

# Barcodetrennung mit ausschließen der Trenner-Seite

In folgendem Ordner der Squeeze Installation:

SQUEEZE\htdocs\repository\<Mandantenname>\UserExits\Global\

muss die PHP-Datei BarcodeSeparation.php liegen.

Unter Barcode Types sind die möglichen Barcode-Typen aufgeführt welche verwendet werden können. Diese können dann über die Stapelklasseneigenschaften konfiguriert werden.

Wichtig: Der Wert für den Barcode Typ ist case-sensitiv (Groß-Kleinschreibung beachten!).

```
<?php

use Squeeze\SqueezeConfig;
use Squeeze\xBatchClass;
use Squeeze\xDoc;
use Squeeze\xQueueEntry;
use Squeeze\xReturnObject;
use Squeeze\xTools;

/**
 * @param xDoc $xDoc
 * @param string $nextStep
 * @param string $newStatus
 * @return xReturnObject
 * @throws Exception
 */
function BarcodeSeparation(xDoc $xDoc, string $nextStep, string $newStatus): xReturnObject
{
    $logger = Logger::getLogger("main");
```

```

// =====
// Get Barcode Engine from Batch Class Settings
// =====
$isBarcodeConfigured = true;
$xQueueEntry = new xQueueEntry();
$xQueueEntry->getByxDocId($xDoc->id);
$batchClass = new xBatchClass();
$batchClass->getById($xQueueEntry->batchclassid);
$splitBarcodeType = strtolower(trim($batchClass-
>getSettingValue('SplitBarcodeType')));

$splitBarcodePattern = $batchClass->getSettingValue('SplitBarcodePattern');
$splitFixPages = intval(trim($batchClass->getSettingValue('SplitFixPages')));

if ($splitBarcodeType === null or $splitBarcodeType == '') {
    $isBarcodeConfigured = false;
    //$logger->warn('DocId = '.$xDoc->id.' Barcode Split not possible because
SplitBarcodeType is missing!');
} else {
    if ($splitBarcodePattern === null or $splitBarcodePattern == '') {
        $isBarcodeConfigured = false;
        //$logger->warn('DocId = '.$xDoc->id.' Barcode Split not possible because
SplitBarcodePattern is missing!');
    }
}

if ($isBarcodeConfigured) {
    // get splits by barcode and return the result
    return splitByBarcode($xDoc, $logger, $splitBarcodeType, $splitBarcodePattern);
}

if ($splitFixPages > 0) {
    // get splits by fixed pages and return the result
    return splitFixPages($xDoc, $logger, $splitFixPages);
}

// return an empty split result
return new xReturnObject(false, 400, 'No Split configured', []);
}

/**

```

```

* @param xDoc $xDoc
* @param Logger $logger
* @param int $fixedPageSize
* @return xReturnObject
* @throws Exception
*/
function splitFixPages(xDoc $xDoc, Logger $logger, int $fixedPageSize): xReturnObject
{
    $config = new SqueezeConfig();
    $repo = $config->get("repository.work");
    $splits = [];
    $keepFieldValues = [];
    $absolutePath = xTools::buildAbsolutePath($repo, $xDoc->repoPath, false);
    $files = array_diff(scandir($absolutePath . "Viewer"), ['.', '..']);
    $currentPage = 0;
    foreach ($files as $page => $file) {
        $currentPage++;
        $modulo = $currentPage % $fixedPageSize;
        if ($modulo === 0) {
            $logger->debug(' DocId = ' . $xDoc->id . ' New split at page ' . $currentPage . '
because of fixed page size');
            $splits[$currentPage][] = ['page' => $currentPage, 'type' => 'FixedPageSplit',
'value' => 'FixedPageSplit-' . $fixedPageSize, 'fields' => $keepFieldValues, 'suppressOcr' =>
false];
        }
    }
    return new xReturnObject(true, 200, 'Fixed Page Split', $splits);
}

/**
* @param xDoc $xDoc
* @param Logger $logger
* @param string $splitBarcodeType
* @param string $splitBarcodePattern
* @return xReturnObject
*/
function splitByBarcode(xDoc $xDoc, Logger $logger, string $splitBarcodeType, string
$splitBarcodePattern): xReturnObject
{

```

```

// =====
// Barcode Types
// =====
// AZTEC
// CODABAR
// CODE_39
// CODE_93
// CODE_128
// COMPOSITE
// DATABAR
// DATA_MATRIX
// DATABAR_EXP
// EAN_2
// EAN_5
// EAN_8
// EAN_13
// ITF
// ISBN_10
// ISBN_13
// MAXICODE
// PDF_417
// QR_CODE
// RSS_14
// RSS_EXPANDED
// UPC_A
// UPC_E
// UPC_EAN_EXTENSION
// UNKNOWN

$currentSplit = 0;
$suppressOCR = 0;
$splits = [];
$keepFieldValues = [];

$splitBarcodeType = str_replace(' ', '', $splitBarcodeType);
$splitBarcodeType = trim($splitBarcodeType, ',;');
$arrSplitBarcodeTypes = explode(';', $splitBarcodeType);

$logger->debug(' DocId = ' . $xDoc->id . ' Execute Barcode-Split');

```

```

foreach ($xDoc->barcodes as $page => $barcodes) {

    $logger->debug(' DocId = ' . $xDoc->id . ' Checking Barcodes for Page ' .
$page);
    $barcodeType = '';
    $barcodeValue = '';
    $isSplitPage = false;

    if (count($barcodes) == 0) {
        if ($page == 1) {
            $logger->debug(' DocId = ' . $xDoc->id . ' No Barcode on Page ' . $page . ' . Split
because it is the first page');
            $currentSplit = $page;
        }
    } else {
        foreach ($barcodes as $barcode) {
            foreach ($arrSplitBarcodeTypes as $splitBarcodeType) {
                if (strtolower($barcode['type']) === strtolower($splitBarcodeType))
{
                    $matches = null;
                    xTools::mb_preg_match_all("/$splitBarcodePattern/i",
$barcode['value'], $matches, PREG_OFFSET_CAPTURE);
                    $hit = $matches[0];
                    foreach ($hit as $match) {
                        if ($match[0] == '') {
                            continue;
                        }
                        $logger->debug(' DocId = ' . $xDoc->id . ' Start Split here!
Barcode "' . $matchmatch[0] . '" of type "' . $barcode['type'] . '" found on page ' .
$page);

                        //$suppressOCR = 1;
                        $currentSplit = $page - 1;
                        $barcodeType = $barcode['type'];
                        $barcodeValue = $barcode['value'];
                        $isSplitPage = true;

                        // Next Page
                        break 3;
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}

// add page
if ($isSplitPage) {
    $logger->debug("IS SPLIT PAGE - DO NOT ADD PAGE $page");
} else {
    $splits[$currentSplit][] = ['page' => $page, 'type' =>$barcodeType, 'value' => $barcodeValue]
}

return new xReturnObject(true, 200, 'Barcode Split', $splits);
}

```

Ob die Trennung funktioniert, sieht man wenn in der Bildaufbereitung aus einem Dokument mehrere werden und im Squeeze.log.

Ob die Erkennung des Attachment Barcodes funktioniert hat, sieht man daran das ab diesem Barcode kein Text mehr im Viewer markiert werden kann.

---

Revision #6

Created 3 June 2022 11:00:56 by Jan Vogl

Updated 1 August 2023 11:55:09 by Ulrich Eckhardt