For now, the scripts actually rename each layer based on user input, however, I want it to be fully automated, therefore the input for rename.jsx would be the primary part of the filename Example filename “MIC\_3S\_51\_F1\_NURIA-1.psd”, therefore:

o Brand = MIC\_

o Product = 3S\_

o Fabric type/family = 51\_

o Style of Legs = F1\_

o Family = NURIA

So: MIC\_3S\_51\_F1\_NURIA

In this step, user used to input the same but manual, I want the script to do it by itself automatically now. Extracting the primary part and based and adding the numbers at the end as it used to do.

For the second part of the naming, I would like to combine the add\_suffix.jsx in the equation, this script used to add “-” + an input number by the user, as you see, this number is also in the filename before .psd , therefore, the script should automatically detect the “-x” number and input it based on the filename.

I want to make these scripts to become 1 and be fully automatic based on the name of the file the script has been run.

Extraction of the key information from the filename; PSD filename complete structure example:

o Brand = MIC\_

o Product = 3S\_

o Fabric type/family = 51\_

o Style of Legs = F1\_

o Family = NURIA

o Camera View = -1

o File type = .psd

Full example = MIC\_3S\_51\_F1\_NURIA-1.psd

Here are the old scripts:

“reame.jsx”:

function main() {

    // user settings

    var prefs = new Object();

    prefs.countFrom     = 1;    // number to start counting from (default: 1)

    prefs.zeroPadding   = 1;    // number of digits to use for the layer number (defaul: 3)

    prefs.nameSeparator = '';  // character to insert between the layer name and number (default: ' ')

    prefs.topToBottom   = false; // rename layers top to bottom (false) or bottom to top (true)

    // prompt for layer name

    prefs.layerPattern = prompt('Enter the name.\n');

    // rename layers

    if (prefs.layerPattern) {

        renameLayers(activeDocument, prefs);

    }

}

///////////////////////////////////////////////////////////////////////////////

// renameLayers - rename layers, top to bottom, or bottom to top

///////////////////////////////////////////////////////////////////////////////

function renameLayers(*ref*, *prefs*) {

    // declare local variables

    var len = ref.layers.length;

    // rename layers top to bottom

    if (prefs.topToBottom) {

        for (var i = 0; i < len; i++) {

            rename();

        }

    }

    // rename layers bottom to top

    else {

        for (var i = len - 1; i >= 0; i--) {

            rename();

        }

    }

    // rename - rename layer

    function rename() {

        var layer = ref.layers[i];

        var vis = layer.visible;

        // check for groups

        if (layer.typename == 'LayerSet') {

    //      renameLayers(layer, prefs);

    return 0;

        }

        // rename layer

        else {

            layer.name = prefs.layerPattern + prefs.nameSeparator +

            (prefs.countFrom.toString());

        //      (prefs.countFrom + Math.pow(10, prefs.zeroPadding)).toString().substr(1);

            if (!vis) {

                layer.visible = false;

            }

            prefs.countFrom++;

        }

    }

}

///////////////////////////////////////////////////////////////////////////////

// isCorrectVersion - check for Adobe Photoshop CS (v26) or higher

///////////////////////////////////////////////////////////////////////////////

function isCorrectVersion() {

    if (parseInt(version, 10) >= 26) {

        return true;

    }

    else {

        alert('This script requires Adobe Photoshop CS or higher.', 'Wrong Version', false);

        return false;

    }

}

///////////////////////////////////////////////////////////////////////////////

// isOpenDocs - ensure at least one document is open

///////////////////////////////////////////////////////////////////////////////

function isOpenDocs() {

    if (documents.length) {

        return true;

    }

    else {

        alert('There are no documents open.', 'No Documents Open', false);

        return false;

    }

}

///////////////////////////////////////////////////////////////////////////////

// showError - display error message if something goes wrong

///////////////////////////////////////////////////////////////////////////////

function showError(*err*) {

    if (confirm('An unknown error has occurred.\n' +

        'Would you like to see more information?', true, 'Unknown Error')) {

            alert(err + ': on line ' + err.line, 'Script Error', true);

    }

}

///////////////////////////////////////////////////////////////////////////////

// test initial conditions prior to running main function

///////////////////////////////////////////////////////////////////////////////

if (isCorrectVersion() && isOpenDocs()) {

    try {

        main();

    }

    catch(e) {

        if (e.number != 8007) { // don't report error on user cancel

            showError(e);

        }

    }

}

“add\_suffix.jsx”:

function main() {

    // user settings

    var prefs = new Object();

    prefs.nameSeparator = '-';  // character to insert between the layer name and number (default: ' ')

    prefs.topToBottom   = false; // rename layers top to bottom (false) or bottom to top (true)

    // prompt for layer name

    prefs.layerPattern = prompt('Enter the view number:\n');

    // rename layers

    if (prefs.layerPattern) {

        addSuffix(activeDocument, prefs);

    }

}

///////////////////////////////////////////////////////////////////////////////

// addSuffix - rename layers, top to bottom, or bottom to top

///////////////////////////////////////////////////////////////////////////////

function addSuffix(*ref*, *prefs*) {

    // declare local variables

    var len = ref.layers.length;

    // rename layers top to bottom

    if (prefs.topToBottom) {

        for (var i = 0; i < len; i++) {

            rename();

        }

    }

    // rename layers bottom to top

    else {

        for (var i = len - 1; i >= 0; i--) {

            rename();

        }

    }

    // rename - rename layer

    function rename() {

        var layer = ref.layers[i];

        var vis = layer.visible;

        // check for groups

        if (layer.typename == 'LayerSet') {

    //      addSuffix(layer, prefs);

    return 0;

        }

        // rename layer

        else {

            layer.name = layer.name + prefs.nameSeparator + prefs.layerPattern;

            if (!vis) {

                layer.visible = false;

            }

            prefs.countFrom++;

        }

    }

}

///////////////////////////////////////////////////////////////////////////////

// isCorrectVersion - check for Adobe Photoshop CS (v8) or higher

///////////////////////////////////////////////////////////////////////////////

function isCorrectVersion() {

    if (parseInt(version, 10) >= 26) {

        return true;

    }

    else {

        alert('This script requires Adobe Photoshop CS or higher.', 'Wrong Version', false);

        return false;

    }

}

///////////////////////////////////////////////////////////////////////////////

// isOpenDocs - ensure at least one document is open

///////////////////////////////////////////////////////////////////////////////

function isOpenDocs() {

    if (documents.length) {

        return true;

    }

    else {

        alert('There are no documents open.', 'No Documents Open', false);

        return false;

    }

}

///////////////////////////////////////////////////////////////////////////////

// showError - display error message if something goes wrong

///////////////////////////////////////////////////////////////////////////////

function showError(*err*) {

    if (confirm('An unknown error has occurred.\n' +

        'Would you like to see more information?', true, 'Unknown Error')) {

            alert(err + ': on line ' + err.line, 'Script Error', true);

    }

}

///////////////////////////////////////////////////////////////////////////////

// test initial conditions prior to running main function

///////////////////////////////////////////////////////////////////////////////

if (isCorrectVersion() && isOpenDocs()) {

    try {

        main();

    }

    catch(e) {

        if (e.number != 8007) { // don't report error on user cancel

            showError(e);

        }

    }

}