

Automaatio dynaamiselle julkaisuvalikoimalle

Case – Thero Design Oy

LAB-ammattikorkeakoulu

Tieto- ja viestintäteknikan insinööri (AMK)

2024

Kuura Mäenpää

Tiivistelmä

Tekijä(t)	Julkaisun laji	Valmistumisaika
Kuura Mäenpää	Opinnäytetyö, AMK	2024
	Sivumäärä	
	31	
Työn nimi		
Automaatio dynaamiselle julkaisuvalikoimalle		
Case – Thero Design Oy		
Tutkinto ja koulutusala		
Tieto- ja Viestintäteknikan insinööri (AMK)		
Toimeksiantajaorganisaatio (jos opinnäytetyöllä on toimeksiantaja)		
Thero Design Oy		
Tiivistelmä		
<p>Opinnäytteessä toteutetaan annetulla toimeenpanolla, kuvitteellisilla materiaaleilla, oikean tapauksen ja InDesign plug-in EasyCatalogia käyttäen julkaisuautomatisointi. Automaatiolla luodaan usea eri julkaisuversio samaa datalähdettä käyttäen, muuttaen dynaamisesti sen sisällön riippuen versiosta.</p> <p>EasyCatalog vie julkaisujen taiton automatisoinnin pitkälle, korvaten vanhat leikkaa ja liimaa -tekniikat tai vastaavat yksinkertaisemmat ohjelmat. Kuten InDesignia, EasyCatalogia voi ohjata JavaScriptin avulla, joka mahdollistaa hankalimmat tehtävät tai normaalisti haastavammat ratkaisut, kuten dynaamisen taiton, jossa taitto halutaan toistaa niin monta kertaa kuin erilaista dataa löytyy ja kuinka monta eri versiota datasta halutaan.</p> <p>Samaa koodia ja samaa oppia voi käyttää muissakin tapauksissa. Vaikka opinnäytteessä toteutetaan eri kieliversioiden taiton mahdollistamista, voidaan samaa koodia käyttää, jos halutaan eri datasisältö ilman käyttäjän manuaalista syöttöä.</p>		
Asiasanat		
automaatio, indesign, easycatalog		

Abstract

Author(s)	Type of Publication	Published
Kuura Mäenpää	Thesis, UAS	2024
	Number of Pages	
	31	
Title of Publication		
Automation for variety of dynamic publications		
Case – Thero Design Oy		
Degree, Field of Study		
Information and Communication technology engineer (UAS)		
Organisation of the client (if the thesis work is commissioned by another party)		
Thero Design Oy		
Abstract		
<p>In this work a solution is executed for a given assignment with hypothetical materials, for a real case and using InDesign plug-in EasyCatalog, a publishing automation, which will automatically create multiple publication versions from the same set of data, changing its contents dynamically depending on the version of publication.</p> <p>EasyCatalog takes automation of creation of publications far, replacing old copy and paste -techniques or equivalent simpler programs. Just like InDesign, EasyCatalog can be controlled by using JavaScript, which enables more complicated tasks and normally challenging solutions like dynamic publication, where pagination is repeated as long as there are more data entries available and more versions of the publication is needed.</p> <p>Same script and same teachings can be used in other cases. Even though this execution is for creating versions for different languages, the same script can be used, if different data content is wanted without the manual input from the user.</p>		
Keywords		
automation, indesign, easycatalog		

Sisällys

1	Johdanto.....	1
2	Automatisointi julkaisuissa	2
2.1	Automatisoinnin haasteet.....	2
2.2	Käytettävät ohjelmistot ja tekniikat.....	3
2.2.1	Adobe InDesign	4
2.2.2	EasyCatalog.....	6
2.2.3	Ohjelmistokohtaiset ohjelmointikielet	8
3	Toteutus	11
3.1	Lähtötilanne	11
3.1.1	InDesign-pohja	12
3.1.2	Formatting rules	13
3.1.3	EasyCatalog paneelit	14
3.1.4	Datalähde.....	16
3.2	Ohjelmointi	19
3.2.1	Skriptin muuttujat	20
3.2.2	Taiton ohjelmointi.....	22
3.2.3	Version vaihto skriptillä	24
3.3	Ongelmakohdat.....	25
3.4	Jatkokehitys	27
4	Yhteenveto	29
	Lähteet.....	30

1 Johdanto

Kun puhutaan erilaisten julkaisujen taitosta, perinteisesti tähän kuuluu monta osapuolta riippumatta julkaisun muodosta. Julkaisu voi olla esimerkiksi tarjouslehtinen, tuoteluettelo tai mikä tahansa muu dataan liittyvä digitaalinen tai fyysinen julkaisu, jolloin taitossa on mukana yleensä itse taittaja, markkinointipäällikkö, tuotepäällikkö ja sisällöntuottaja. Taitossa ilman automatisointia taittaja ottaa vanhalla leikkaa liimaa tekniikalla datasta tarvittavat tiedot julkaisuun, mutta erilaiset automaatiot täyttävät datan julkaisuun ilman tätä tarvetta.

EasyCatalog on vain yksi työkaluista, joilla automatisointi mahdollistetaan. Sitä käyttäen, saadaan muodostettua suhde datan ja Adobe InDesignin välille niin että dataa voi päivittää InDesignista käsin, mutta ennen kaikkea InDesigniin tulostuva data on suoraan sisällöntuottajan tai tuotepäällikön luomaa ja markkinointipäällikön asettamin rajauksin. Automatisoinnissa jätetään välistä riski siitä, että data on julkaisussa väärää.

Thero Design Oy on yritys, joka on tarjonnut erilaisia julkaisupalveluita, joista yksi on ollut AutoTaitto. Erilaisille asiakkaille, erilaisia julkaisuja, EasyCatalogia käyttäen on luotu julkaisuautomaatio ratkaisu jokaiselle asiakkaalle tapauskohtaisesti. Tässä tapauksessa EasyCatalogin perusominaisuuksien tarjoama automatisointi ei kuitenkaan ollut riittävä. Vaikka kokonaisia julkaisuja voidaan luoda yhdellä napin painalluksella, EasyCatalog ei taivu siihen, että se loisi samasta julkaisuista eri versioita jonkin yhden muuttujan perusteella, vaan napin painallus täytyy toistaa niin monta kertaa kuin on eri versioita. Opinnäytteen tekijä on ollut julkaisuautomaation kehittäjänä neljä vuotta.

Tässä opinnäytteessä tutkitaan ratkaisua pidemmälle menevästä automatisoinnista, joka vapauttaa taittajan mahdollisiin muihin tehtäviin tarvitsematta vahtia EasyCatalogin tekemää automaattista taittoa, kunnes voi käynnistää seuraavan version taiton. Taittaja voi käynnistää vain yhden taiton, jonka jälkeen loput hoituu automaattisesti ja taittaja voi joko viimeistellä julkaisut tai lähettää ne suoraan joko painoon ja/tai digialustalle.

2 Automatisointi julkaisuissa

2.1 Automatisoinnin haasteet

Automatisoinnista on paljon hyötyä toistuvissa samankaltaisissa prosesseissa. Monet julkaisut noudattavat tietynlaista kaavaa siitä, mitä jokaisen julkaisun tulee näyttää ja vaatii usein paljon resursseja yhden julkaisun tekemiseen. Mitä enemmän julkaisuja tehdään samalla kaaviolla ja mitä pidempiä julkaisut ovat, sitä enemmän erilaisten prosessien automatisointi hyödyttää julkaisijaa. Oli se sitten yksinkertainen yhden asetuksen muokkaaminen kaikkiin kuviin kerralla tai tuotetietojen rikastuttaminen sivulle, jokainen automaatio on tarkoitettu helpottamaan ja pienentämään käytettyjä resursseja.

Oli kyseessä minkä tahansa asian automatisointi, se perustuu siihen, että syöttö ennen automaatiota pysyy aina samassa formaatissa, ellei hyödynnetä tekoälyä. Formaatin muuttuessa asiat, joille on luotu tietynlainen automaatio, eivät enää toimikkaan. On myös tärkeää, että syötteestä löytyy kaikki automatisointiin tarvittava informaatio, siksi ennen julkaisun automatisoinnin aloittamista on tärkeää sopia syötöstä sekä taitajan, että datan käsitelijöiden kanssa.

Eryityisesti julkaisuiden automatisoinnissa on haastavaa, että harvoin kaikki julkaisussa näkyvä informaatio vastaa täydellisesti satalähteestä löytyvää muotoilua. Silloin siirrytään kompromissien tekemiseen, joko muokaten itse syötettä tarvittavaan muotoon tai tehdään kompromisseja julkaisun puolella siitä mitä informaatiota näytetään ja missä muodossa. Esimerkiksi pisteiden korvaaminen pilkuilla suomenkielisen desimaalinmerkinnän ilmaise-miseksi on yleinen syötön muokkaus automaatiossa.

Esimerkiksi tuotteisiin perustuvassa julkaisun automatisoinnissa on myös haastavaa, että harvoin tuotteet ovat identtisiä toistensa kanssa ja kaikkien tuotteiden kohdalla halutaan käyttää aina samaa informaatiota. Joku tuote saattaa tarvita tietoinaan pituuden ja leveyden, mutta joku toinen saattaa tarvita painon. Silloin on tärkeää keskustella tuotepäällikön ja markkinointipäällikön kanssa siitä mitä halutaan näyttää ja millä ehdoilla. Automaatiolla voidaan tällaiset kompromissit toteuttaa, jos loogiset ehdot tietojen esiintymiselle löytyvät. Esimerkiksi helposti toteutettavissa oleva logiikka on että, jos tuotteella ei ole dataa asiasta, sitä ei tule julkaisuun. Mutta muitakin ehtoja voidaan haluta asettaa, kuten tietyn tuotekategorian perusteella esitettävät tiedot voidaan määrittää erikseen tai esimerkiksi päätetään, haetaanko leveyttä ollenkaan, jos pituutta ei löydy.

Automatisoinnissa ehtojen määrittely onkin tärkein asia, joka on tarkkaan käytävä läpi jokaisen asiakkaan kohdalla erikseen. Riippuen tapauksesta kompromisseja joudutaan usein tekemään joko syötön puolella tai sitten taitovaiheessa ja usein julkaisut edelleen vaativat

pientä säätöä taittajalta, jos automaatio ei taipunut haluttuun lopputulokseen annetuilla datalähteillä. Automaatio kuitenkin helpottaa taittajan työtä tapauksesta riippumatta, jättäen usein pelkän viimeistelyn taittajan tehtäväksi. Usein tähän sisältyy esimerkiksi julkaisun kansien tekeminen, mainossivujen tai elementtien sijoittaminen, tekstin rivittäminen tai kuvien rajaaminen ja koon säätäminen.

Eriyisesti silloin kun samasta julkaisusta halutaan monta eri versiota, automatisointi nopeuttaa työtä huomattavasti ja mahdollistaa samaa resurssimäärää käyttäen useamman julkaisun tekemisen kuin oli ennen mahdollista. Esimerkiksi samasta julkaisusta halutaan usein eri kieliversiot erilaisten yleisöjen saavuttamisen vuoksi tai julkaisun kansainvälisyyden vuoksi. Taittaja voi käyttää automatisointia jokaiselle tarvittavalle kielelle ja tehdä vain tarpeelliset loppuviimeistelyt, jolloin aika, mikä aikaisemmin käytettiin datan käsittelemiseen, ”leikkaamiseen ja liimaamiseen”, voi automatisoinnin käyttöönoton jälkeen siirtää toiseen käyttöön.

Julkaisujen automaatio ei kuitenkaan normaalisti käynnisty itsestään, vaan se vaatii käyttäjän manuaalisen käynnistyksen. Käyttäjä, joka on yleensä myös taittaja, valitsee halutut tuotteet, julkaisutyyppin sekä version taittavaksi. Joka kerta, kun jokin näistä kolmesta muuttuu, on käyttäjän käynnistettävä automatisointi uudestaan halutuilla valinnoilla ja samasta julkaisusta voidaan saada useampi variaatio.

Datamäärän suuruus vaikuttaa hyvin paljon siihen, miten kauan automatisoitu taitto kestää. Pienellä datalla taiton kestokin on pieni, mutta riippuen käytettävissä olevasta laitteistosta ja ohjelmistosta, suuren datan taitossa saattaa kestää useita tunteja. Myös aiemmin mainitut datan käsittelyyn tehtävät logiikat pidentävät latausaikaa, kun data päivitetään.

Silloin kun halutaan esimerkiksi päivittäin taittaa julkaisuja isosta datamassasta ja usealla eri versiolla, se saattaa sitoa käyttäjän vahtimaan taiton valmistumista ja käytettävän laitteiston siihen asti, että taitto on valmis. Vaikka perinteisessäkin taitossa laitteisto ja taittaja ovat varattuja ja automatiikan avulla säästetään aikaa, ei ole ideaalia, että käyttäjä joutuu vahtimaan taiton valmistumista vain käynnistääkseen seuraavan version taittamisen, jos automatisointi on jo viety siihen pisteeseen, ettei taittajan tarvitse sitä käsitellä enää jälkeinpäin ollenkaan.

2.2 Käytettävät ohjelmistot ja tekniikat

Erilaisten julkaisujen taittamiseen tarvitaan usein erilaisia ominaisuuksia, jotka ovat hyvin julkaisuihin keskittyneitä ratkaisuja. Esimerkiksi monisivuisessa julkaisussa on tärkeää, että on mahdollisuus luoda käytettävässä ohjelmistossa useampia sivuja samaan tiedostoon.

Näillä saa luotua PDF-tiedostoja, jotka voi sitten lähettää halutulle alustalle julkaistavaksi tai painettavaksi.

Kun useampi kuin yksi sisällöntuottaja työstää samaa julkaisua varten eri sisältöjä, on myös tärkeää, että käytettävät ohjelmistot toimivat keskenään ja mahdollisesti useammassa eri käyttöjärjestelmässä. On tärkeää, että käytettävät ohjelmistot tukevat taitossa tarvittuja erilaisia tiedostomuotoja ja nämä tiedostomuodot ovat taitto-ohjelmistossa vähintään avattavissa.

Taitossa käytettävät ohjelmistot täytyy myös olla kykeneviä joustavaan visuaaliseen muokkaukseen. Käyttäjän täytyy pystyä vapaasti siirtämään ja muokkaamaan tekstejä ja kuvia ohjelman käyttöliittymässä, jotta saadaan luotua visuaalisesti miellyttävä julkaisu. Tähän sisältyy sisällön skaalaus, tekstinkäsittely ja erilaisten objektien luominen, muokkaaminen ja poistaminen.

Sisällön automatisoinnin mahdollistamiseksi tarvitaan myös ohjelma, joka vähintään pystyy lukemaan tarvittua datalähdettä ja asettamaan datalähteestä informaatiota sivulle. Julkaisun laadun takaamisen vuoksi on myös hyödyllistä, jos datalle pystyy asettamaan ehtoja ja sitä pystyisi käsittelemään niiden ehtojen perusteella. Usein on myös tarpeellista saada ryhmitettyä dataa niin että useampi tuote jakaisivat saman paikan julkaisussa, jolloin se julkaisun paikka edustaa tuoteryhmää yhden tuotteen sijaan. Datan perusteella pitäisi myös pystyä yhdistämään oikea kuva oikealle tuotteelle, ja ohjelman täytyy pystyä tuomaan kuva valitusta lähteestä datan perusteella.

2.2.1 Adobe InDesign

Adobe on tietokoneohjelmistoja tarjoava yhtiö, jonka Creative Cloud -ohjelmistossa on tarjolla laaja valikoima erilaisia median muokkausohjelmia. Adoben laajalla valikoimalla voi luoda muun muassa grafiikkaa, videoita ja 3D-malleja, jotka usein toimivat Adoben muissa Creative Cloud -ohjelmistoissa. Esimerkiksi Adobe Illustratorin puolella luotu vektorigrafiikka voidaan tuoda muihin Adoben InDesigniin jatkokäsittelyyn ilman että sitä pitää käsitellä kuvana vektorigrafiikan sijaan (Adobe 2023a).

Adobe InDesign on yksi Adoben Creative Cloudiin kuuluva ohjelma, joka on luotu käytettäväksi erilaisten fyysisten sekä digitaalisten julkaisujen tuottamiseen (Adobe 2023b). InDesignin ominaisuuksiin kuuluvat Parent-sivujen luonti ja niihin pohjautuvat sivukohtaiset suunnittelut antavat luovuuden vapauden ohjelman käyttäjälle.

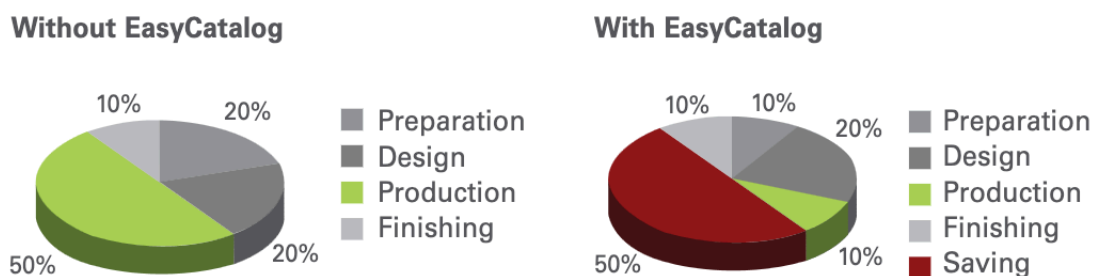
InDesigniin löytyy myös paljon eri automaatiikkaan mahdollistavia ominaisuuksia, jossa yhtenä merkittävänä ominaisuutena on ”Scripts Panel”, jonka kautta voi käynnistää itse luomia

skriptejä. Näillä skripteillä voi esimerkiksi massana käsitellä kaikki dokumentin kuvat, asettaen niihin tietyn tyylin tai rajauksen tai niillä voi massana käsitellä useita dokumentteja kerralla, jos skripti on vain siihen kirjoitettu. (Adobe 2018.)

On myös huomioitava Parent-sivujen erilaiset ominaisuudet, kuten juokseva tekstikehys, joka jatkaa tekstiä automaattisesti seuraavalle sivulle, jos teksti ei mahdu yhdelle sivulle. Parent-sivulla voi myös asettaa automaattisen sivunumeroinnin, InDesignin erikoismerkkiominaisuudella, josta löytyy myös muita dynaamisesti muuttuvia arvoja, kuten esimerkiksi päivämäärä. (Adobe 2023c.)

Tavallisesti kun InDesignilla luodaan pohja taittoa varten, tehdään dokumentti, jossa on kaikki mahdolliset erilaiset visuaaliset pohjapiirroksot halutulle julkaisulle omina Parent-sivunaan. Esimerkiksi usein sivureunoissa värilliset palkit, sivunumerot ja brändäys liitetään julkaisuiden jokaiselle sivulle, riippumatta sen sisällöstä, jolloin ne kannattaa tehdä Parent-sivulla. Joskus nämä elementit vaihtavat esimerkiksi väriä, jolloin luodaan uusi Parent-sivu uudelle värille. Joskus myös poikkeuksellisesti ei haluta Parent-sivun elementtejä ollenkaan, kuten esimerkiksi koko sivun peittävän mainoksen kohdalla tai kuvasta julkaisun keskellä. Joskus näille sivuille myös lisätään apuviivoja ohjaamaan itse sivujen sisältöä, mutta sisältöä ei Parent-sivuille liitetä. Alla olevassa kuvassa 1 näkyy esimerkki automaatiolla luodusta julkaisusta, jossa on käytetty apuviivoja.

65bit Software Ltd.:n mukaan julkaisun automatisointi säästää julkaisuun käytettävää aikaa 50 % verrattuna aikaan mikä tavallisesti käytetään käsin tehdyssä julkaisussa. Suurin osa tästä säästetystä ajasta tulee itse tuotannosta, mutta myös valmisteluihin käytetty aika pienenee, kun dataa hallitaan vain yhdessä paikassa, jonka voi yhdistää EasyCatalogiin. Kuvattu säästö on visualisoitu EasyCatalog Flyerista löytyvässä kuviossa 1. (65bit Software Ltd. 2024b.)



Kuvio 1. Säästetty aika EasyCatalogia käyttäen (65bit Software Ltd. 2024b).

EasyCatalog pystyy avaamaan InDesignissa esimerkiksi Excel tiedostoja, XML-sanomia paikallisena tiedostona tai nettiosoitteen kautta ja monessa tapauksessa myös PIM-palveluiden kautta (65bit Software Ltd. 2024c, 12–13). Erilaisien datalähteiden lukeminen mahdollistaa useamman datalähteen käytön yhdessä taitossa ja joustavuuden eri tapauksia varten riippuen tapauksien datan sijainnista.

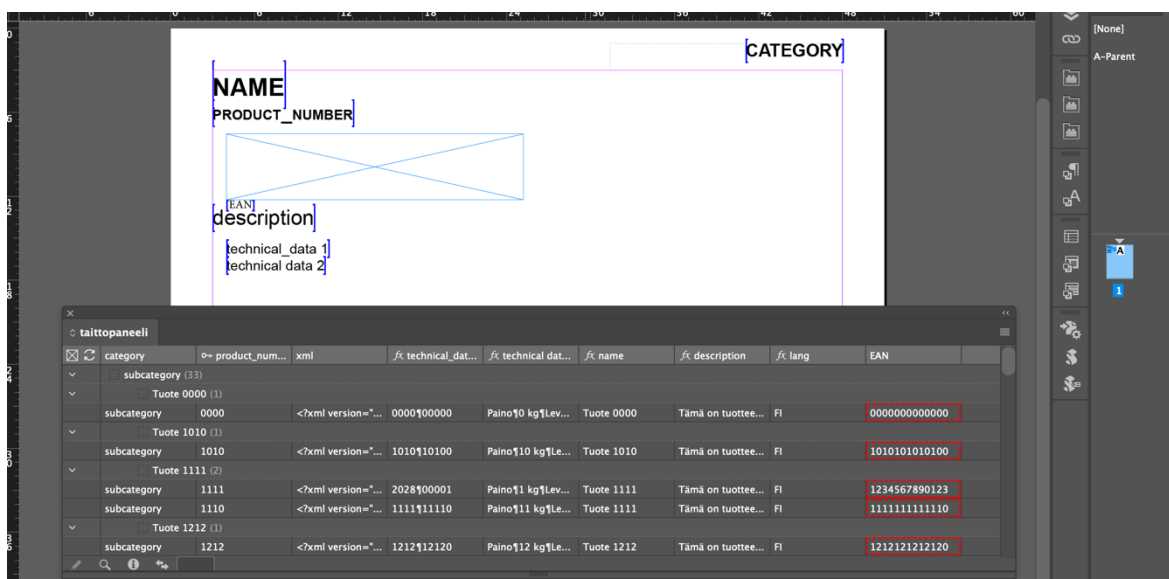
EasyCatalogin perusohjelmistoon kuuluu vain datan linkittäminen InDesign-dokumenttiin, mutta esimerkiksi tässä tapauksessa on käytössä EasyCatalogin lisämooduleja. Tässä tapauksessa on käytetty XML Data Provider -moduulia, joka mahdollistaa XML datalähteen käytön, sekä Scripting Module -moduulia, joka laajentaa InDesignin omaa ohjelmointia toimimaan EasyCatalogin kanssa (65bit Software Ltd. 2010, 5).

Yleisesti käytettävä lisämoduuli on kuitenkin Pagination -moduuli. Se lisää EasyCatalogin valikkoon "Pagination" vaihtoehdon, joka mahdollistaa pohjien tekemisen, joissa vain paria nappia painamalla, voidaan luoda koko julkaisu (65bit Software Ltd. 2024c, 71). Paneeliin sekä pohjaan voidaan luoda funktioita, jotka päättävät esimerkiksi mitä elementtiä käytetään missäkin kohdassa ja EasyCatalog lisää sivuja tarpeen mukaan, että kaikki datarivit mahtuvat taittoon.

Automaatiossa käytettävän pohjan luomiseen tarvitaan vain tieto, että mikä data halutaan liittää mihin kohtaan, jonka jälkeen voidaan EasyCatalog -paneelistä linkittää pohjaan haluttu tieto. Linkeistä muodostetaan elementtejä, jotka toimivat samalla tavalla kuin mikä tahansa muukin InDesign elementti. Jos on esimerkiksi tekstiä, sen voi linkittää omalle

rivilleen, ja sille voidaan antaa omat kappale- ja merkkityylinsä. Jos kyseessä on kuva, tehdään linkitys graafiseen kehykseen missä ei ole tekstiä ja se toimii kuten muutkin vastaavanlaiset kehykset. (65bit Software Ltd. 2024d.)

Paneelissa voi myös ryhmitellä datarivit Lisäksi EasyCatalog paneelin sisällä voi luoda funktioita, joilla voidaan joko käsitellä dataa tai yhdistellä sitä tarpeen tullen (65bit Software Ltd. 2024c, 72). Esimerkkitapuksessa on ryhmitelty tuotteet kategorian sekä tuotenimen perusteella (Kuva 2). Nämä tiedot ovat lisäksi haettu XML-lähteestä itse tehdyillä funktioilla (65bit Software Ltd. 2024a).



Kuva 2. Avattu EasyCatalog paneeli, joka on linkitetty pohjaan.

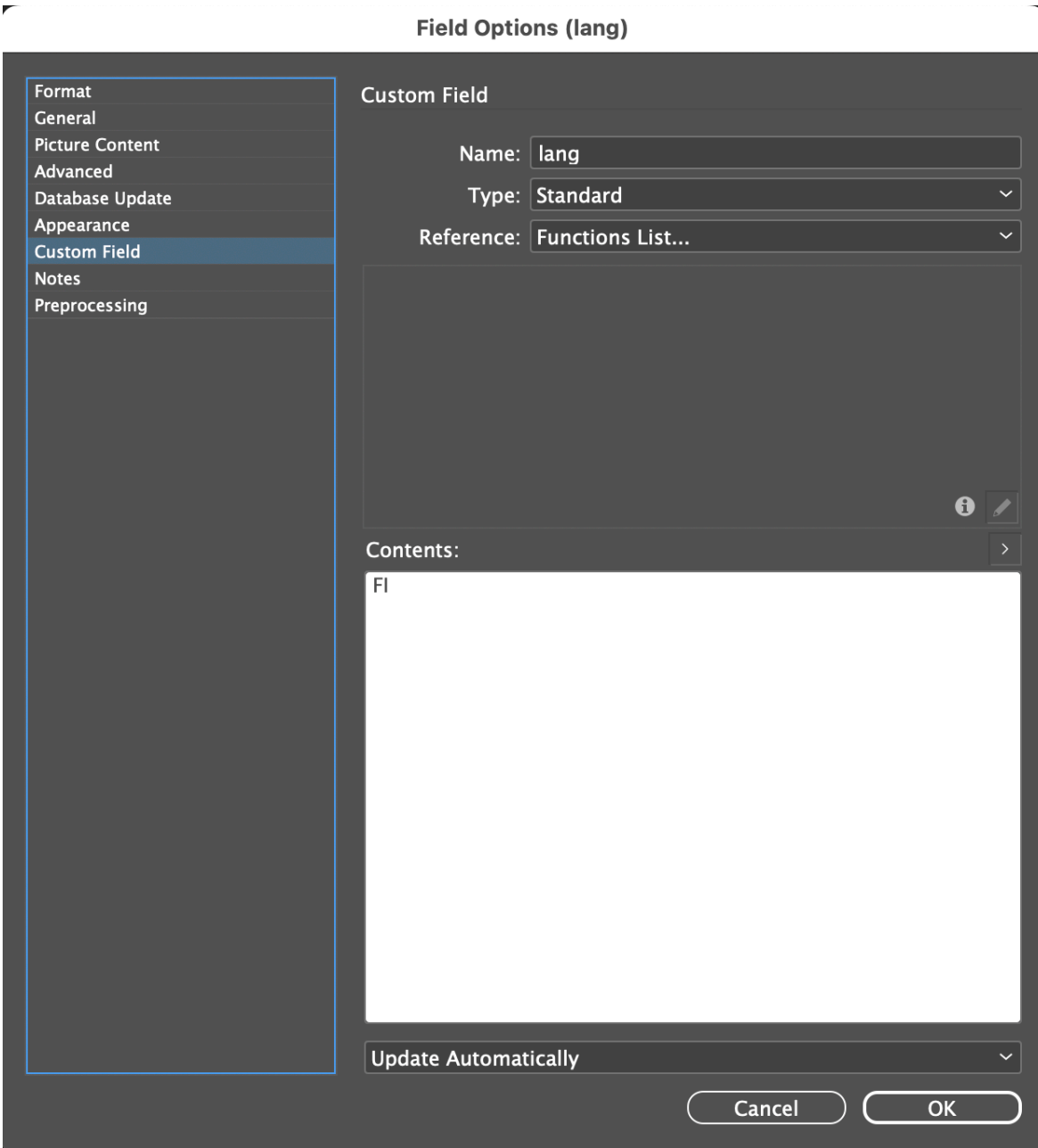
2.2.3 Ohjelmistokohtaiset ohjelmointikielet

EasyCatalog tukee LUA-ohjelmointikieltä, johon perustuvat moni sen toiminnoista (65bit Software Ltd. 2024c, 111). LUA-ohjelmointikieleen myös perustuu EasyCatalogin omat sisäänrakennetut valmiit funktiot, joita voi kehittäjä käyttää datan tulkitsemiseen ja käsittelyyn.

Paneelista, joka on yksin linkitetty InDesign dokumenttiin, käytetään opinnäytteessä termiä "taittopaneeli". Taittopaneelissa voidaan käsitellä dataa, sitä yhdistelemällä, putsaamalla tai luomalla esimerkiksi laskemalla tuotteita ryhmän sisällä tai luomalla uusia muuttujia, joita käytetään muissa EasyCatalog funktioissa. Kun datan käsittelyä tehdään paneelissa, paneeliin tehdään oma kustomoitu kenttä käsittelyä varten. Nämä kustomoidut kentät näkyvät paneelissa erikseen sillä, että kentän eli paneelin sarakkeen otsikon edessä "fx" merkintä. Tämä merkki näkyy kuvassa 2 esimerkiksi "lang" -kentän edessä.

Kuvassa 3 näkyy kyseisen "lang" -kentän asetukset. Funktiot sekä mikä tahansa mahdollinen muu sisältö kirjoitetaan "Contents:" osion alapuolelle ja EasyCatalog automaattisesti

tulkistee tekstin ilman funktiokutsuja tavalliseksi merkkijonoksi. Funktioiden tulokset luetaan myös merkkijonona automaattisesti, ellei kentän asetuksissa ole toisin ilmaistu. Näissä asetuksissa pystyy asettamaan kentän sisällön muoto. Jos kentän sisältö on numeraalinen tai grafiikkaa merkkijonon sijaan, kentän asetuksissa pitää se näin ilmaista, jos kentän arvoa halutaan käyttää numeraalisena tai grafiikkana.



The image shows a software interface titled "Field Options (lang)". On the left, a vertical sidebar lists several categories: Format, General, Picture Content, Advanced, Database Update, Appearance, Custom Field (highlighted in blue), Notes, and Preprocessing. The main area is titled "Custom Field" and contains the following fields:

- Name: lang
- Type: Standard (dropdown menu)
- Reference: Functions List... (dropdown menu)
- Contents: FI (text input field)

At the bottom of the main area, there is a dropdown menu labeled "Update Automatically" and two buttons: "Cancel" and "OK".

Kuva 3. EasyCatalog kustomoidun kentän asetukset.

Esimerkiksi, jos tapaukseen liittyy useampi datalähde, jokainen niistä tarvitsee oman paneelinsa, mutta paneelit voidaan sitten yhdistää EasyCatalog funktioilla yhteen paneeliin. Paneeli, joka sisältää noutofunktiot käytetään yleensä taittopaneelina. Kustomoituja kenttiä

voidaan myös tehdä lisäpaneelissa, mutta yleensä ne tehdään vain taittopaneelille, jotta voidaan säilyttää automaation ylläpitäminen mahdollisimman helppona.

EasyCatalogissa voi myös kirjoittaa LUA-ohjelmointikieltä suoraan sellaisenaan, jos olemassa olevat funktiot eivät taivu tekemään sitä, mitä tapauksen kohdalla vaaditaan. Olemassa olevat funktiot voidaan myös kirjoittaa LUA-ohjelmointikielellä. Esimerkiksi tavallinen EasyCatalogin funktio, joka hakee sisällön toisesta kentästä, FIELDSTR() voidaan kirjoittaa LUA-ohjelmointikielellä FIELD.get().content() (65bit Software Ltd. 2024a ja 65bit Software Ltd. 2024f, Class FIELD).

Jos paneelissa käytetään LUA-ohjelmointikielen funktiota, joutuu kuitenkin sen kustomoidun kentän sisällä käyttämään pelkästään LUA-ohjelmointikielen funktioita. Kustomoidun kentän tyyppi asetetaan silloin "Advanced". EasyCatalog funktioita käyttäessä kentän tyyppi on "Standard" ja paneeli kääntää EasyCatalog funktiot LUA-funktioiksi (Kuva 3). Jos haluaa käyttää näitä funktioita sekaisin, pitää tehdä erilliset kentät eri funktioille tai kääntää EasyCatalog funktiot LUA-ohjelmointikieleen.

EasyCatalogin Scripting Module -moduuli sallii käytettäväksi myös käytettäväksi AppleScriptin, JavaScriptin ja Visual Basicin (65bit Software Ltd. 2010, 7). Näitä kieliä voi käyttää InDesignin omassa Scripts paneelissa (Adobe 2018, 1). InDesign ohjautuu myös näiden kielten avulla antaen kehittäjälle mahdollisuuden luoda skriptin, joka viittaa InDesignin ja EasyCatalogin ominaisuuksiin suoraan.

ExtendScript on Adobe Systemsin kehittämä kieli, joka on JavaScriptista kehitetty variaatio. ExtendScript toimii hyvin samalla tavalla kuin JavaScript, muutamaa poikkeusta lukuun ottamatta ja ExtendScriptin työkaluista löytyy myös Adoben ohjelmistossa käytettäviä ilmaisia. Tätä JavaScriptin alalajia eli ExtendScriptiä on käytetty myös opinnäytteesä. (Adobe 2024b.)

3 Toteutus

3.1 Lähtötilanne

Toteutuksen tarkoituksena vähentää käyttäjän tekemää manuaalista syöttöä lisäämällä jo olemassa olevaan automatisoituun prosessiin julkaisuversion vaihtaminen sekä uuden version automaation käynnistäminen. Vaikka EasyCatalog ei yksinään taivu monen julkaisuversion automaattiseen taittoon, Adobe InDesignissa käytettävällä ExtendScriptillä voi automatisoida niitä osia, mitkä EasyCatalogin sisäänrakennetuista ominaisuuksista puuttuu. Tavoitteena on kirjoittaa skripti, jonka käyttäjä käynnistää vain kerran ja kaikki tarvittavat versiot taittavat automaattisesti. Tämän tavoitteen saavuttamiseksi seuraavat toiminnot täytyy tapahtua automaatiolla:

- ennalta rakennettujen EasyCatalog-paneelien avaus
- paneelien datalähteiden varmennus ja paneelin päivitys datalähteen perusteella
- toimivien taittoasetusten valitseminen
- halutun InDesign-pohjan avaus, taitto, tallennus ja sulkeminen
- haettavan datan kielen vaihtaminen
- taiton päätteeksi oletusasetusten takaisin tuominen manuaalista taittoa varten
- automaattisesti avattujen materiaalien sulkeminen.

Kun kaikki yllä olevista askeleista on automatisoitu, on tämän toteutuksen tavoite saavutettu. Osa askeleista on jo aiemmin toteutettu, sillä taittaminen tapahtuu jo automaattisesti, mutta kaikki muut askeleet on automatisoitava InDesignin ExtendScriptillä, joka viittaa myös EasyCatalogin Scripting -moduulin avulla EasyCatalog paneeleihin.

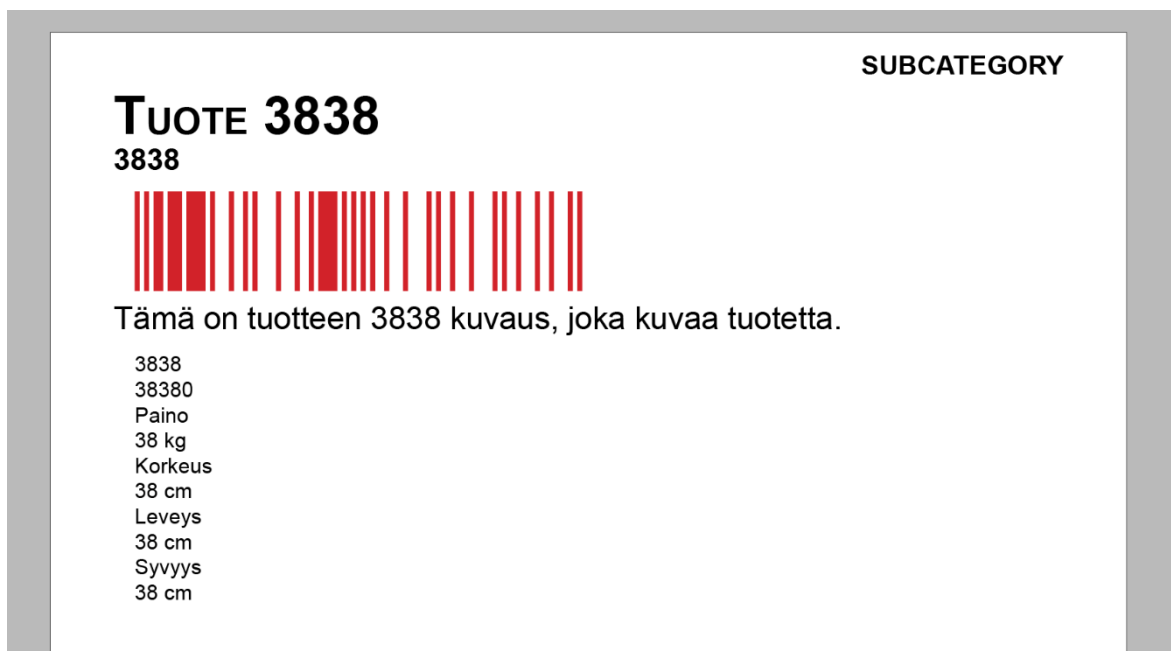
Kuvitteellisella asiakkaalla on tarve automatisoida tuotekorttien taittoa. Tuotekortit taitetaan kerran päivässä viidellä eri kielellä ja ne taitetaan kaikista uusista tai päivitetystä tuotteista. Tuotekortteja ei tarvitse automaation jälkeen muokata käsin, sillä tuotekorttien listatessa vain teknisiä tietoja, ei niiden tarvitse olla yksilöllisiä. Tuotekortteja ei käytetä myynnin helpottamiseen vaan tuotteen kuluttaja voi tarkistaa tuotekortista tuotteen teknisiä tietoja esimerkiksi sen rajoituksista tai yhteensopivuuksista, joten visuaalisuus ei ole tärkeässä asemassa.

Eri kieliversiossa tuotekortin käytettävät yksiköt sekä otsikot muuttuvat datan kielen lisäksi. Automaatiolla luodut eri kieliset tiedostot nimetään päätteellä, jossa tiedoston pääte on taitettu kieli. Tuotekortit ovat PDF-tiedostoja ja kielen lisäksi ne nimetään tuotekoodilla. Koska kuvitteellinen asiakas on jo entuudestaan tuttu automaation kanssa, asiakkaalta löytyy sekä EasyCatalog paneelit, että Adobe InDesign pohja, jotka ovat olleet käytössä ahkerasti.

Kumpikaan data tai pohja eivät tarvitse enää käsin viimeistelyä automaation jälkeen, sillä olemassa oleva ratkaisu on ollut riittävä.

3.1.1 InDesign-pohja

Kuvassa 4 näkyy valmiiksi taitettu tiedosto kuvitteellisesta datasta. Valmiissa tiedostossa näkyy tuotenimi, sen koodi, viivakoodi, kuvaus sekä simuloidut tekniset tiedot. Lisäksi ylänurkassa on tuotteen kategoria. Teknisiä tietoja on vaihteleva määrä jokaisella tuotteella ja jokainen tekninen tieto tulee ensin otsikon kanssa ja sen alapuolella on sen tiedon arvo.



Kuva 4. Suomen kielellä taitettu tuote InDesign pohjaan.

InDesign pohjana toimii perus InDesign tiedosto eli .indd-päätteellä oleva tiedosto. Koska julkaisun sivumäärää ei voida ennalta määrittää varmaksi, template-tiedoston eli .indt-päätteisen tiedoston sijasta käytetään tavallista .indd-päätteistä tiedostoa pohjana. Template-tiedosto ei toiminut siinä kohtaa, kun tuotekortin sivumäärää ei tiedetä etukäteen, vaikka template-tiedosto onkin nopeampi taitettava. Siksi tavallinen InDesign-tiedosto, joka käyttää taitossaan EasyCatalogin Formatting Rules -taitotapaa skriptin kautta on tarpeellinen.

Pohjan Parent-sivulla on pelkästään pääsääntöisesti käytetty tekstikehys, joka on asetettu pääkehykseksi. Pääkehystenä sen sisältö jatkuu seuraavalle sivulle, jos tila ei riitä. Lisäksi parent-sivulla löytyy tekstikehys, johon on linkitetty kategoriainformaatio taittopaneelistä. Tälle tekstikehykselle on asetuksiin laitettu päivittymään Furniture Datan mukana, joka tarkoittaa siis sitä, ettei se päivity ennen kuin sille annetaan lupa päivittyä.

Parent sivuilla oleville elementeille täytyy antaa Furniture Data -asetus, jotta Parent-sivulla oleva elementti päivittyisi. Furniture Datan voi laittaa päivittymään taiton yhteydessä, mutta sen päivittyessä se lakkaa olemasta Parent-sivun elementti ja muuttuu tavalliseksi elementiksi, joka sijaitsee numeroidulla sivulla, Parent-sivun sijasta. Kun elementti ei ole enää identtinen jokaisella sivulla, vaan siihen linkitetty EasyCatalogin kautta tuleva tieto on erilainen jokaisen tuotteen kohdalla, se ei enää voi olla Parent-sivun elementti.

Myöhemmin päivittämistä käytetään automatisoidussa taitossa, jossa taittoa käsitellään vielä automaation jälkeen. Jos esimerkiksi taittaja päättää automatisoinnin jälkeen lisätä tai poistaa tuotteita, sivulle 4 liikkuneet tuotteet voivat olla eri kategorian alla, jolloin halutaan päivittää Furniture Data vasta sitten kun käsin muokkaukset on tehty. Opinnäytteesä Parent-sivulla sijaitsevat kategoriakehys ja päättekstikehys ovat ainoat elementit, mitä pohjassa näkyy ennen kuin automaatiolla tuodaan loput elementit ja loput elementeistä on Formatting Rules -valikon sisällä.

3.1.2 Formatting rules

Kankeasti EasyCatalogin ominaisuuden, ”Formatting Rules”, suomeksi käännettynä, muotoilun säännöt, tarkoittavat InDesign pohjassa sitä mitä otsikko sanoo. EasyCatalogin avulla pystyy luomaan niin sanottuja sääntöjä. Säännöt ovat elementtejä, joille annetaan ehtoja, millä ne taittavat. Yleinen käyttö näille säännöille ovat tapaukset, joissa tuotteissa on paljon isoja variaatiota yhden datalähteen sisällä objektien sijaintien ja tyylien kanssa. Erilaisten sääntöjen tekemistä kannattaa välttää, sillä mitä enemmän sääntöjä on, sitä raskaammaksi pienten muokkausten ja päivityksien tekemisestä tulee. Muutokset ja päivitykset joudutaan tekemään kaikkiin käytössä oleviin elementteihin.

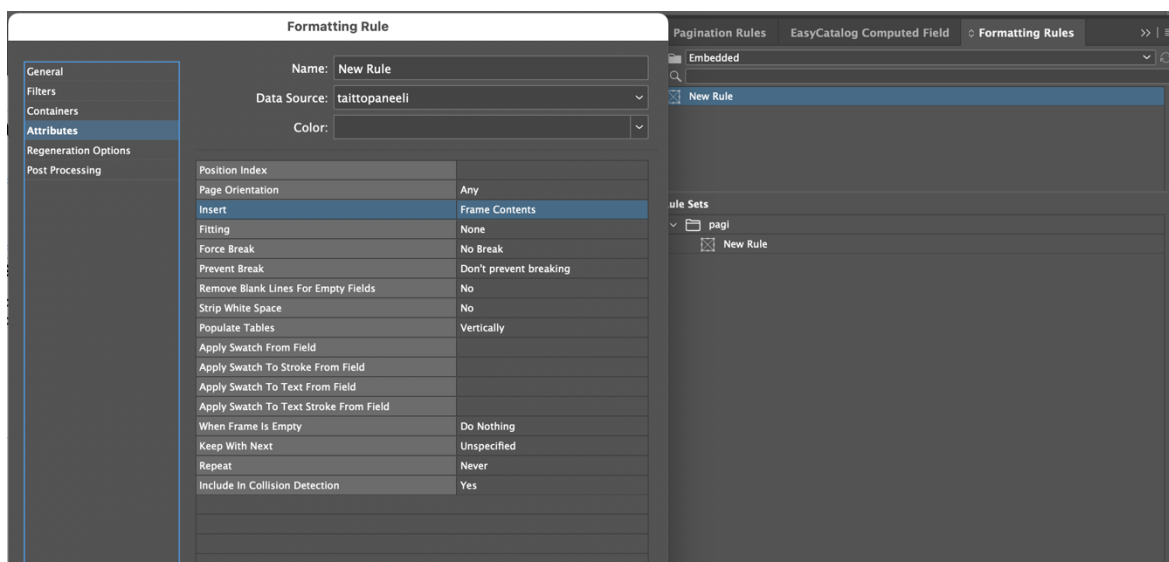
Näistä säännöistä voidaan muodostaa settejä, jotka voivat sisältää useamman säännön eri ehdoilla. Valmiin sääntösetin tai säännön itsessään voi myös yhdistää pohjassa olemassa olevaan tekstikehykseen, jolloin sääntösetin tai säännön sisältö tulee taittaessa kyseiseen kehykseen. Tavallisempi tapa käyttää sääntösettejä on kuitenkin EasyCatalogin taittotoinnin valikon kautta valitsemana taittotapana. Taittovalikossa näkyy vain sääntösetit, mutta tällä voi esimerkiksi taittaja itse valita oikean setin oikeaa tuoterypystä varten, jos halutaan käyttää samaa pohjaa erilaisiin julkaisuihin.

Säännöt

Tämän tapauksen pohjassa on vain yksi sääntö, joka muodostaa yhden sääntösetin. Kyseessä olevassa säännössä on linkitetty taittopaneelistä tuotteen informaatiota kuten on kuvitettu kuvassa 2. Sääntöön eli elementtiin sisältyy otsikkona toimiva tuotteen nimi, jonka alapuolella on loput tuotteen informaatiosta, sen kuvaus, tekniset tiedot ja erityisenä

lisäyksenä riville lisätty kuvakehys, johon on linkitetty tuotteen viivakoodi (Kuva 2). Taitetuna nämä tiedot täyttyvät paneelista löytyvällä tuotteiden datalla (Kuva 4).

Tekstikehykselle on asetettu toistuvaksi "Once per Record", joka tarkoittaa, että tekstikehys toistuu niin monta kertaa kuin paneelista löytyy rivejä. Lisäksi tekstikehykselle on asetettu arvo, jotta kehys itse ei tule taittoon, vaan pelkästään sen sisältö. Silloin taiton jälkeen kaikki taitettu sisältö on saman päätekstikehyksen sisällä tekstinä, eikä ankkuroituna objektina, viivakoodi grafiikkaa lukuun ottamatta (Kuva 5). Tämä mahdollistaa sen, että yhden tuotteen tiedot voivat jatkaa seuraavalle sivulle, jos ne eivät mahdu yhdelle sivulle. Elementin sisältöön lisätään kappaleenvaihto muun datan perään, jotta useamman tuotteen tulostuessa samaan tiedostoon, seuraavan tuotteen otsikko ei tulostu samalle riville kuin edellisen tuotteen viimeinen rivi.



Kuva 5. Formatting Rules ja säännön attribuutit.

3.1.3 EasyCatalog paneelit

EasyCatalogin avulla luoduista datapaneeleista tärkeimmäksi muodostuu itse tuotepaneeli. Ilman tuotepaneelia ei itse taitto onnistu, ellei palata takaisin perinteiseen taittoon ilman automaatiota. EasyCatalog-paneeleita luodaan niin monta kuin eri datalähteitä löytyy, joka tarkoittaa sitä, että XML-tiedostojen määrän ollessa kolme, on paneeleita myös kolme.

Tuotepaneelista löytyy kaikki tuotteet sekä niiden data. Tuotepaneelissa määritellään taitettavaksi haluttu data omiin kenttiinsä ja siihen tuodaan myös muiden datalähteiden info-maatio lisäpaneeleista. Tuotepaneelista löytyy kieliversiointi attribuuttien sisällöille niille attribuuteille, joilla kieliversiointia löytyy. Ilman kieliversiointia optiota kielelle ei löydy. Rakenne- ja suodatus halutuille taitetuille tiedoille viittaa sekä kieliversioiduttomiin että

kieliversioituihin valitulla kielellä. Kieli valitaankin kustomoidussa kentässä, jossa käyttäjä vaihtaa kentän sisällön eri kielten käyttämistä avainten vaihtoehdoista. Tapauksessa, jossa kieliavain on syötetty väärin tai ei ollenkaan, taitettava sisältö ei ole enää kieliversioitu vaan se taittaa vain kieliversioimattomat tuotteen attribuutit, kääntämättä attribuuttien otsikoita taikka hakematta kategorioiden nimiä. Skriptin käyttöönotossa käyttäjän virhesyöttö estetään, sillä tämä osuus on automatisoitu.

Lisäpaneelita onkin kaksi muuta kappaletta. Kuva 6:ssä näkyvästä alemmasta paneelista löytää kaikki käännökset olemassa oleville attribuuteille. Tässä paneelissa tuotteiden sijasta, jokainen rivi on attribuutti ja sen tuottavat arvot ovat attribuutin käännöksiä. Attribuutin käännökset on hyvä olla erillisessä datalähteessä kuin tuotteet, sillä jos käännökset olisivat tuotetietojen seassa, niiden täytyisi löytyä jokaisen tuotteen kohdalta erikseen. Isomman datamassan kohdalla ne suurentaisivat massaa entisestään ja lisääisivät latausaikaa.

The image shows two screenshots of software panels. The top panel is titled "languages" and contains a table with two columns: "lang" and "value". The bottom panel is titled "attribuutti käännös" and contains a table with three columns: "key", "FI", and "EN".

lang	value
eng	EN
fin	FI

key	FI	EN
width	Leveys	Width
height	Korkeus	Height
weight	Paino	Weight
description	Kuvaus	Description
name	Nimi	Name
code	Koodi	Code
EAN	EAN	EAN

Kuva 6. Lisäpaneelit kielille ja attribuuttien käännöksille.

Skriptin avuksi löytyy vielä yksi paneeli, joka on luotu nimenomaan vain skriptiä varten. Kuvas 6 tämä paneeli näkyy nimellä "languages". Paneeli on luotu excel-tiedostosta, josta löytyy vain kaksi saraketta. Toisessa sarakkeessa löytyy kielen avain, joka on sama kuin XML-tiedostossa käytetty optio ja toisessa sarakkeessa on vapaamuotoinen kielen otsikko, joka on luotu siksi että EasyCatalog tarvitsee vähintään kaksi saraketta paneelin luomiseen.

Kyseisessä excel-tiedostossa voi kontrolloida kaikkia haluttuja skriptin taittamia kieliversioita, lisäämällä tai poistamalla siitä rivejä.

3.1.4 Datalähde

Käytössä oleva data on XML-tiedostomuodossa, joista on luotu jokaisesta tiedostosta oma EasyCatalog paneelinsa. Eri tiedostot sisältävät erilaista dataa liittyen joko itse tuotteisiin tai tuotteisiin liittyviin attribuutteihin. Tärkein XML-tiedosto on itse tuoteinformaation sisältävä XML-sanoma, jossa löytyy kaikki tuotteet sekä niihin tuotteisiin liittyvät attribuutit XML-rakenteen sisällä. Kaksi muuta XML-tiedostoa sisältävät kategoriat, niiden eri kielten käännökset ja toinen sisältää attribuuttien otsikoiden käännökset. Tämä tapa esittää tuotetietoa XML-lähteessä on hyvin tavallinen, sillä XML-tiedosto on yleensä vain tuloste datasta jostain toisesta olemassa olevasta järjestelmästä. Silloin kun EasyCatalog ei pysty yhdistämään suoraan järjestelmään, XML-datalähde on yksi vaihtoehto Excel datalähteen rinnalla.

Suuri vaihtelu datan sisällössä tuotteesta toiseen tekee myös esitettävän datan pituuden arvioinnista hankalaa. Tässä datassa kuitenkin XML:ssä oli helposti poimittavissa tarvittavat tulostettavat tiedot attribuuttien omista optioista, sillä kaikki tulostukseen halutut tiedot ovat saman option sisällä.

XML-tiedostot sisältävät kaiken tuotedatan sekä niihin liittyviin attribuuttien datan. Tässä esimerkissä attribuuttien tiedoista halutaan vain käännös, mutta joissakin tapauksissa voidaan haluta muutakin tietoa, joka on yhdistetty attribuuteille, kuten attribuutin kentän tyyppistä datajärjestelmän sisällä. Esimerkiksi sieltä voi saada myös tiedon siitä, jos attribuutti kuuluu johonkin ryhmään tai jos attribuutti on järjestelmässä monivalinta- vai tekstikenttä. Yksi käyttötapa attribuutin, missä voidaan hyödyntää attribuutin kenttätyyppiä, on esimerkiksi tapaus missä kenttätyyppi on numeraalinen ja sille on yhdistetty yksikkö kuten paino tai valuutta, jotka vaihtuvat kielen mukaan "kilogrammasta" englanninkieliseen käännökseen "kilogram". Joskus halutaan käyttää myös täysin eri yksikköä. Tässä esimerkkitapauksessa eri yksiköille löytyy kuitenkin pääte tuotedatan seasta itsessään ja informaatiota attribuutin tyyppistä ei tarvita, kun datasta on täysi hallinta kokeilussa.

Tärkeintä XML-tiedoston rakenteessa on se, että data, jota käytetään paneelin avaininformaationa, eli data, jota käytetään jokaisen rivin muodostamiseen, on XML-rakenteessa saman tasoista tietoa. Eli kun avaintietona käytetään tuotteen ID-numeroa, ID-numeron täytyy löytyä jokaisen halutun datarivin XML-tiedosta samantasoisena tietona kuin kaikkien muiden rivien tietona. Kaikkiin ID-numeroihin ei voida viitata, jos ei tiedetä, kuinka syvällä XML-rakenteen sisällä se on.

Tästä syystä XML-rakenne on muodostettu niin että kategoria rakenne ei tule XML-rakenteen perusteella, vaan sen sijaan se löytyy tuotteen sisältä attribuuttina. Käytetyssä XML-tiedostossa löytyy 39 eri tuotetta, joilla on oma XML-node julkaistavan datan sisällä. Tämän tuote-noden sisällä löytyy kaikki informaatio, joka halutaan taittaa julkaisuun (Kuva 7). Tällä tavalla kaikkiin tuotteisiin voidaan viitata vain yhdellä tarkalla XPath -lausekkeella, jolla paneeli voidaan muodostaa. Attribuuttien oma XML tiedosto toimii samalla tavalla, missä jokainen attribuutti on paneelissa oma rivinsä.

Kuten kuvassa 7 näkyy, kaikki tuotteen attribuutit menevät samalla nimellä mutta jokaisella attribuutilla on optio "id", jolla voi hakea XPath-lausekkeella tarkemmin oikean tiedon. Lisäksi joiltakin attribuuteilta löytyy lisäoptio "language" eli "kieli", jolla voidaan hakea oikea kielimuoto. Kolmas lisäoptio on epätavallisempi ratkaisu, joka toimii attribuuttien ryhmittelemiseen silloin kun tuotteelta löytyy enemmän informaatiota kuin mitä halutaan julkaisuun tuoda. Optio "group" eli "ryhmä" kertoo mitkä kaikki attribuutit kuuluvat teknisiin tietoihin. Tämä optio voi vaihdella ja ryhmän sisällä olevat attribuutit voivat muuttua tuotteesta toiseen, mutta viittaamalla ryhmään "technical_information" XPath-lausekkeella attribuutin id:n sijaan, voidaan hakea kerralla kaikki informaatio mitä yhdelle tuotteelle halutaan, riippumatta siitä mitä attribuutteja teknisissä tiedoissa on. Tämä lisä optio on hyödyllinen silloin kun julkaisuun haluttujen attribuuttien määrä on joko hyvin pitkä, tuntematon tai jatkuvasti muuttuva.

```
<publication>
  <product>
    <categories>
      <category use="PRINT">subcategory</category>
    </categories>
    <attribute id="product_number">1111</attribute>
    <attribute id="EAN">1234567890123</attribute>
    <attribute id="date" group="technical_information">2028</attribute>
    <attribute id="code" group="technical_information">00001</attribute>
    <localized>
      <attribute id="name" language="FI">Tuote 1111</attribute>
      <attribute id="name" language="EN">Product 1111</attribute>
      <attribute id="description" language="FI">Tämä on tuotteen 1111 kuvaus, joka kuvaa tuotetta.</attribute>
      <attribute id="description" language="EN">This is product 1111's description, which describes the product.</attribute>
      <attribute id="weight" group="technical_information" language="FI">1 kg</attribute>
      <attribute id="weight" group="technical_information" language="EN">1 lb</attribute>
      <attribute id="height" group="technical_information" language="FI">1 cm</attribute>
      <attribute id="height" group="technical_information" language="EN">1 in</attribute>
      <attribute id="width" group="technical_information" language="FI">1 cm</attribute>
      <attribute id="width" group="technical_information" language="EN">1 in</attribute>
      <attribute id="depth" group="technical_information" language="FI">1 cm</attribute>
      <attribute id="depth" group="technical_information" language="EN">1 in</attribute>
    </localized>
  </product>
  <product>
    <categories>
      <category use="PRINT">subcategory</category>
    </categories>
    <attribute id="product_number">2222</attribute>
    <attribute id="EAN">2234567890223</attribute>
    <attribute id="date" group="technical_information">2028</attribute>
    <attribute id="code" group="technical_information">00002</attribute>
  </localized>
</publication>
```

Kuva 7. Tuotedatan XML-tiedoston esimerkkiotos tuoterakenteen kanssa.

Datan tuominen paneeliin

Paneeli on luotu EasyCatalogin XML Data Source -pohjalla, johon annetaan XPath, joka viittaa XML-nodeen, jolla halutaan muodostaa rivi. Kuvassa 8 näkyvä viittaus "Record XPath" osiossa määrittää sen, kuinka monta riviä datapaneeliin tulee. Sen alapuolella voidaan määrittää valmiiksi kentät, mitä halutaan tuoda paneeliin. Jokin näistä tiedoista voidaan sitten määrittää avaintiedoksi, joka täytyy olla uniikki kaikille datariveille. Jos tieto ei kuitenkaan ole uniikki, voidaan määrittää useampi avainkenttä, mutta on aina suositeltavaa, että tuotteelta löytyy vähintään yksi sen tunnistamiseen tarkoitettu tieto. (65bit Software Ltd. 2007.)

Data Source Configuration

Name: OK

Source: Cancel

Macinto...di-EasyCatalog-kehitys-2023-02-03:ideaali XML:testts.xml

Reveal Choose...

Record XPath

Evaluate

Examples:

	Name	XPath	Example
	category	categories/category/text()	
↔	product_number	attribute[@id="product_num...	
	xml	.	<xml>
	EAN	attribute[@id="EAN"]/text()	

New..
Edit..
Delete

Kuva 8. XML-datalähteen konfiguraatio EasyCatalog paneelia luodessa.

Muut konfiguraatiossa tuodut kentät ovat muuttumattomia tietoja, kuten kategoriat ja EAN numero. Datarivikohtainen eli tuotekohtainen XML-noden voi myös tuoda erikseen omaan kenttäänsä, vaikka se ei ole aina tarpeellista. Tässä tapauksessa tuodaankin XML-noden

kenttänä ja jätetään kaikki kieliversioidut attribuutit tuotavaksi EasyCatalog funktioilla, sillä muutoin pitäisi tuoda kaikki eri kieliversiot omina kenttinaan.

3.2 Ohjelmointi

Paneeliin tehdyt funktiot sallivat datan hakemisen XML-tiedostosta attribuutin optioiden perusteella. Sen sijaan, että haetaan jokaista yksittäistä attribuuttia erikseen, haetaan massa attribuutteja kerralla yhdellä EVALUATEXPath() EasyCatalog funktiolla. Samaan funktioon on liitetty attribuutin avaimen haku sekä attribuutin arvon haku, ja myöhemmin tuo avain muutetaan käännökseksi, joka haetaan lisäpaneelistä käyttäen attribuutin avainta sekä kielikentässä asetettua kieliavainta.

EVALUATEXPath() funktion muuttujina on EasyCatalog kenttä, XPath ja mahdollinen erotin eri arvoille. EasyCatalog kenttä johon viitataan, on kenttä mihin on tuotu tuotteen XML-node (65bit Software Ltd. 2024a). Tällä voidaan tuoda XPath-lausekkeella tietoja, samalla tavalla kuin paneelin konfiguraatiossa, mutta koska käytämme EasyCatalog funktiota, voimme yhdistää useamman funktion XPath:iä muodostaessa.

Tavallisesti XPathiä käyttäessä voi käyttää vain yksinkertaisia ehtoja tuomalla kaikki data, joka sopii yhteen lausekkeeseen annettuihin hakuehtoihin. Useamman funktion avulla varmistetaan, että tuomme juuri sen tiedon mikä julkaisuun halutaan, tuomatta ylimääräistä tietoa datalähteestä. EasyCatalogin CONCAT() funktiota käyttäen voi muodostaa XPath lausekkeen joka muuttuu muualla asetetun arvon mukaan, yhdistämällä osan joka ei muutu, osaan jonka halutaan muuttuvan eri julkaisu versiota taittaessa.

Paneelistä löytyessä kenttä, jossa on arvona pelkästään sen hetkisen taitettavan kielen arvo, "EN", voidaan kenttään viitata FIELDSTR() funktiolla. XPathin voi silloin muodostaa ja sisällyttää EVALUATEXPath() funktion sisäpuolelle vastaavasti: CONCAT('/product/localized/attribute[@id="',FIELDNAME(),"][@language="',FIELDSTR(lang),"]'/text(')). Tällä tavalla voidaan hakea halutulla id:llä arvona ainoastaan valittu kieli.

Funktiossa FIELDNAME() funktio viittaa kentän nimeen, jolloin haettava attribuutti id:llä on sama kuin kustomoidun kentän annettu nimi. EVALUATEXPath() funktion sisälle lisäämällä eri arvojen erottimen, saadaan myös haettua kaikki "technical_information" ryhmään kuuluvien attribuuttien arvot halutulla erottimella. Lisäämällä XPath lausekkeeseen "ja" komennon myös attribuuttien "id" option tuomiseen saadaan myös tuotua attribuuttien avain, jota voi käyttää attribuutin käännöksen hakemisessa.

Samaa logiikkaa, useamman funktion käytöstä sisäkkäin, voi käyttää käännösten haussa XREFFIELD() funktion avulla sekä tavallisten attribuuttien sisältöjen haussa on

hyväksikäytetty myös EasyCatalog funktiota CONCAT(), joka toimii yhdistämään eri arvoja funktioiden sisällä. Silloin voidaan antaa haettavaksi arvoksi avain + kieli, riippumatta funktiosta. XREFFIELD() funktion muuttujina on kenttä, josta etsitään, arvo, mitä etsitään, kenttä, jonka arvo halutaan palauttaa ja mistä paneelista halutaan tuoda tietoa. Viimeinen muuttuja on vapaaehtoinen. Jos sen jättää tyhjäksi, funktio toimii vain oman paneelin sisällä.

Tätä tietoa käyttäen voi muodostaa funktion: XREFFIELD(key, "attribuutin avain", FIELDSTR(lang), attribuutti käänös). Tässä funktiossa kuitenkin "attribuutin avain" on määritettävä erikseen kun haetaan käännöksiä "technical_information" attribuuteille. Koska attribuuttien määrää ei tiedetä eikä tarkkaa listaa attribuuteista ole, on tämä järkevintä toteuttaa LUA-ohjelmointikielellä sen sijaan. XREFFIELD() funktion voi kääntää LUA kieleen "DATASOURCE.get('attribuutti käänös'):xreffield("key","attribuutin avain",field(lang)):content()", jonka voi sijoittaa for-silmukan sisään toteutumaan niin monta kertaa kuin avaimia täytyy hakea.

Tällä tavalla paneeli on luotu niin, että vain "lang" nimisessä kentässä löytyvällä arvolla voi muokata koko julkaisun kieltä, sillä kaikki data joka vaatii kielikohtaista hakua, viittaavat samaan kenttään. Kun "lang" -kentän arvo vaihtaa, vaihtuu koko taiton sisältö. Mikäli "lang" -kentässä on arvo, jota ei löydy datalähteestä, tulee kielikohtaiset kentät tyhjinä ja niiden arvoista ei taitu mikään. Manuaalisessa taitossa, tämä arvo vaihdetaan käsin, kun versiota vaihdetaan. Usean version automatisointiin, "lang" -kentän sisällön vaihto ExtendScriptillä on kriittistä tässä toteutuksessa.

Nykyisen kuvitteellisen asiakkaan EasyCatalogilla automaatioitu työprosessin käynnistäminen skriptin avulla aloitetaan, kuten mikä tahansa muu InDesign skripti. Ohjelmointikielenä valittu ExtendScript vaatii JavaScript tiedoston päätteellä ".jsx", jonka voi luoda eri tekstieditoreissa tai suositeltavasti koodieditoreissa, mutta tässä tapauksessa käytettiin Visual Studio Code -ohjelmaa.

3.2.1 Skriptin muuttujat

Jotta sama skripti olisi helppo muokata jatkossa muihin käyttötarkoituksiin, sen alussa muodostetaan globaalit muuttujat. Yksi tärkein muuttuja tavallista taittoa varten on taitettavan paneelin nimi, johon voidaan sitten viitata myöhemmin skriptissä. Toinen, aivan yhtä tärkeä muuttuja on taittoon käytettävä EasyCatalog paneelin konfiguraatio, johon on asetettu esi-asetukset ryhmittelystä sekä itse taiton asetukset.

Koska taitossa käytetään myös toista paneelia lähteenä, jossa on listattuna tässä tapauksessa kielet, on sekin nimettävä globaaleissa muuttujissa. Tästä lähdepaneelistä on

muuttujissa nimetty myös lähdekentän nimi sekä taittopaneelin kentän nimi mihin on lähdepaneelin arvo tarkoitus asettaa.

Lopuksi vielä varmuuden vuoksi asetetaan muuttuja, jossa nimetään InDesignissa tehdyt esiasetukset PDF-tiedoston vientiin, sekä konfiguraatio mihin paneeli halutaan palauttaa taiton jälkeen, jos skriptin käyttämä konfiguraatio ei ole sama kuin manuaalisessa käytössä ollut konfiguraatio. Kaikki muuttujat opinnäytteeseen on kuvassa 9.

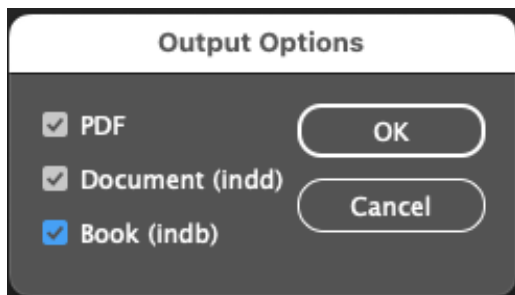
```

17 pdfPreset1 = "[Smallest File Size]"; // PDF Export Preset
18 listDS = "languages"; // Resource panel
19 dataDS = "taittopaneeli"; // Target panel
20 fieldToSet = "lang"; // Field in the target panel
21 fieldToGet = "value"; // Field in the resource panel
22 DSconfig = "script"; // Configuration for the script use
23 DSconfigRet = "manuaali"; // Configuration for manual use

```

Kuva 9. Muuttujat .jsx tiedoston alussa.

Loput tapauskohtaiset muuttujat tulevat käyttäjän omalla ohjauksella skriptin nostamalla valikoilla, jotka nousevat näkyviin heti kun skriptin käynnistää. Kuvassa 10 näkyvässä valikossa on annettu mahdollisuus valita haluttu tiedostomuoto, antaen kolme eri vaihtoehtoa, joista voi valita useamman kerrallaan halutessaan. Kaikki valitut tiedostotyypit tallentuvat saman kansion alle. Lopputuloksena asiakas tarvitsee PDF-tiedoston, mutta usein automatisoinnin jälkeen käyttäjät haluavat itse vielä muokata tiedostoja.



Kuva 10. "Output Options" valikko tuodaan funktiolla myDisplayDialog().

Kun käyttäjä on valikosta valinnut halutut tiedostomuodot, funktiot File.openDialog() ja Folder.selectDialog() on ohjelmoitu kysymään käyttäjää valitsemaan InDesign tiedoston, jota käytetään pohjana, sekä kansion, minne halutaan lopulliset taitetut tiedostot tuottavat. Molemmat näistä funktioista avaavat käyttöjärjestelmäkohtaisen resurssienhallinta ikkunan. Käyttäjä valitsee halutun sijainnin ja palauttavat kansiopolon nimetylle muuttujalle.

3.2.2 Taiton ohjelmointi

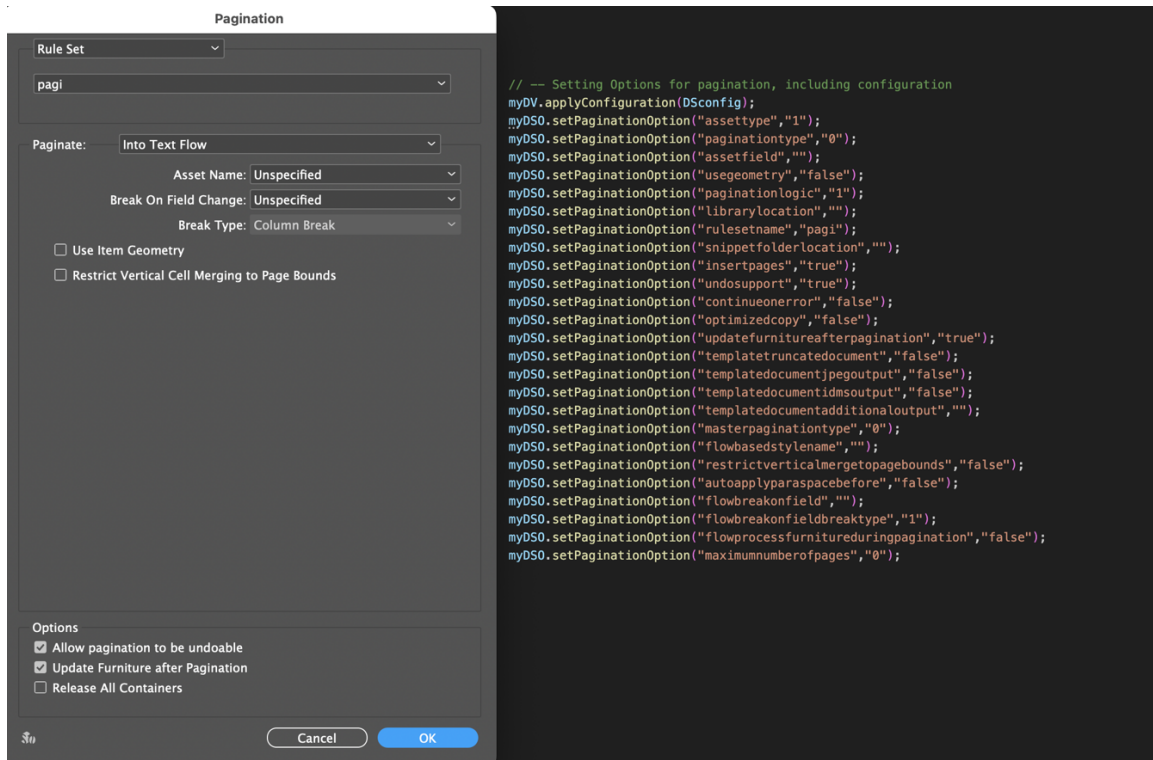
Kun kaikki muuttujat on muodostettu onnistuneesti, voidaan jatkaa itse taittoon. Skriptillä tarkistetaan materiaalien olemassaolo ennen niiden avaamista. keskeyttäen Skriptin ajaminen keskeytetään, jos jokin muuttujista on puuttuva tai väärin. Tähän käytetään JavaScriptin try-catch funktiota, palauttaen catch toimintoon muuttujan nimen, jos se on kirjoitettu esimerkiksi väärin tai kansiopolkua ei ole valittu (Kuva 11).

```
111 // -- Easycatalog panel setup
112 var myEC = app.easycatalogObject;
113 var myDS1 = app.easycatalogObject.datasources.item(listDS);
114 myDS1.synchronizeWithDataSource();
115 if (myDS1 == null)
116 {
117     throw ("Datasource not found:" + listDS);
118 }
119 var myDS = myEC.datasources.item(dataDS);
120
121 if(confirm('Synchronize with the Data Source?')) {
122     myDS.synchronizeWithDataSource();
123 }
```

Kuva 11. Try-catch toiminto sekä datan synkronointi.

Kun paneeliobjektit on avattu onnistuneesti, väliin lisätään myös varmistus siitä, että halutaanko paneelin dataa päivittää (Kuva 11). Silloin voidaan konfiguraatioon asettaa näkyviin vain tuotteet, joissa on uusia päivityksiä, mutta voidaan myös halutessa pitää vanhempi versio päivityksestä, jos on esimerkiksi paneeli jo aiemmin päivitetty vahingossa ennen automaatiota. Joskus paneelia avattaessa, paneeli päivittyy automaattisesti, jos se ei ole ollut jo valmiiksi auki. Jos konfiguraatioon on lisätty rajaus vain päivitetyille tuotteille, on tärkeää saada mahdollisuus kieltäytyä päivityksestä.

EasyCatalogin omat taittoasetukset voidaan saada kahdella eri tavalla. Jos paneeliin tallennettuun konfiguraatioon on tallennettu myös taittoasetukset, voidaan vain käyttää myDV.applyConfiguration() funktiota käyttäen aiemmin nimettyä muuttujaa "DSconfig" sen sisällä. Jos kuitenkin halutaan jostain syystä käyttää eri asetuksia kuin mitä konfiguraatioon on tallennettu, on sekin mahdollista, viitata jokaiseen taittoasetukseen erikseen. EasyCatalog on tehnyt tästä helppoa. Taittoon tarvittavat asetukset voi nappia painamalla kopioida koodiin. Kuvassa 12 näkyy samat taittoasetukset EasyCatalogin käyttöliittymässä ja kopioituna koodiin.



Kuva 12. EasyCatalog taittoasetukset verrattuna skriptin taittoasetuksiin.

Kyseisessä tehtävänannossa ei ole tarvetta muodostaa taittoasetuksia erikseen koodin riveillä, sillä skriptin käyttämä konfiguraatio on tehty vain ja ainoastaan koodin käytettäväksi, mutta konfiguraatio on silti asetettava. Taittoasetusten lisäksi konfiguraatiosta löytyy taitossa käytettävä ryhmittelyasetus sekä tallennettu tuotejärjestys, jos tuotteiden järjestyksellä on väliä.

Taitto tapahtuu niin että, skripti tekee valinnan taittopaneelistä yksi rivi kerrallaan, josta se taittaa rivin tiedot avattuun pohjaan. Aiemmin käyttäjän valitsevat muuttujat kertovat, millä kaikilla muodoilla tiedosto tallennetaan ennen kuin se suljetaan, jotta varmistetaan puhdas pohja seuraavalle riville. Puhtaan pohjan avaaminen on kannattavaa, vaikka saman pohjan voi päivittää uusilla tiedoilla, siltä varalta, että jos tuotteessa on ollut dataa tai puuttunut dataa, joka on ylitsekirjoittanut joitakin asetuksia, jotka on tallennettu alkuperäiseen pohjaan. Esimerkiksi jos pohjassa on laitettu poistumaan rivejä, jos dataa ei löydy, ei pohjan päivitys tuo puuttuvaa riviä takaisin, vaikka uudella tuotteella löytyisi dataa sille riville. Tästä muodostuu ensimmäinen tarvittu for-silmukka, joka toimii seuraavasti:

- valitsee paneelin tuoterivin
- avaa valitun pohjan, mihin on tallennettu EasyCatalogin esiasetukset
- rikastuttaa pohjan tuoterivin informaatiolla
- tallentaa pohjan tuotekoodilla ja kieliversiolla ennen sen sulkemista.

For-silmukkaa varten lasketaan, kuinka monta riviä taittopaneelissa on. Silmukka jatkaa toimintaansa niin kauan, kunnes kaikki taittopaneelin rivit ovat käyty läpi. Yhden rivin läpikäymiseen tarvitsee noin yhden sekunnin riippuen sen tietokoneen tehokkuudesta, jossa skripti ajetaan. Datassa, josta löytyy 39 uniikkia tuotetta, muodostaa paneelin 39:llä tuoterivillä, jolloin yhden version taitossa kestää noin puoli minuuttia, jos mukaan ei lasketa aikaa, kauanko paneelin päivityksessä kestää aikaa.

Paneelin konfiguraatiot

Kun skripti tekee valinnan paneelista, se valitsee vain näkyvät rivit ja taittaa näkyvästä valinnasta. Koska tuotteita on niputettu yhteen ja tuotenipusta halutaan tehdä yksi tiedosto, on myös mahdollistettava niputettu valinta paneelista. Paneelin ryhmittely täytyy luoda niin että päällimmäinen ryhmä on tämä tuotenippu ja tämä ryhmittely tallennetaan silloin konfiguraationa. Manuaalisessa taitossa on kuitenkin hidasteeksi, jos ryhmittelyä ei ole tehty kategorioiden mukaan, joten kategoriatyylinen ryhmittely on siksi aina manuaalisen taiton perusta. Tämän konfiguraation palautus tapahtuu kuitenkin vasta kun koko taitto on tehty.

Konfiguraatioon on myös tallennettu taittoasetukset (Kuva 12). Tuotejärjestykseen ei tarvitse ottaa konfiguraatioon kantaa tässä tapauksessa, sillä jokainen tuoterivi taitetaan omaan tiedostoonsa joka tapauksessa. Järjestykseen otetaan kantaa paneelissa vain silloin kun samaan tiedostoon halutaan taittaa useampi tuote. Taitossa käytettävä ryhmittely on sama kuin manuaalisessa käytössä näkyvä kuva 2:en ryhmittely, ilman kategoriaa ryhmänä. Tuotteet on ryhmitelty nimen perusteella.

Kun for-silmukoista on poistuttu, palautetaan taittopaneeliin sen alkuperäinen konfiguraatio, jota käytetään manuaaliseen taittoon. Tässä tapauksessa näillä eri konfiguraatioilla ei ole eroavaisuuksia itse paneelin taittoasetuksissa, sillä manuaalisesti toteutettava taitto on samanlainen kuin automaatiossa toteutettava julkaisu, mutta manuaalisessa julkaisussa on kategoriat mukana, helpottaen manuaalista taittoa tuotteita valittaessa.

3.2.3 Version vaihto skriptillä

Taittoversion, eli kielen, vaihto manuaalisessa taitossa tapahtuu niin että käyttäjä vaihtaa EasyCatalog paneelissa joko konfiguraatiota tai sitten vaihtaa paneelissa olevan tietyn kentän arvon haluttuun uuteen arvoon. Silloin jos versioita on monta erilaista tai versioiden määrä on elävä, konfiguraatiota jokaiselle versiolle ei kannata tehdä, sillä sen ylläpitäminen täytyisi tapahtua aina kehittäjän kautta. Jos datan versio on helposti luettavissa datasta itsestään, voidaan versiointi tehdä yhdessä kentässä, jota pidetään samassa arvossa kuin missä tahansa ohjelmointikielessä muuttujia.

Käännöksiä voi tulla lisää tai poistua kuvitteellisen asiakkaan resurssien mukaan ja se on myös helppo lukea datasta itsestään, joten se kannattaa pitää taittajan hallinnoimana tietona, että mitä versioita halutaan taittaa. Skripti kuitenkin vaatii jostain ennalta listatun tiedon erilaisista versioista joka tapauksessa. Ylläpidon helpottamiseksi, luodaan vain yksinkertainen excel-tiedosto, jossa on yksi sarake kielen avaimelle ja toinen sarake, missä voi olla mitä tahansa sisältöä. Excel-tiedostosta luodaan EasyCatalog paneeli, joka vaatii vähintään kaksi Excel-tiedoston saraketta toimiakseen. Excel-paneeli on skriptissä nimetty listattuna muuttujien kohdalla, joka on helppo muuttaa, jos käyttötarkoitus muuttuu.

Koska Excel-tiedostoon voi nyt viitata samalla tavalla kuin taittopaneeliin, voidaan luoda toinen for-silmukka taitto silmukan ympärille, joka toistuu myös niin monta kertaa kuin excel-paneelissa on rivejä, eli kieliä. Ensimmäisellä kerralla silmukan sisällä, se tallentaa ensimmäisen kielen talteen, jonka jälkeen se toteuttaa seuraavat askeleet:

- valitse excel-paneelin rivi ja poimi kieliavain sarakkeesta
- sijoita kieliavain taittopaneelin "lang" -kenttään
- päivitä taittopaneelin kentät.

Taittosilmukan jälkeen, kieli vaihtuu ennen seuraavaa taittosilmukkaa ja viimeisen kierroksen jälkeen tallennettu ensimmäinen kieli palautetaan takaisin paneelin oletusasetuksena. Jos taittopaneeliin rakennetut funktiot ovat monimutkaisia ja/tai monitasoisia, voi paneelin päivitys kestää pidemmänkin aikaa, mutta esimerkkitiedotalla tämä päivitys kestää alle sekunnin. On kuitenkin huomioitava, että esimerkkitiedot on rakennettu vain simuloimaan oikeaa tuotetdataa ja sen suppea koko ja lähdetyyppi nopeuttavat päivitystä. Jos kyseessä olisi mikä tahansa muu datalähde kuin paikallinen XML-tiedosto tai excel-tiedosto, saattaa lataamisessa kestää huomattavasti pidempi aika, joka saattaa olla pidempi kuin itse taittoon käytetty aika.

Kun molemmat for-silmukat ovat valmiita, eikä löydy enempää rivejä excel-paneelissa eikä taittopaneelissa, skripti palauttaa aikaisemmin skriptiin tallennetut oletusasetukset manuaalista taittoa varten. Nämä sisältävät oletuskielen sekä konfiguraation, jossa on manuaalisessa taitossa käytetty ryhmittely.

3.3 Ongelmakohdat

Koska automaatio on tehty ".indd" -tiedostopohjalla, kestää taittoprosessissa tavallista kauemmin. ".indt" -tiedostolla kestää vähemmän aikaa per yksi valmis tiedosto. Vaikka eroavaisuus per tiedosto onkin alle sekunnin, on se silti huomattava ero isommalla datamäärällä. Kaikenlaiset automaatiot hyödyntävät aina eniten, mitä isompi datamäärä on, joka on totta

tässäkin tapauksessa, mutta mitä suurempi datamäärä, sitä huomattavampi ero on siinä, milloin koko taitto valmistuu.

Vaikka itse taittaja voi tehdä jotain muuta silloin kun skripti on käynnissä, tietokone, missä skriptiä ajetaan, on varattu niin kauan kunnes taitto on valmis. Jos samalla tietokoneella yrittää tehdä jotain muuta samalla, voi se hidastaa taittoa entisestään ja pahimmassa tapauksessa myös pysäyttää sen kokonaan. Toisin kuin Adobe Photoshopissa, InDesignissa ei voi rajata ohjelman käyttämää muistin määrää, joten raskaiden prosessien aikana, muu koneen käyttö voi olla haastavaa, riippuen tietokoneen toimintakyvystä.

Koska tietokoneen jättää mielellään valvomatta silloin kun skriptiä ajetaan, jää myös sen valmistuminen helposti huomaamatta. Jos kyseessä on aikakriittinen taitto, sen valmistumisaikaa on vaikea arvioida ja siksi myös taitto skriptin suorittaminen on hankalaa aikatauluttaa, jolloin yleensä pitää tyytyä siihen, mitä automaatio tuottaa ilman, että aikaa jäisi käsin säätelylle jälkeensä. Tälle ei kuitenkaan tällaisessa taitossa olekaan tarvetta, ellei datassa itsessään ole virheitä, jolloin on suositeltavampaa korjata virhe dataan ja suorittaa automaatio uudestaan ongelmakohtissa kuin tehdä korjaus lopulliseen tiedostoon suoraan.

Silloin kun tiedetään että halutaan ajaa skripti tietyin väliajoin samasta datasta, joutuu taittaja tai muu käyttäjä silti käynnistämään skriptin manuaalisesti avaamalla ensin InDesignin ja sitten InDesignin käyttöliittymästä käynnistämään skriptin, sillä InDesignin käyttöoikeudet eivät riitä siihen, että InDesignin toimintoja toteutettaisiin automaatiolla ohjelman ulkopuolelta. Koska InDesigniin ei ole rakennettu minkäänlaista ajastusvaihtoehtoa, on sellaisen toteutus mahdollista tavallisella InDesignin työpöytä versiolla.

Tapauskohtaiset tarpeet

Jokaisella tapauksella on omat vaatimuksensa siitä, että miten tiedostot nimetään. Usein lopullisen tiedoston nimeäminen oikein on ainoa tapa, jolla voidaan tunnistaa mikä tiedosto kuuluu mihin tapaukseen. Varsinkin tällaisessa tapauksessa, jossa luodaan tiedosto per datarivi, on äärimmäisen tärkeää, että tiedostojen joukosta pystyy tunnistamaan mitä siitä löytää jo ennen sen avaamista.

Joskus tiedostoja halutaan myös viedä myös automaattisesti eteenpäin oikeisiin paikkoihin, esimerkiksi latauslinkin taakse nettisivulle tai vastaavasti valikoidaan jo, taitetuista vain tietyt painoon tai tulostukseen fyysiseen jakeluun.

Tässäkin tapauksessa on otettu huomioon, että tiedostot tarvitsevat uniikin nimen, että ne voidaan ylipäätään tallentaa omina tiedostoinaan eikä vahingossa toistensa päälle. Yksinkertaisuuden vuoksi, helpointa on käyttää nimeämisessä datarivin tunnistetta muuttujana ja

kieltä toisena muuttujana. Mahdollisesti nimeämisessä voidaan käyttää muitakin tunnisteita tai merkkejä.

Voimme kuitenkin huomioida sen, että missä hypoteettinen asiakas tulee säilyttämään tiedostoja ja ottaa huomioon, jos sijainnilla on rajoituksia. Tähän esimerkkiin on otettu mukaan Microsoft OneDriven palvelu, jossa löytyy rajoituksia sekä yksittäisille erikoismerkeille mutta myös merkkijonoille. On siis pidettävä huolta, ettei tiedoston nimessä käytetä näitä merkkejä tai merkkijonoja. Jos kuitenkin halutaan erikoismerkkejä, on myös mahdollista, että niihin pitää viitata Unicodella. Tässä tapauksessa merkki "α" on käännetty unicodeksi "\u00A4", jolloin lopulliseksi nimeämiseksi on asetettu "[data avain] + "\u00A4" + [kieli] + [haluttu tiedostomuoto]".

3.4 Jatkokehitys

Vaikka skripti suoriutuukin hienosti vaatien käyttäjältä vain muutaman napin painalluksella, se on silti muutama napin painallus, jotka täytyy painaa ennen kuin voi käyttäjä irrottaa kätensä taitosta. Lisäksi tietokone, jossa taitto tehdään, on taiton ajan käyttämätön mihinkään muuhun, sillä skripti pitää InDesignin aktiivisena ja minkään muun asian tekeminen samalla tietokoneella on hidasta. Riippuen datan suuruudesta, tämä saattaa olla pitkäkin aika ja vaikka ihmisresurssi on tästä vapaana, tietokone istumassa varattuna vain tähän tehtävään ei ole ideaali.

Vaikka käyttäjän syöttämät tiedot valikkoihin on helposti korvattavissa, jos halutaan säästyä vain yhdellä napin painalluksella, jolla käynnistää InDesignin, voi skriptin asettaa skriptiksi, joka käynnistyy aina InDesignin käynnistyessä, mutta jos InDesignia on tarkoitus käyttää samalla tietokoneella johonkin muuhun asiaan, ei sekään ole vaihtoehto, että skripti käynnistetään aina InDesignin käynnistyessä. InDesignia ei myöskään voi käynnistää virtuaalikoneella sen käyttöoikeuksien rajauksen takia (Adobe 2024a, 6.6).

Yksi vaihtoehto, millä voidaan kokonaan poistaa käyttäjän syötön taiton automatisoinnin aloittamiseksi, löytyy kuitenkin Adobelta itseltään. Adoben InDesign Server on tuote, joka toimii täysin kuten tavallinen InDesign, mutta sillä ei ole käyttöliittymää. Se vaatisi skriptin muokkausta niin että se ei ohjaa pelkästään EasyCatalogia, mutta myös InDesignia itseään. InDesign Server on luotu juuri siihen tarkoitukseen, että voi käyttää InDesignin ominaisuuksia, mutta niin että koko prosessi on automatisoitu. Serverin käynnistyksen ajastus ja skriptin käynnistyksen ajastuksen voi suorittaa serverillä.

Käytettyjä resursseja voi vähentää myös nopeuttamalla itse prosessia, mikä kestää taitossa. Olemassa olevilla oletuksilla tehty pohja hidastaa itse taittoa sillä, että InDesign joutuu avaamaan, taittamaan, tallentamaan ja sitten sulkemaan jokaisen tiedoston minkä se

luo. EasyCatalogissa on myös ominaisuus, joka tekee saman prosessin täydentämällä ennalta tehtyyn pohjaan tietoja. Jos tätä toteutusta pystyisi kehittämään niin, että sivumäärän voisi ennakoita, voitaisiin tällä käyttää EasyCatalogin omaa ".indt" päätteistä template -tiedosto taittoa, jossa yhden tiedoston taittaminen on huomattavasti nopeampaa.

Myös paneelissa on tehtävissä funktioiden osalta optimointia, joka nopeuttaisi sen toimintaa ja datan päivittämistä. Joka kerta kielen vaihtuessa, paneelin funktiot päivittyvät, kun ne hakevat uuden kielen. Näin sen kuuluukin tapahtua, mutta on myös otettava huomioon vaihtoehto, jossa käytettäisiin itse kirjoitettuja LUA-ohjelmointikielen funktioita, jotka on optimoitu jokaisen tapauksen dataan sekä haluttuun tulokseen. EasyCatalogin omat funktiot ovat rakennettu yleispäteviksi moneen käyttötarkoitukseen, mutta niiden alla on sama LUA-ohjelmointikieli, jonne on lisätty ehtoja ja tarkistuksia mitä ei välttämättä tarvitse huomioida ollenkaan.

Tietysti aivan ensimmäinen kehitykseen tehtävä muutos olisi poistaa ylimääräiset funktiot paneelista, jotka vain nopeuttivat sen luontia. Moni kenttä paneelissa luotiin vain kopiaimalla ja liittämällä edellisestä samankaltaisesta kentästä, joissa viitataan kentän nimeen ainoana muuttujana. Joskus nimi myös pilkotaan, että siitä saadaan haluttava osio. Jos nämä kaksi funktiota korvaa vain staattisella muuttujalla, paneelin asetuksiin kestää vähän enemmän aikaa, mutta jokainen kenttä ei joutuisi menemään silloin kahden ylimääräisen funktion läpi vain hakeakseen tiedon siitä, että mitä tietoa halutaan hakea.

4 Yhteenveto

Opinnäytteessä tutkitaan EasyCatalogilla luodun julkaisuautomaation viemistä pidemmälle, sillä että aiemmin manuaalisesti tehtävä valinta tehdään automaattisesti. Versiovalinta sekä tuoterajauksen valinta on aiemmin ollut manuaalisesti käyttäjän hoidettava vastuu, joka opinnäytteessä muutettiin automaation hoidettavaksi tehtäväksi.

EasyCatalogin avulla tehty automatisointi vähentää tarvittuja resursseja ja ihmisten tekemiä virheitä, joita sattuu melkein pakostikin, kun tehdään perinteistä taittoa. Usein taiton jälkeen halutaan vielä käsin muokata lopputulosta, mutta julkaisussa, jossa ulkonäkö ei ole prioriteetti vaan datan oikeellisuus, ei ihmisen muokkausta tarvita jälkeenkään. Data voidaan hakea asiakkaan omasta järjestelmästä, joka on jo yhdistetty muihinkin paikkoihin ja sen oikeellisuus on jo varmistettu.

ExtendScriptin avulla mahdollistettiin vielä pidemmälle viety automatisointi, joka ei taita pelkästään yhtä julkaisutyyppiä kaikkien tuotteiden kohdalla vaan myös eri versioinnit. Opinnäytteessä toteutettiin vain kieliversiointi, mutta skriptiin voi helposti vaihtaa jonkin muunkin muuttujan, joka vaihtaa erilaista julkaisuversiota.

Toteutettu automatisointi on yhtä napin painallusta vaille täydellinen automatisointimahdollisuus, jossa halutessaan käyttäjä voi painaa InDesignin päälle, jolloin skripti ajautuu itsensä taittaen kaikki ohjelmoidut julkaisuversiot läpi. Käyttäjä voi myös valita itse käytettävät InDesign -pohjat ja sijainnin minne valmiit tiedostot halutaan viedä.

Minkä tahansa tasoinen automatisointi irrottaa olemassa olevia resursseja julkaisuiden luomisesta ja minimoi virheiden tapahtumisen riskin datan oikeellisuudessa. EasyCatalog itseään on jo hyödyllinen työkalu, mutta vaivan näkeminen EasyCatalogin automatisointiin on pieni suhteessa siihen, miten paljon aikaa säästää taittajan pöydältä.

Riippuen siitä miten paljon julkaisuja tehdään, sitä enemmän kannattaa harkita automatisointia, huolimatta siitä, vaatiiko julkaisu vielä käsin säätämistä jälkeenkään. Automatisoinnin voi viedä juuri niin pitkälle kuin haluaa, mutta oli se minkä tasoista tahansa, automatisointiin sijoittaminen luo säästöjä myöhemmin.

Lähteet

Adobe. 2018. Adobe InDesign CC 2018 Scripting ReadMe. PDF-dokumentti. Viitattu: 5.5.2024. Saatavissa: <https://www.cleverprinting.de/download/InDesignScripting.pdf>.

Adobe. 2023a. Importing graphics formats into InDesign. WWW-dokumentti. Viitattu 18.5.2024. Saatavissa: <https://helpx.adobe.com/indesign/using/importing-files-applications.html>

Adobe. 2023b. InDesign / Common questions. WWW-dokumentti. Viitattu 16.4.2024. Saatavissa: <https://helpx.adobe.com/indesign/faq.html>.

Adobe. 2023c. Text variables. WWW-dokumentti. Viitattu 16.4.2024. Saatavissa: https://helpx.adobe.com/indesign/using/text-variables.html#text_variables

Adobe. 2024a. Adobe General Terms of Use. Viitattu 16.4.2024. Saatavissa: <https://www.adobe.com/legal/terms.html>.

Adobe. 2024b. Syntax differences between expression engines. WWW-dokumentti. Viitattu 13.4.2024. Saatavissa: <https://helpx.adobe.com/after-effects/using/legacy-and-extend-script-engine.html>.

Tevela. 2024. Koulu, Kevät 2024 kausiesite, Varhaiskasvatus. EBUB-kirja. Viitattu 16.4.2024 Saatavilla: <https://kuvastot.tevela.fi/tevela-tarjousesite-varhaiskasvatus/>.

65bit Software Ltd. 2007. XML Data Provider User Guide. PDF-dokumentti. Viitattu 13.4.2024. Saatavissa: https://www.65bit.com/downloads_new/files/easycatalog/documentation/English/XML%20Data%20Provider%20User%20Guide.pdf.

65bit Software Ltd. 2010. Scripting Reference. PDF-dokumentti. Viitattu 13.4.2024. Saatavissa: <http://www.naolis.com/easycatalog/EasyCatalog-Scripting.pdf>.

65bit Software Ltd. 2024a. Commands. HTML-dokumentti. Viitattu 16.4.2024. Ei saatavilla.

65bit Software Ltd. 2024b. EasyCatalog Flyer. PDF-dokumentti. Viitattu 13.4.2024. Saatavissa: https://www.65bit.com/downloads_new/files/easycatalog/documentation/English/Leaflet.pdf.

65bit Software Ltd. 2024c. EasyCatalog User Guide. PDF-dokumentti. Viitattu 13.4.2024. Saatavissa:

https://www.65bit.com/downloads_new/files/easycatalog/documentation/English/EasyCatalog%20Manual.pdf.

65bit Software Ltd. 2024d. EasyCatalog White Paper. PDF-dokumentti. Viitattu 13.4.2024. Saatavissa:

https://www.65bit.com/downloads_new/files/easycatalog/documentation/English/EasyCatalog%20-%205%20Steps.pdf.

65bit Software Ltd. 2024f. LUADOC.HTML. HTML-dokumentti. Viitattu 16.4.2024. Ei saatavilla.

65bit Software Ltd. 2024e. Scripting Module User Guide. PDF-dokumentti. Viitattu 13.4.2024. Saatavissa:

https://www.65bit.com/downloads_new/Scripting%20Documentation/modules/EasyCatalog%20Scripting.html.