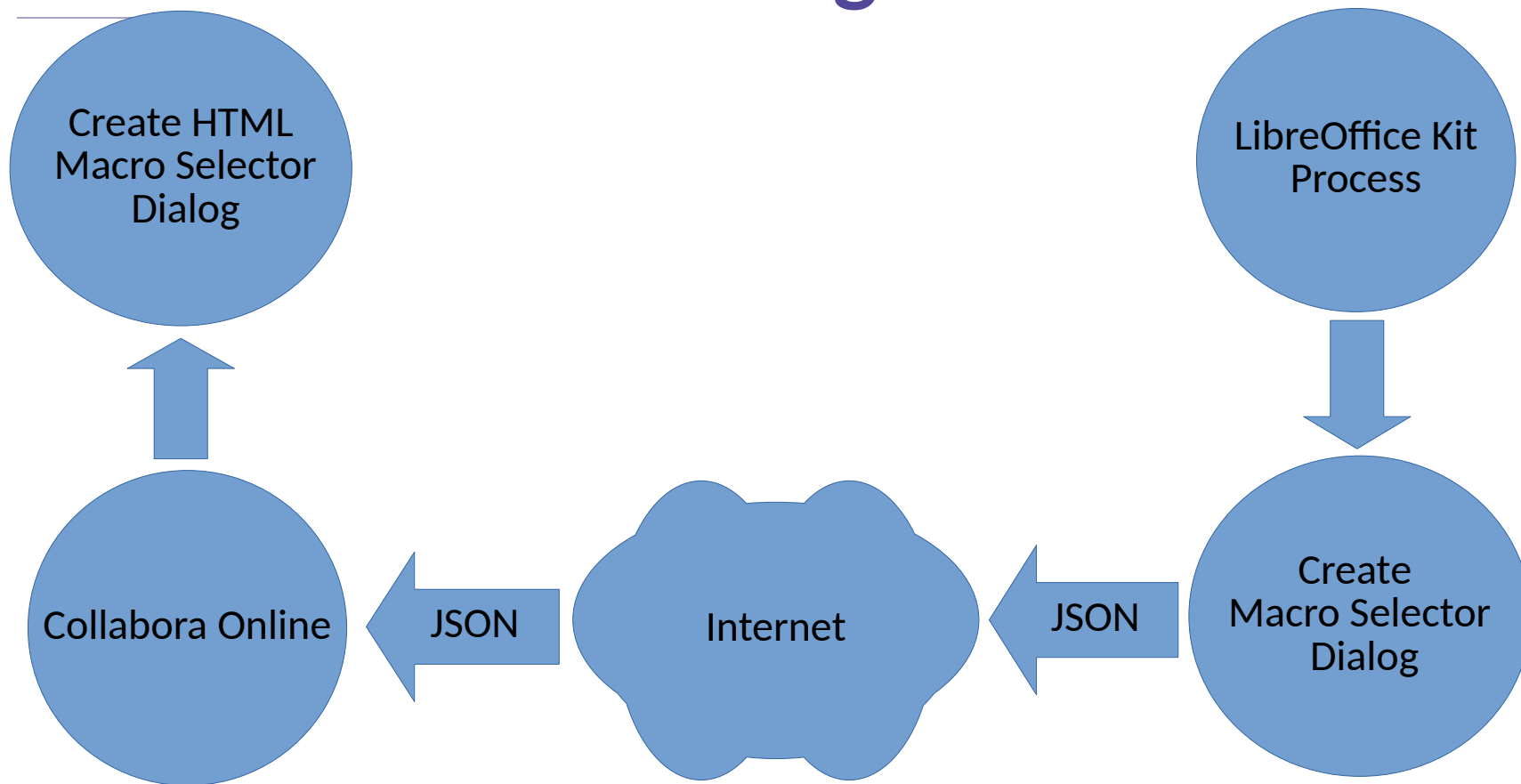
# Macro Selector Dialog

vcl/source/window/dialog.cxx

```
void Dialog::DumpAsPropertyTree(tools::JsonWriter& rJsonWriter)
{
    SystemWindow::DumpAsPropertyTree(rJsonWriter);
    rJsonWriter.put("title", GetText());
    if (vcl::Window* pActionArea = get_action_area())
    {
        if (!pActionArea->IsVisible())
            rJsonWriter.put("collapsed", "true");
    }

    OUString sDialogId = OStringToOUString(GetHelpId(), RTL_TEXTENCODING_UTF8);
    sal_Int32 nStartPos = sDialogId.lastIndexOf('/');
    nStartPos = nStartPos >= 0 ? nStartPos + 1 : 0;
    rJsonWriter.put("dialogid", sDialogId.copy(nStartPos));
}
```

# Macro Selector Dialog





# Macro Selector Dialog

loleaflet/src/control/Control.JSDialog.js

```
onJSDialog: function(e) {  
    var that = this;  
    var posX = 0;  
    var posY = 0;  
    var data = e.data;  
    var callback = e.callback;  
    var isSnackbar = data.type === 'snackbar';  
    var isModalPopup = data.type === 'modalpopup' || isSnackbar;  
  
    if (data.action === 'fadeout')  
    {  
        if (data.id && this.dialogs[data.id]) {  
            var container = this.dialogs[data.id].container;  
            L.DomUtil.addClass(container, 'fadeout');  
            container.onanimationend = function() { that.close  
            // be sure it will be removed
```

# Macro Selector Dialog

lleaflet/src/control/Control.JSDialog.js

```
var content = L.DomUtil.create('div', 'lokdialog ui-dialog-content ui-  
  
var builder = new L.control.jsDialogBuilder({windowId: data.id, mobile  
jsdialog', callback: callback});  
  
if (isModalPopup && !isSnackbar) {  
    var overlay = L.DomUtil.create('div', builder.options.cssClass  
ble ? 'cancellable' : ''), document.body);  
    overlay.id = data.id + '-overlay';  
    if (data.cancellable)  
        overlay.onclick = function () { that.closePopover(data  
    }  
    builder.build(content, [data]);
```

# Macro Selector Dialog

The screenshot shows the Macro Selector dialog in the Collabora Online interface. The dialog is titled "MACRO SELECTOR" and contains the following text: "Select the library that contains the macro you want. Then select the macro under 'Macro name'." Below this text, there are two columns: "LIBRARY" and "MACRO NAME". The "LIBRARY" column lists several libraries, including "Capitalise", "HelloWorld", "InsertText", "LibreLogo", "NamedRanges", "SetCellColor", and "pythonSamples". The "MACRO NAME" column lists macros under the selected "LibreLogo" library: "DefineNamedRange", "DeleteNamedRange", and "GetNamedRanges". The "NamedRanges" library is currently selected, and the "DefineNamedRange" macro is highlighted. At the bottom of the dialog, there are buttons for "Help", "Cancel", and "Run".

The background spreadsheet shows a table with columns C, D, E, and F, and rows 1 to 36. The right sidebar shows the "Style" panel with various formatting options.

# Macro Selector Dialog

- Re-creation for each user interaction
- Improve only partial update
- more improves?

# Macro Selector Dialog

loleaflet/src/control/Control.JSDialog.js

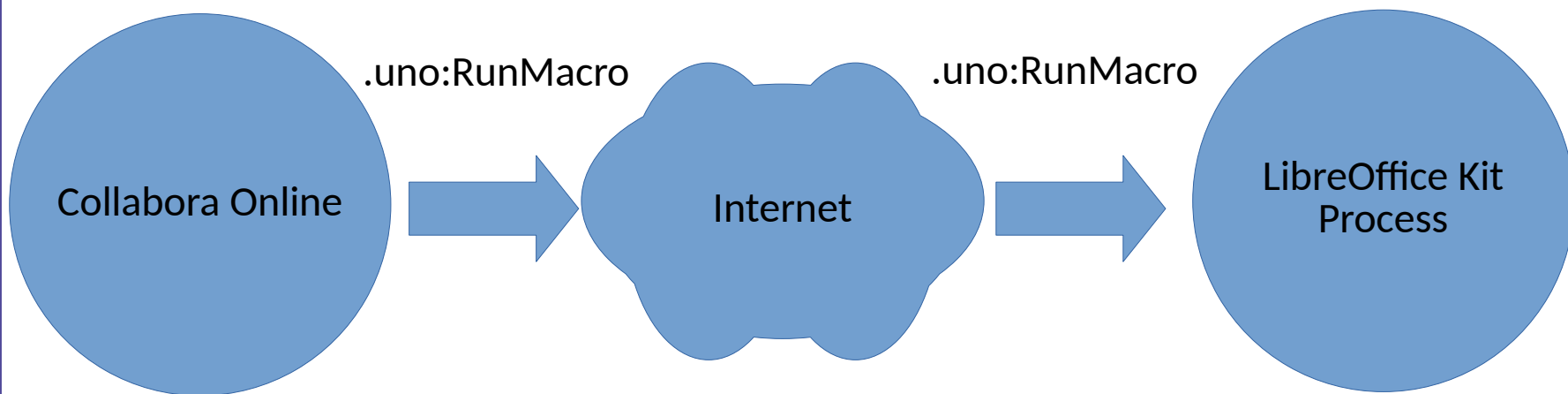
```
onJSUpdate: function (e) {  
    var data = e.data;  
  
    if (data.jsonType !== 'dialog')  
        return;  
  
    var dialog = this.dialogs[data.id] ? this.dialogs[data.id].control :  
    if (!dialog)  
        return;  
  
    var control = dialog.querySelector('[id=\'\' + data.control.id  
    if (!control) {  
        console.warn('jsdialogupdate: not found control with :  
        return;  
    }  
}
```

# Macro Selector Dialog

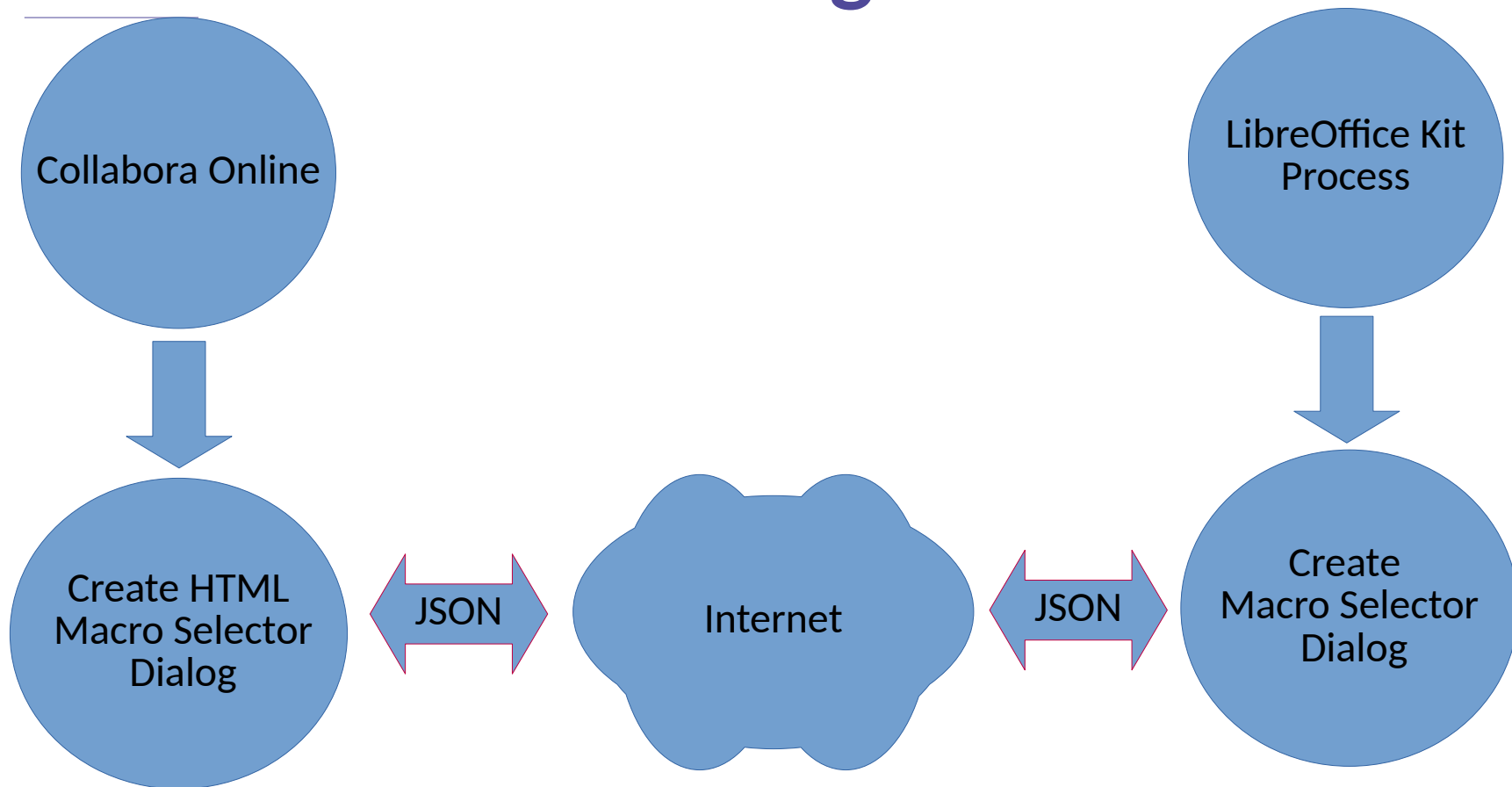
lleaflet/src/control/Control.JSDialog.js

```
control.style.visibility = 'hidden';  
var builder = new L.control.jsDialogBuilder({windowId: data.id,  
    mobileWizard: this,  
    map: this.map,  
    cssClass: 'jsdialog'});  
  
var temporaryParent = L.DomUtil.create('div');  
builder.build(temporaryParent, [data.control], false);  
parent.insertBefore(temporaryParent.firstChild, control.nextSibling);  
L.DomUtil.remove(control);
```

# Run Script

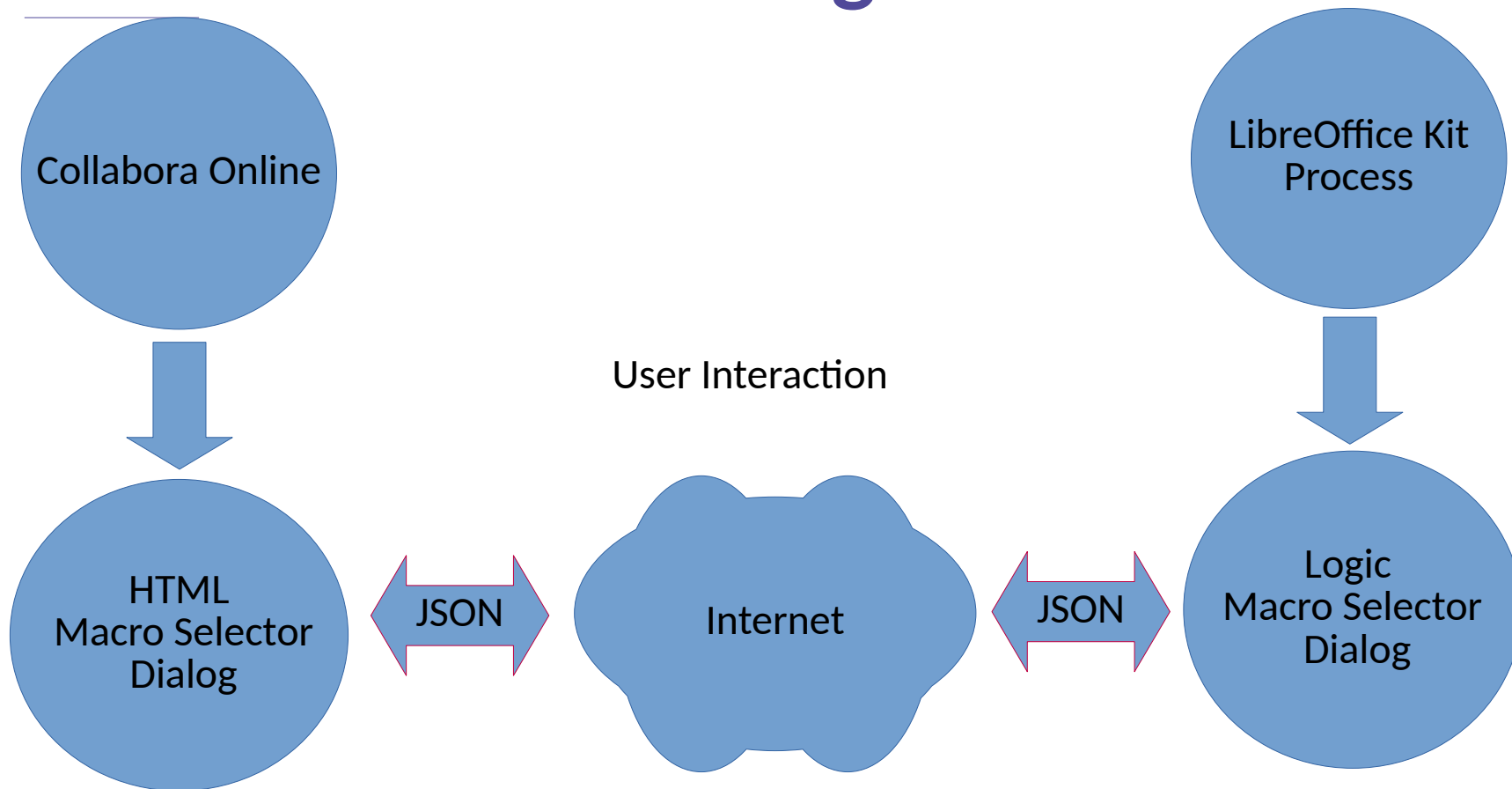


# Macro Selector Dialog

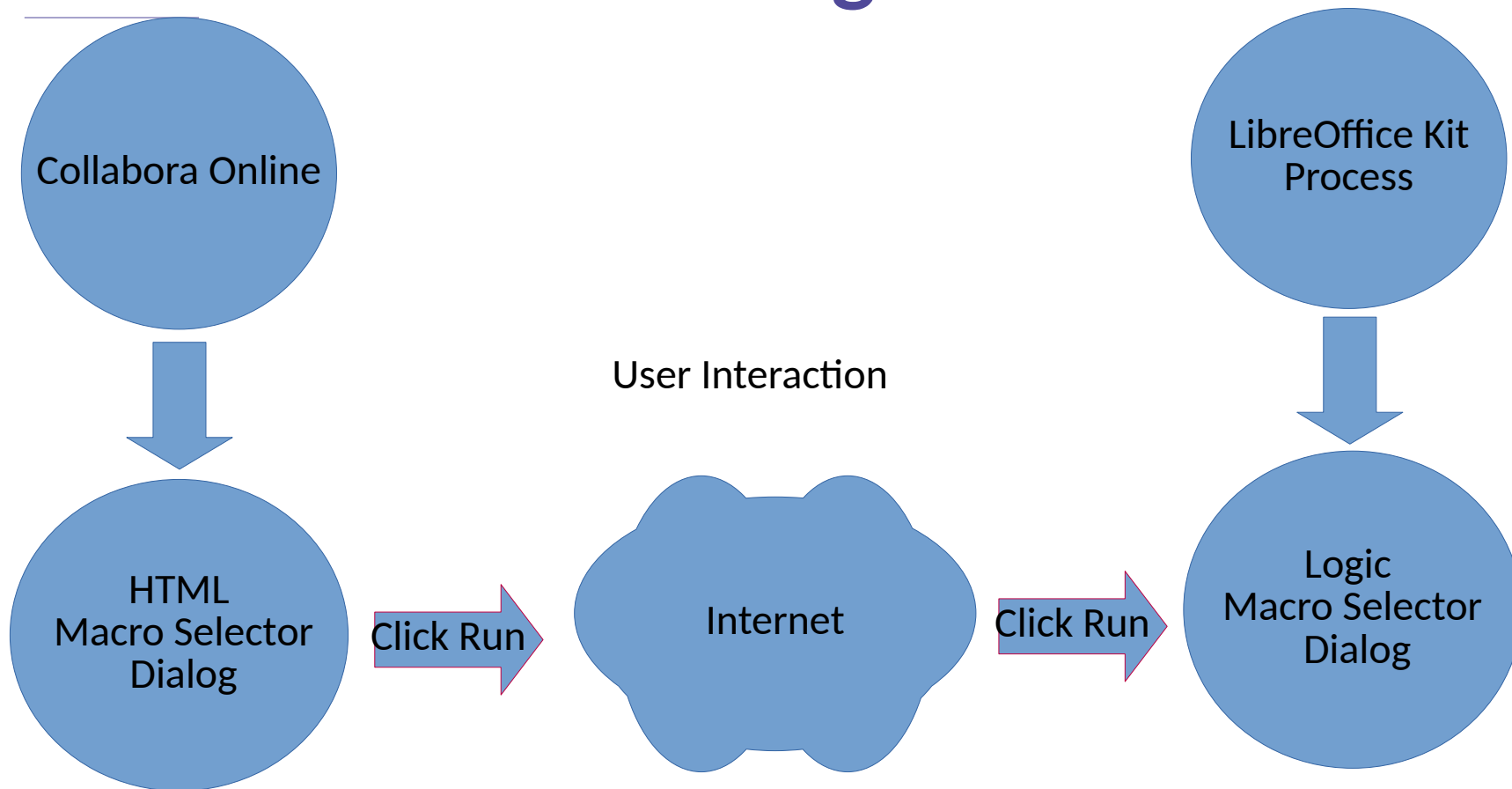




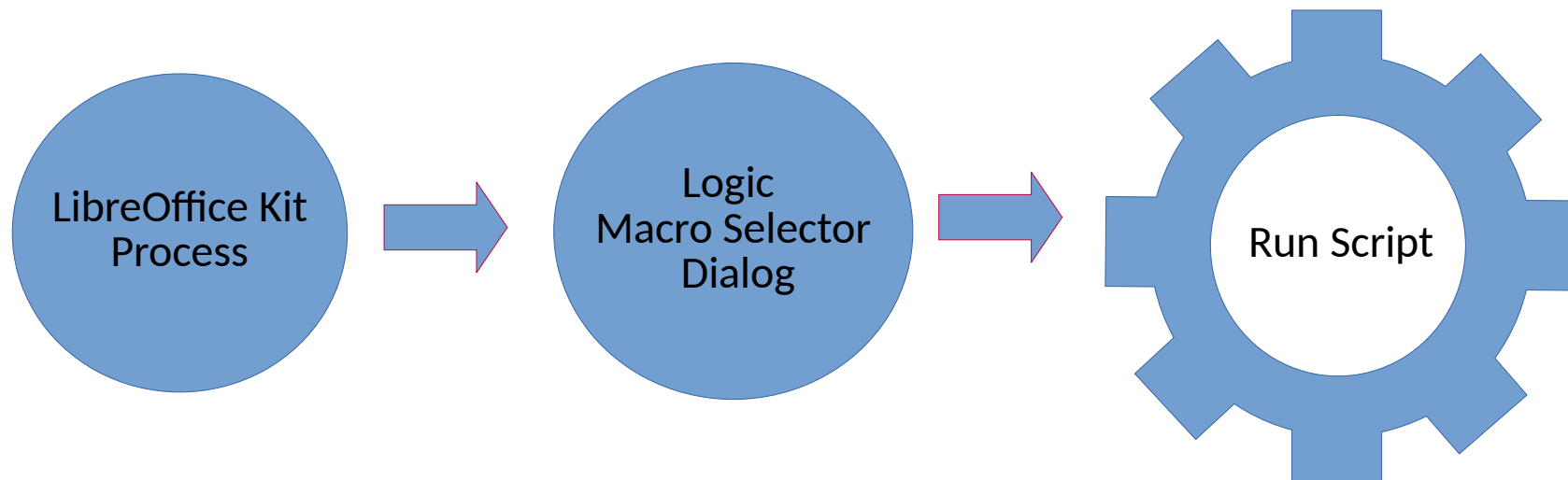
# Macro Selector Dialog



# Macro Selector Dialog



# Macro Selector Dialog



2021 - All about



Collabora Online

# Thanks !



Collabora  
Online

By Henry Castro

@CollaboraOffice  
hcastro@collabora.com  
Collaboraoffice.com