



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Daniel Sulasalmi

# IQ Vision -kiinteistövalvomon migraatioprosessin automatisointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

11.01.2021

Tekijä Otsikko Sivumäärä Aika	Daniel Sulasalmi IQ Vision -kiinteistövalvomon migraatioprosessin automaatio sointi 33 sivua 11.01.2021
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Ammatillinen pääaine	Automaatiotekniikka
Ohjaajat	Yliopettaja Erkki Räsänen Huoltopäällikkö Jouni Marjomäki
<p>Markkinoilta poistuvan vanhan 963-kiinteistövalvomon tuki on loppumassa ja uuteen IQ Vision -kiinteistövalvomoon päivittäminen on monessa kiinteistössä edessä. Insinööriyön tavoitteena oli automatisoida ja nopeuttaa migraatioprosessia Trendin vanhasta 963-valvomosta uuteen IQ Vision -valvomoon. Työn aloitusvaiheessa migraatioprosessi vaati paljon manuaalista työtä, mikä lisäsi valvomon päivittämisen hintaa asiakkaalle ja mahdollisti enemmän virheitä kääntöprosessissa.</p> <p>Työn alkuvaiheessa perehdyttiin molempien valvomosovellusten toimintoihin ja etsittiin eroavaisuuksia. Tämän jälkeen tutkittiin migraatiossa tapahtuvia virheitä ja sitä, miten virheet korjataan grafiikalla. Kun virheet oli korjattu grafiikalla manuaalisesti, tutkittiin, miltä korjaukset näyttivät tekstitiedoston näkymässä. Näkymän avulla selvitettiin, mitä lisäominaisuuksia jokainen objekti vaati. Tämän avulla saatiin kehitettyä ”etsi ja korvaa” -toiminnot, joiden avulla koko grafiikkasivulle saatiin muutokset tehtyä kerralla.</p> <p>Muutoksien tekeminen sivu kerrallaan ei kuitenkaan nopeuttanut prosessia tarpeeksi ja mahdollisti virheille, koska jokainen sivu täytyi käydä läpi yksi kerrallaan. Tämän takia keksittiin keino tehdä muutokset koko valvomoon kerralla Notepad++ -sovelluksen avulla. Notepad++ mahdollisti muutokset koko valvomoon ja lisäksi toi mahdollisuuden käyttää makroja vaikeampiin muutostöihin. Sovelluksen ansiosta muutostyöhön kuluva aika puolittui, ja monet ennen manuaalisesti tehtävät muutokset saatiin näin automatisoitua.</p>	
Avainsanat	IQ Vision, kiinteistövalvomo, Niagara, Trend

Author Title Number of Pages Date	Daniel Sulasalmi Automation of the Migration Process for the IQ Vision Building Control System Interface 33 pages 11 January 2021
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation Technology
Instructors	Jouni Marjomäki, Quartermaster Erkki Räsänen, Principal Lecturer
<p>Trend's old building control system interface 963 will not be supported in future and many customers will have to upgrade to a new platform called IQ Vision. To make transition easier, Trend has created a migration tool for transferring the old 963 graphics to the new platform. Unfortunately, the migration tool is not perfect and many graphics do not work the same way in IQ Vision. The goal of this thesis project was to make migration to the new IQ Vision easier and cheaper. At the start of this project, the migration process included a lot of manual work, which enabled appearance of bugs. Because of manual work, the migration is also expensive for the customer and they may not want to upgrade their interface.</p> <p>The project started by getting to know both interfaces better and finding differences between them. Getting better knowledge about these systems helped to understand the process of migration. After better understanding, it was time to see what bugs the migration caused and how to fix them. The graphics could be viewed as a text file in IQ Vision, so it was checked how changed objects looked like in a text file. In a text file, it is possible to do a search and replace function, making editing the graphics much faster.</p> <p>Search and replace function is good for making changes one graphic page at a time but to make process even faster, a new tool for this had to be found. The new tool was Note-pad++. It made possible to make these changes to the entire platform and allowed a use of macros for the difficult tasks. As result, the migration process has now a lot less manual work, is at least twice as fast, and has less bugs.</p>	
Keywords	IQ Vision, building control system, Niagara, Trend

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Insinööriyöhön liittyvät aiheet	2
2.1	Insinööriyön rajaus ja taustat	2
2.2	Rakennusautomaatio	2
2.3	RAU valvomo	3
3	Trend valvomot	3
3.1	Trend	3
3.2	963-valvomo	4
3.3	IQ Vision	5
3.4	Valvomoiden eroavaisuudet	5
3.5	96x Schematic Export	6
4	Migraatio	7
4.1	Migraation vaiheet	7
4.2	Aloitustilanne	8
4.3	IQ Vision migraatiotyökalu	9
4.4	IQ Vision grafiikkakuvat migraation jälkeen	11
4.5	IQ Visionissa tarvittavat muutokset	12
4.6	Grafiikoiden toiminta ja muutokset "Px Editor" -tilassa	14
4.7	Grafiikoiden toiminta ja muutokset tekstieditorista tarkasteltuna	18
4.8	Pisteiden muokkaukset	21
4.9	Trendipisteet	21
4.10	"Custom" -pisteen luominen	22
5	Migraation automatisointi	24
5.1	Grafiikoiden muokkaus "etsi ja korvaa" -tyylillä	24
5.2	Notepad++	26
5.3	Pistetietokannan muutokset Notepad ++ -ohjelman avulla	27

5.4	Pistetietokannan muutokset IQ Visionin omien työkalujen avulla	29
6	Automatisoinnin hyöty migraatiossa	29
6.1	Säästetty aika	29
6.2	Virhemarginaalin pieneneminen	30
6.3	Pythonin tarjoamat lisämahdollisuudet	30
7	Yhteenveto	31
	Lähteet	33

## Termit ja lyhenteet

Migraatio Vanhan järjestelmän siirtäminen uuteen.

Python Monipuolinen ohjelmointikieli.

RAU Rakennusautomaatio.

IP Internetin protokollaosoite.

SET System Engineering Tool, eli järjestelmäsuunnittelu-työkalu.

HTML Hypertext Markup Language, eli hypertekstin merkintäkieli.

SQL Structured Query Language, kyselykieli.

## 1 Johdanto

Insinööriyön tavoitteena on helpottaa ja automatisoida vanhan kiinteistövalvomojärjestelmän migraatiota uuteen. Markkinoilla jo monta vuotta olleen Trendin 963-valvomojärjestelmän tuki on loppumassa, ja uudemman sukupolven valvomo on tullut markkinoille. Uusi Trendin IQ Vision -valvomo perustuu Niagara-pohjaan ja eroaa monin tavoin vanhasta 963-valvomosta. Trend on kehittänyt migraatiotyökalun, jonka avulla valvomon siirtäminen uudelle alustalle on helpompaa, mutta migraatiotyökalusta huolimatta joudutaan tekemään paljon pieniä muutoksia käsin, jotka lisäävät valvomon päivittämisen hintaa sekä pidentävät migraation kestoa. Tämän takia kynnys valvomon päivittäminen uudelle alustalle kasvaa, ja kilpailijoiden tarjoamat vaihtoehdot voivat tulla halvemmaksi.

Työ tehdään Assemblin Oy:lle. Assemblin toimii Suomessa, Ruotsissa ja Norjassa ja on Suomen suurimpia talotekniikkayrityksiä. Assemblin työllistää noin 5900 henkilöä Pohjoismaissa, ja sen liikevaihto on noin 970 miljoonaa euroa. Assemblin on tehnyt monta Suomessa olevaa Trendin 963 -valvomoa. Jos vanhalle valvomolle tapahtuu jotain, niin siihen ei saa tukea, ja uuteen valvomoon päivittäminen on tällöin ainoa vaihtoehto. Tullevaisuudessa on todennäköistä, että monia valvomoita joudutaan päivittämään. Tämän takia on tärkeää automatisoida päivitysprosessi mahdollisimman pitkälle. (1.)

Insinööriyö aloitetaan tutustumalla IQ Vision -valvomoon ja sen toimintoihin sekä eroavaisuuksiin vanhaan valvomoon verrattaessa. Kun valvomoiden eroavaisuudet ja toiminnot on käyty läpi, on tutustuttava migraatioprosessiin ja sen vajavaisuuksiin. Migraatio sisältää monia vaiheita uuteen valvomoon päivitettäessä, mutta eniten työtä teettää grafiikkasivujen manuaalisen korjaamisen määrä. Tavoitteena on saada migraatio onnistumaan miltei täydellisesti, mutta tämä vaatii melko paljon tekstitiedostojen tutkimista ja ohjelmointia. Insinööriyön aikana on tarkoitus käydä myös Python-ohjelmointikielen alkeiskurssi, jotta migraatiolle saadaan kirjoitettua toimivat skriptit.

## 2 Insinööriyöhön liittyvät aiheet

### 2.1 Insinööriyöhön rajaus ja taustat

Työssä keskitytään suurilta osin uuden IQ Vision -valvomon toimintoihin ja muutoksiin. Eroja verrataan vanhaan 963 -valvomoon. Erojen perusteella pyritään ymmärtämään migraatiotyökalun toimintoja ja keksiä tapoja migraation helpottamiseen. Kun tavat migraation helpottamiseen on löydetty, selvitetään, miten paljon aikaa optimoidulla tavalla tehty migraatio säästää. Työssä ei käsitellä rakennusautomaation toimintoja kuin vain pintapuolisesti, koska migraation kannalta toimintojen ymmärtäminen ei ole välttämätöntä.

Tällä hetkellä migraatiossa menee paljon aikaa. Tämän takia myös oikein tehty migraatio maksaa kiinteistöille melko paljon. Vanhalla 963-valvomolla toimivia kiinteistöjä on vielä paljon käytössä, eikä kallis migraatio houkuta päivittämiseen, vaikka tämä olisikin välttämätöntä tehdä tuen puutteen takia. Migraatio sisältää tiettyjä pakollisia manuaalisia töitä, mutta ne ovat samat valvomon koosta riippumatta. Esimerkiksi vanhaan valvomoon vaaditaan tiettyjä lisäyksiä, jotta se toimii selaimessa. Manuaalisen työn määrä ajallisesti ei näissä asioissa ole kuitenkaan ratkaisevassa roolissa, koska kuluva aika on vähäinen.

Tulevaisuudessa valvomoiden päivittämisen kysyntä tulee kasvamaan ja mahdollisimman nopea, sekä laadukas päivittäminen antaa mahdollisuuden monen asiakkaan tilauksen toteuttamiseen ja pienentää työjonon kasaantumista. Ideana on nopeuttaa ja pienentää virheiden määrää tällä hetkellä käsintehdyssä migraatiotyössä. Tämä myös pienentää kiinteistöjen kynnystä valvomon päivittämiseen, koska migraation laatu paranee ja hinta laskee.

### 2.2 Rakennusautomaatio

Rakennusautomaatio yhdistää kiinteistön eri osa-alueet samaan järjestelmään, jolloin niitä on helppo seurata ja ohjata. Tämän ansiosta esimerkiksi hälytykset, lämpötilat, valaistukset sekä erilaiset koneet saadaan helposti ohjelmoitua tekemään haluttuja asioita. Rakennusautomaation ideana on myös lisätä kiinteistöjen energiatehokkuutta ja kerätä



dataa esimerkiksi hälytyksistä ja lämpötiloista. Isoissa kiinteistöissä tämä helpottaa käyttöä ja sopivan lämpötilan ylläpitoa. Vikatilanteissa vika saadaan tietoon ja selville huomattavasti nopeammin, kun rakennusautomaatio on kunnossa ja ajan tasalla.

### 2.3 RAU-valvomo

Valvomolla tarkoitetaan usein tietokonetta, josta on graafinen näkymä kaikkiin kiinteistön laitteisiin ja isoissa kiinteistöissä myös tasokuviin, joista tarvittaessa pääsee näkemään huoneiden lämpötiloja ja lämpötilojen asetuksia. Isoissa kiinteistöissä valvomon osuus rakennusautomaatiossa nousee todella tärkeäksi, koska valvomon kautta pystytään esimerkiksi asettamaan lämpötiloja, muuttamaan ilmanvaihtokoneiden puhaltimien nopeuksia ja tarvittaessa pysäyttämään ne huollon ajaksi.

Trendin valvomoissa puhaltimien ja eri koneiden tiedot tulevat grafiikalle pistetietokannan avulla. Pistetiedoilla tarkoitetaan joko mittaus-, tilatieto- tai asetusarvopisteitä. Nämä ovat tietokannassa tunnistettavilla alkukirjaimilla, kuten "S" tarkoittaa mittausta, "K" asetusta, "I" tilatietoa ja "W" kytkintä. Valvomon ansiosta pystyy myös helposti paikantamaan hälytyksiä ja reagoimaan niihin nopeasti.

Valvomon avulla aikaohjelmien käyttö ja muuttaminen helpottuu. Aikaohjelmalla voidaan asettaa esimerkiksi ilmanvaihto tai valaistus toimimaan tiettyinä kellonaikoina ja päivinä. Valvomon kanssa näitä saadaan muutettua helposti tai ottamaan tarvittaessa pois käytöstä.

## 3 Trend-valvomot

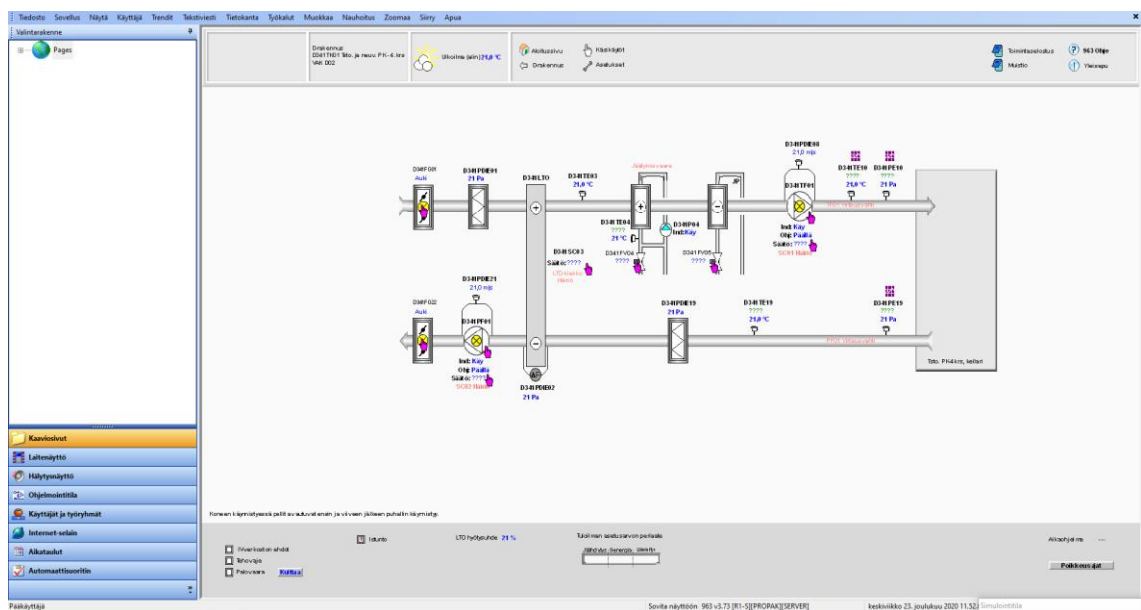
### 3.1 Trend

Trend on vuonna 1980 Yhdistyneessä kuningaskunnassa perustettu rakennusautomaatiojärjestelmien toimittaja. Trendin tuki- ja jakeluverkkoon kuuluu yli 50 maata. Yhdysvaltalainen Honeywell osti Trendin vuonna 2005. Trend kuitenkin valmistaa ja suunnittelee

tuotteensa vieläkin Britanniassa, mutta Honeywellin tuotteita on integroitu mukaan toimintaan. (2; 3.)

### 3.2 963-valvomo

Trendin 963-valvomo on ollut markkinoilla vuodesta 2003, ja se poistui markkinoilta loppuvuodesta 2019. Valvomo perustuu SQL-tietokantaan, josta se hakee dataa sekä tallentaa sitä. 963 on todella helppokäyttöinen, ja isojakin muokkauksia on melko vaivastonta tehdä helpon käyttöliittymän ansiosta. Valvomo-ohjelma on kokonaan Trendin kehittämä, ja tämän takia se toimii hyvin vanhojen Trendin alakeskuksien kanssa. Vaikka valvomoa on kehitelty pitkälle 2000-luvun alusta, sen käyttöliittymä on todella vanhanaikainen ja toimii melko huonosti uusimpien käyttöjärjestelmien kanssa. 963 käyttää esimerkiksi Java-liitäntäisiä, jotka ovat vanhanaikaisia ja niiden tuki on loppunut. Tämä lisää sovelluksen tietoturvariskejä sekä ennustettavuutta tulevaisuuden toiminnan kannalta.

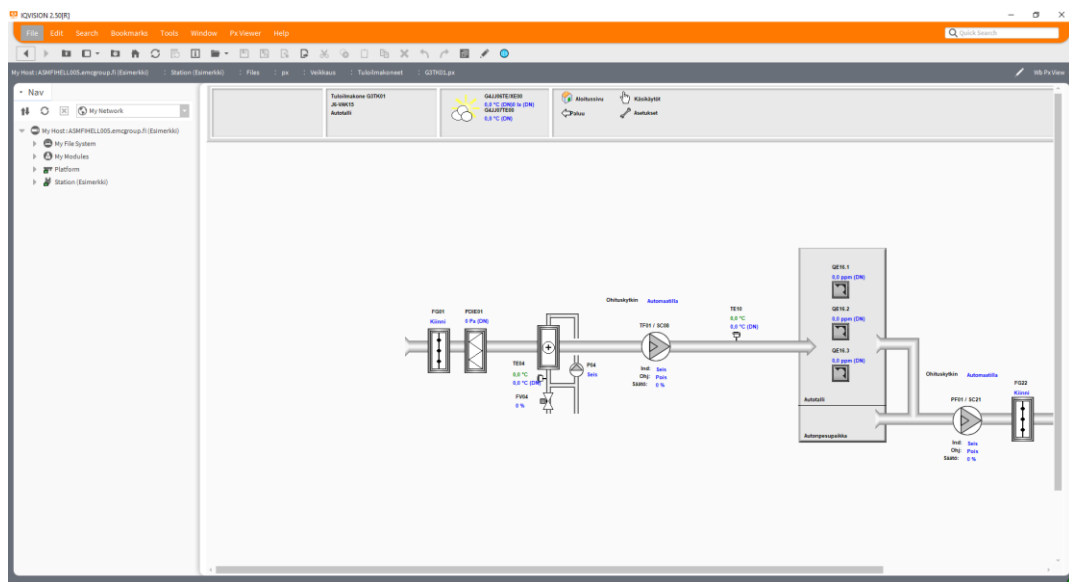


Kuva 1. 963-käyttöliittymä.

### 3.3 IQ Vision

IQ Vision julkaistiin vuonna 2018, ja se tuo Trendin järjestelmät tälle vuosikymmenelle. IQ Vision, tästä eteenpäin Vision, käyttää Niagara 4 -käyttöliittymää, joka hyödyntää HTML 5 web -tekniikkaa. Tämän avulla valvomoa voidaan käyttää selaimen kautta, joka lisää valvomon käyttömahdollisuuksia esimerkiksi kannettaville laitteille.

Vision-valvomon vektorigrafiikoiden avulla kaikki objektit skaalautuvat ruudulle sopivaan kokoon ja objekteihin on mahdollista zoomata. Vision toimii hyvin uusien käyttöjärjestelmien, kuten Windows 10, kanssa. Visionin grafiikoita voi muokata monessa eri näkymässä, joista käytetyimmät ovat "Px editor", missä kuvia muokataan suoraan graafisesti, ja "Text editor", missä grafiikoita voi muokata tekstitiedoston muodossa.



Kuva 2. IQ Vision -käyttöliittymä.

### 3.4 Valvomoiden eroavaisuudet

Vaikka Trend on molempien valvomoiden tarjoaja, on niillä silti monia isoja eroavaisuuksia. 963-valvomo on puhdas työpöytäsovellus ja Vision on palvelinsovellus. Vision ei

käytä SQL-tietokantoja ja sitä käytetään selaimella. 963-valvomon grafiikoissa on pystynyt käyttämään ”plot”-pisteitä, eli grafiikkapisteiden historiaa kerääviä pisteitä, jotka mahdollistavat lämpötilojen ja muiden anturi- sekä asetusarvojen liittämisen grafiikkaan sellaisenaan. Visionin valvomossa tämä ei ole mahdollista ja ”plot”-pisteitä voi käyttää pelkinä historiatietoina. Jos vanhassa valvomossa on käytetty ”plot”-pisteitä, joudutaan uuteen valvomoon tekemään ”custom”-pisteitä. Tämä lisää työmäärää sekä nostaa valvomon ”pistemassaa”.

Rahallisesti valvomoiden isoin ero on Visionin mukana tulleet uudet lisenssit. 963-valvomoon sai lisätä rajattomasti pisteitä samalla hinnalla, mutta Vision-valvomossa eri pistemäärille on omat lisenssit. Tämä luo hankaluuksia valvomon hinnoitteluun, koska lopullisesta pistemäärästä voidaan olla varmoja vasta kaikkien grafiikoiden kääntämisen ja muutoksien jälkeen.

Valvomoiden isot erot johtuvat vuosien aikana tapahtuneesta kehityksestä tietotekniikan alalla, mutta myös valvomoiden kehittäjät eroavat toisistaan. Vanha 963-valvomo on kokonaan Trendin oman kehittelyn ja ideoinnin takana, mutta Vision on liitetty Niagara 4 -käyttöliittymään, mikä tuo hyviä ja huonoja puolia uudistuneeseen valvomoon. Hyvinä puolina voidaan pitää Niagaran laajaa käyttöä maailmalla sekä hyvä tuki- ja kehitysverkosto. Huonoja puolia ovat Trendin mahdollisuudet yhtenäistää vanhaa ja uutta valvomoa keskenään, joista migraation vajavaisuus on esimerkkinä.

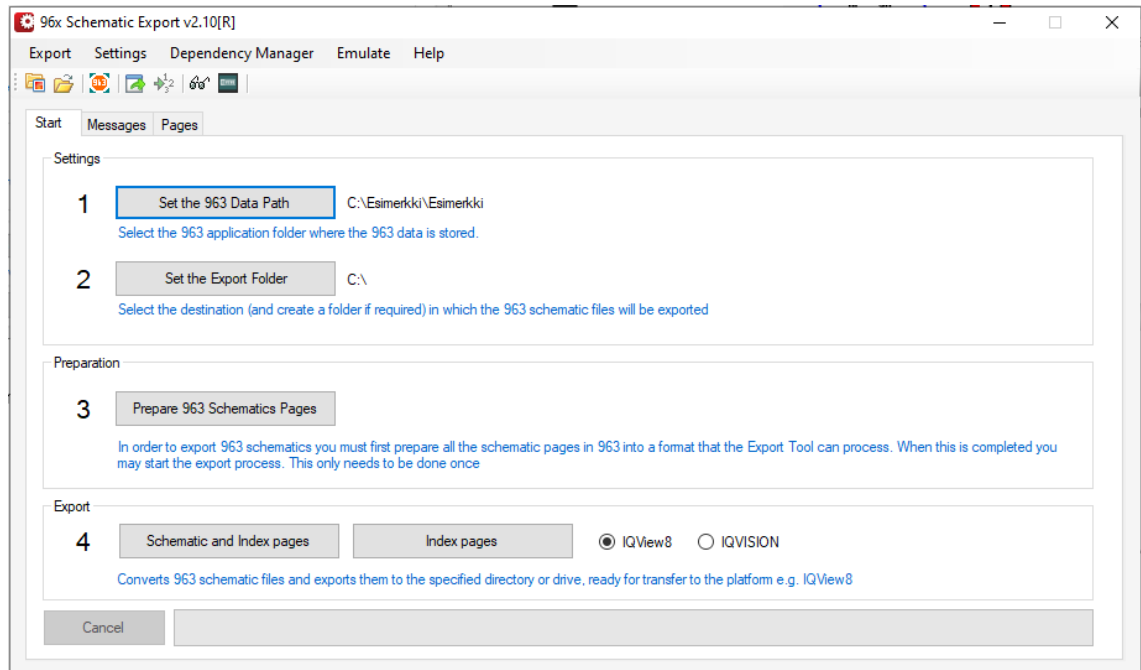
### 3.5 96x Schematic Export

Uuden Vision-valvomon päivittämisen kynnyksen pienentämiseksi, on Trend tehnyt työkalun, jonka avulla saadaan siirrettyä grafiikkasivut vanhasta 963-valvomosta. Pistekannat haetaan yleensä suoraan alakeskuksesta, eli kiinteistöautomaatiota ohjaavalta laitteelta. Kun pistetietokanta on Visionissa, voi grafiikat tuoda vanhasta valvomosta.

Työkalu toimii melko hyvin, ja se osaa käsitellä kaikkia pisteitä, ”plot”-pisteitä lukuun ottamatta. ”Plot”-pisteet eivät käänny uudelle grafiikalle ollenkaan ja joudutaan tekemään uudet pisteet näiden sijasta. Tämä aiheuttaa paljon työtä, riippuen siitä, kuinka paljon ”plot”-pisteitä on käytetty vanhalla grafiikalla.

96x Schematic Export -työkalu kääntää kaikki mahdolliset tiedot vanhasta valvomosta, jotka eivät välttämättä toimi uudessa Vision-valvomossa, esimerkiksi edellisessä kappalessa mainitut ”plot”-pisteet. Näitä ominaisuuksia joudutaan korjaamaan käsin. Myös grafiikkoja joudutaan muokkaamaan käsin. Jos vanhassa valvomossa on tiettyjä objekteja piirrettynä, ei sivu käänny ollenkaan, ja kyseiset objektit täytyy etsiä ja poistaa 963-valvomosta.

(4; 5.)



Kuva 3. 96x Schematic Export -käyttöliittymä.

## 4 Migraatio

### 4.1 Migraation vaiheet

963-valvomon päivitys uuteen Vision-valvomoon sisältää aina seuraavat asiat valvomon koosta ja hinnasta riippumatta.

- 963-valvomon toiminnan pienimuotoinen tarkistus, pistemäärän laskeminen ja grafiikoiden siistiminen ennen migraatiota
- uuden Vision-valvomon lisensointi oikealle pistemäärälle
- pisteiden haku uuteen Vision-valvomoon suoraan kohteen alakeskuksista tai Trendin SET-työkalun avulla
- 963-valvomon vanhojen grafiikoiden kääntäminen ja siirtäminen Vision-valvomoon 96x Schematic Export- työkalun avulla
- grafiikoiden tarkastelu uudessa valvomoympäristössä ja isojen puutteiden korjaukset, kuten siirtymättä jääneiden grafiikkasivujen korjaus ja uudelleen siirtäminen uuteen valvomoon
- kaikkien pienten virheiden korjaukset jokaisella grafiikkasivulla, sekä pisteiden yksikköjen ja desimaalien muuttaminen. Esimerkiksi painikkeiden luonti ja linkkien korjaus
- hälytysreitityksien tekeminen ja hälytysikkunan luonti sekä sähköpostihälytysten reitittäminen
- selainkäytön mahdollistaminen tekemällä käyttäjätilejä rakentamalla navigointipalkki selainikkunan laitaan ja muuttamalla valvomon selainosoite luotetuksi osoitteeksi.

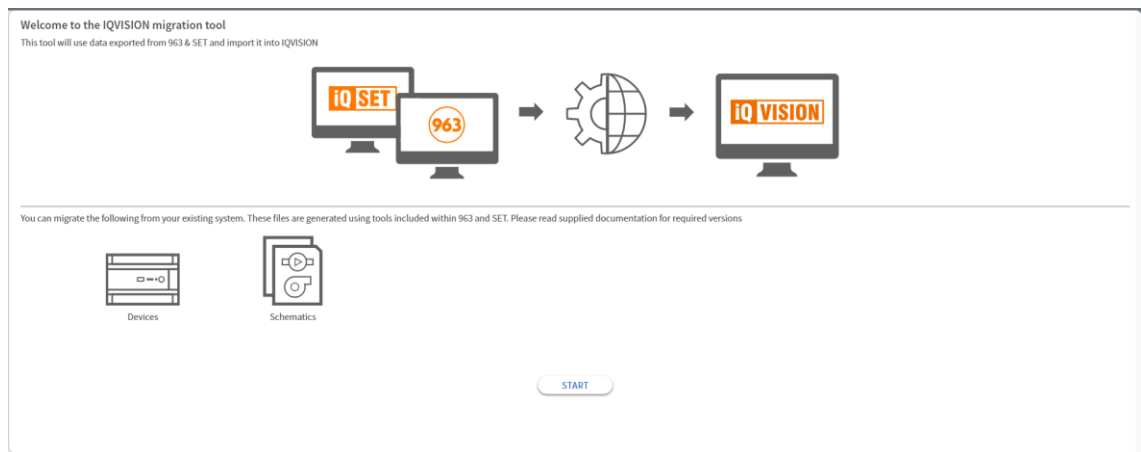
Edellä mainitut vaiheet ovat osana jokaista migraatiota. Isoimman hinta- ja aikaeron valvomon koon mukaan määrittää lisenssit ja grafiikoiden korjaus migraation jälkeen. Monien grafiikkasivujen muokkaus manuaalisesti vie aikaa, ja suuri pistemäärä vaatii isompaa ja kalliimpaa lisenssiä.

#### 4.2 Aloitusilanne

Uutta migraatiota tehdessä alustava työ on isossa roolissa. Visionissa on oltava mahdollisimman paljon piste- ja historiatietoja, ja molempien valvomoiden IP-osoitteet on asetettava samaksi, jotta migraatio saadaan suoritettua onnistuneesti. Tämän jälkeen on tarkastettava 963-valvomosta kaikki grafiikkakuvat, jotka vaativat objektien poistoa. Esimerkiksi ilmanvaihtokanavan suodatin täytyy poistaa jokaisesta tuloilmakoneen kuvasta. Monia objekteja joudutaan tämän takia poistamaan, jotta sivut kääntyisivät uuteen valvomoon. Kun nämä asiat on tehty, voidaan aloittaa migraatio 963-valvomosta uuteen Vision-valvomoon.

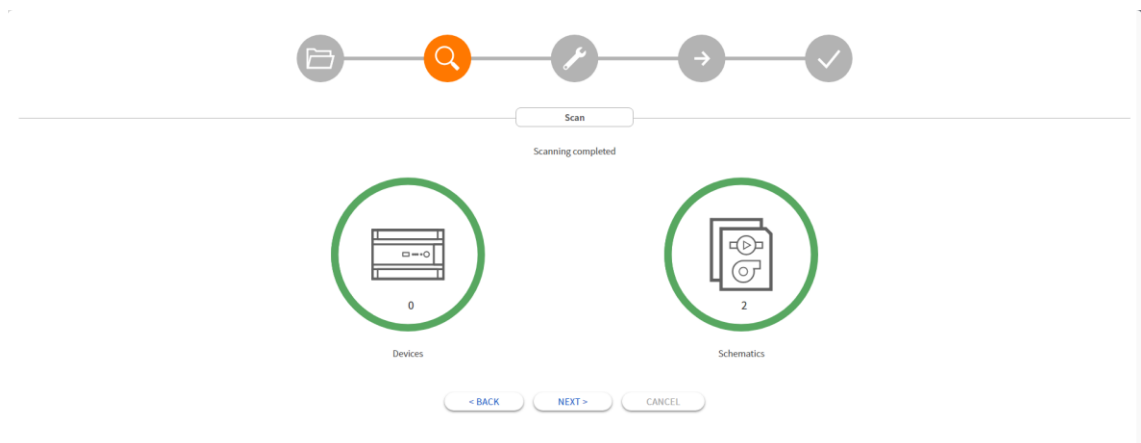
Migraatio tehdään usein ensin omalle tietokoneelle, koska asiakkaan valvomo voi olla vielä keskeneräinen ja tarvittavia lisenssejä ei ole vielä välttämättä saatu. Omalla tietokoneella työn tekeminen on usein helpompaa, koska ei tarvitse mennä asiakkaan tiloihin





Kuva 5 IQ Vision -migraatiotyökalu.

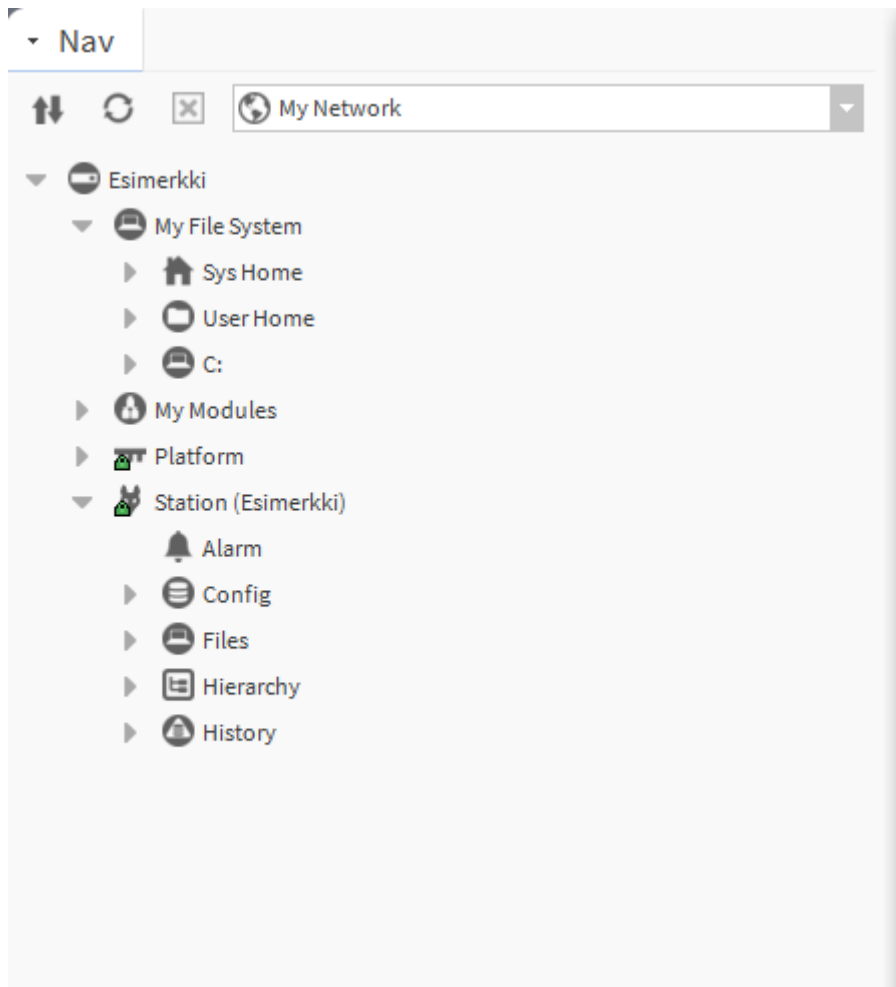
Migraatiotyökalussa valitaan polku, mistä ”Schematics”-kansio löytyy ja työkalu alkaa tuomaan mahdollisia laitteita sekä ”Schematics”-sivuja. Tästä näkee, kuinka paljon sivuja löytyy ja kuinka moni sivu saadaan käännettyä.



Kuva 6 IQ Vision -migraatiotyökalun skannausnäkyminen.

Käännettyt kuvat löytyvät Visionin Nav-puusta ”Files”-kansion alta. Visionissa grafiikkasivujen välillä liikutaan Nav-puun avulla. Sieltä myös löytyvät kaikki valvomon rakentamiseen tarvittavat toiminnot. Esimerkiksi käyttäjien ja hälytysten luominen onnistuu ”Config”-kansion alta.



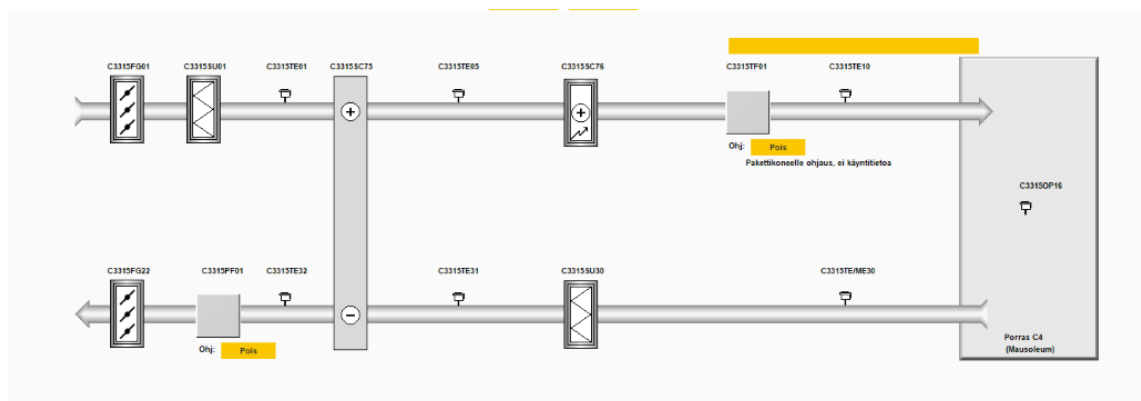


Kuva 7 IQ Visionin Nav-puu. My File System kansion alta löytyvät tietokoneen omat kiintolevyt, ja Stationin alta löytyvät kaikki valvomon rakentamiseen tarvittavat asiat.

#### 4.4 IQ Vision -grafiikkakuvat migraation jälkeen

Kun 963-valvomon kuvia tarkastellaan Visionissa, huomataan, miten keskeneräinen migraatio loppujen lopuksi on. Esimerkiksi migraatio-työkalu luo painikkeita objekteista, jotka eivät ole painikkeita vanhassa valvomossa. Sama toimii myös toisinpäin, eli painikkeet eivät ole painikkeita uudessa valvomossa. Myös kaikki "plot"-pisteet ovat uudessa valvomossa harmaita laatikoita eikä niillä ole mitään ominaisuuksia. Lisäksi tekstejä ei ole suunnattu oikein, ja ne menettävät oikean värinsä migraation jälkeen.

	Käsikytkin	Käsisäätö
TF01 Tulopuhallin	Automaatilla	0,0 % (DN)
PF01 Poistopuhallin	Automaatilla	0,0 % (DN)
SC03 LTO-kiekk	Automaatilla	0,0 % (DN)
FV04 Lämmitysventtiili	Automaatilla	0,0 % (DN)
FV05 Jäähdytysventtiili	Automaatilla	0,0 % (DN)



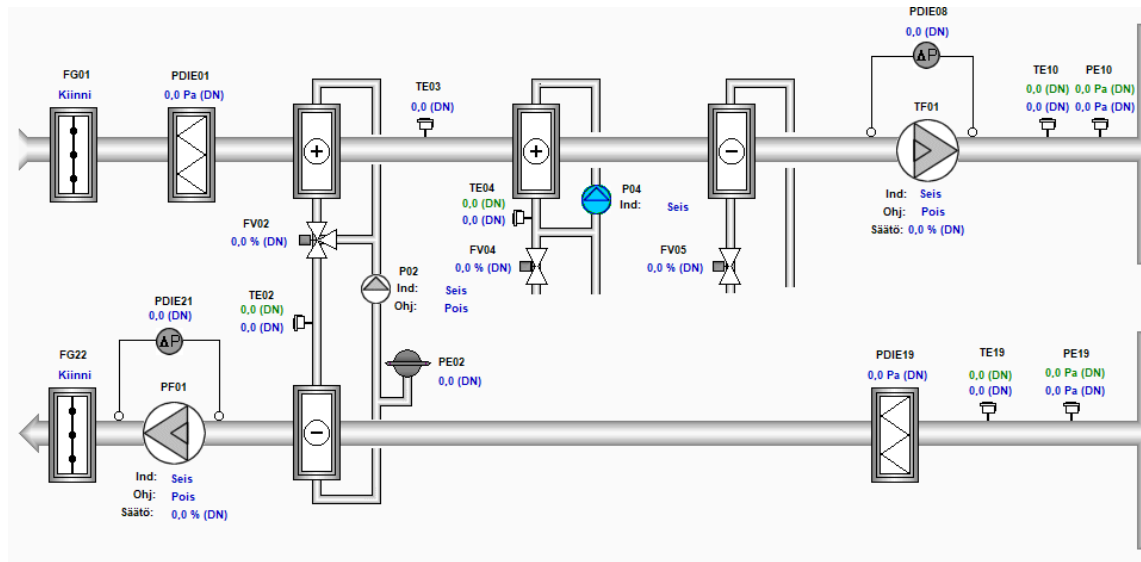
Kuva 8 Grafiikat migraation jälkeen.

#### 4.5 IQ Visionissa tarvittavat muutokset

Jos grafiikat halutaan saada näyttämään samalta, kuin 963-valvomossa pitää tehdä seuraavia muutoksia.

- Kaikki tekstit pitää saada kohdistettua oikein
- tarvittavien tekstien värit pitää palauttaa oikeiksi
- painikkeet pitää saada muutettua toimiviksi ja värykset oikeiksi

- ”plot”-pisteiden luomat harmaat laatikot pitää saada muutettua oikeiksi objekteiksi ja niille pitää luoda uudet ”custom”-pisteet
- tarkastaa kaikki linkit ja huomioida Visionin mukana tulleita uusia ominaisuuksia
- tarkastaa yksiköt ja muuttaa pistetietojen desimaalit.



	Käsikytkin	Käsisäätö
D342 SC01 Taajuusmuuttaja	Automaatilla	0,0 % (DN)
D342 SC02 Taajuusmuuttaja	Automaatilla	0,0 % (DN)
D342 SC03 Taajuusmuuttaja	Automaatilla	0,0 % (DN)
D342 FV02 LTO-venttiili	Automaatilla	0,0 % (DN)
D342 FV04 Lämmitysventtiili	Automaatilla	0,0 % (DN)
D342 FV05 Jäähdytysventtiili	Automaatilla	0,0 % (DN)
D342 FV05.1 Jäähdytysventtiili	Automaatilla	0,0 % (DN)

Kuva 9 Vision-grafiikoiden tavoitenäkymä.

Pelkkien grafiikoiden lisäksi Visionissa pitää tehdä muutoksia pistetietokantoihin ja luoda käyttäjiä, jotta valvomoa pystyy käyttämään selaimessa. Esimerkiksi vanhassa valvomossa prosenteissa ei ole desimaaleja, mutta uudessa valvomossa ne tulevat automaattisesti ja tämä pitää muuttaa vanhan valvomon kaltaiseksi. Myös vanhassa valvomossa celsius-yksiköillä olleet pisteet ovat Visionissa ilman yksikköä, joten niihin pitää lisätä yksiköt. Kohteesta riippuen myös ”custom”-pisteiden teko voi olla todella työlästä, koska niitä voi olla satoja.

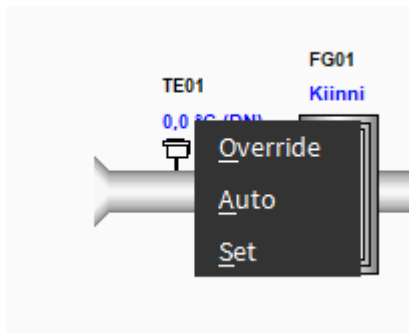


Name	Type	Out	Trend Item Reference	Linked Trend Plots
B33(S0)	String Point	-(down_stale)	B33(S0)	
G3TK05-FV04 LÄMMITYSENTILI	Numeric Writable	0% (down_stale) @ def	D1	
G3TK05-FV02 LTO-VENTTIILI	Numeric Writable	0% (down_stale) @ def	D2	
G3TK05-FV14 JÄLKILÄMMITYSENT.	Numeric Writable	0% (down_stale) @ def	D3	
G3TK05-FV07 JÄLKILÄMMITYSENT.	Numeric Writable	0% (down_stale) @ def	D4	
G3TK05-SC08 SÄÄTÖ	Numeric Writable	0% (down_stale) @ def	D5	
G3TK05-FV05 JÄÄHDYTYSENTILI	Numeric Writable	0% (down_stale) @ def	D6	
G3TK05-SC21.1 SÄÄTÖ	Numeric Writable	0% (down_stale) @ def	D7	
G3TK05-SC21.2 SÄÄTÖ	Numeric Writable	0% (down_stale) @ def	D8	
G31J08-HK01 HOYRYKOSTUTIN	Numeric Writable	0% (down_stale) @ def	D9	
G31J09-HK01 HOYRYKOSTUTIN	Numeric Writable	0% (down_stale) @ def	D10	
G3TK05-PF01 POISTOPUHALLIN	Boolean Writable	false (down_stale) @ def	D11	
G3TK05-PF02 POISTOPUHALLIN	Boolean Writable	false (down_stale) @ def	D12	
G3TK05-FG01 TULOPELTI	Boolean Writable	false (down_stale) @ def	D13	
G4JJ15-16 VIK PARKKO-OHJAUS	Boolean Writable	false (down_stale) @ def	D14	history:SITE00001_L001_D011,P0100
G3TK05-FG22.2 POISTOPELTI	Boolean Writable	false (down_stale) @ def	D15	
G3TK05-PF03 POISTOPUHALLIN	Boolean Writable	false (down_stale) @ def	D16	
G3TK05-P02 LTO-PUMPPU	Boolean Writable	false (down_stale) @ def	D17	
G4JJ08-P04 JÄÄHDYTYSPUMPPU	Boolean Writable	false (down_stale) @ def	D18	
G4JJ07-P04 JÄÄHDYTYSPUMPPU	Boolean Writable	false (down_stale) @ def	D19	
G4JJ09-VIK01 KÄYNNINESTO	Boolean Writable	false (down_stale) @ def	D20	
G4JJ10-VIK01 KÄYNNINESTO	Boolean Writable	false (down_stale) @ def	D21	
G3PK04-PF01 POISTOPUHALLIN	Boolean Writable	false (down_stale) @ def	D22	
G3PK05-PF01 POISTOPUHALLIN	Boolean Writable	false (down_stale) @ def	D23	

Kuva 10 Alakeskuksen pistetietokanta Visionissa.

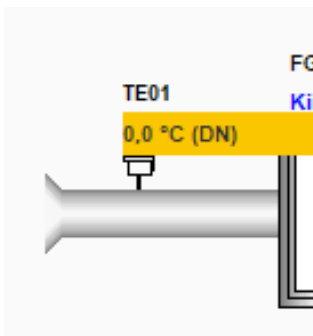
#### 4.6 Grafiikoiden toiminta ja muutokset ”Px Editor” -tilassa

Tarkastellaan aluksi muutoksia ”Px Editor” -tilassa, eli graafisessa muokkausnäkyvässä. Visionissa on monia ominaisuuksia, joita 963-valvomossa ei ole ja niiden käyttö voi olla turhan monimutkaista asiaan perehtymättömälle huoltokäyttäjälle. Esimerkiksi pisteitä voi tarvittaessa muokata suoraan grafiikalta hiiren oikeaa painiketta painamalla, mutta tämä voi aiheuttaa myös muokkauksia vahingossa, joten ominaisuus otetaan usein pois käytöstä. Kyseessä on ominaisuus nimeltä ”popupEnabled” ja sen saa poistettua käytöstä grafiikalta, kun menee objektin ominaisuuksiin ja asettaa sen ”false”-tilaan.



Kuva 11 "popupEnabled"-ominaisuus grafiikalla.

Jos migraatio on tehty omalle koneelle eikä tietokone ole yhteydessä kohteen alakeskuksiin, ovat kaikki pisteet "down"-tilassa. Tällöin Vision antaa pisteille keltaisen värityksen grafiikalle. Tämä nolaa pisteiden väritiedot ja muuttaa ne mustiksi, vaikka oikea väri olisi jokin muu. Pisteiden väriä voidaan muokata, kun tuo ominaisuus on poistettu käytöstä. Tämä ominaisuus on nimeltä "statusEffect". Se saadaan poistettua käytöstä, kun mennään objektin ominaisuuksiin ja asetetaan "statusEffect"-ominaisuuden tila "Color And Blink", "None"-tilaan.



Kuva 12 Pistein väritys, kun alakeskus on "down"-tilassa ja "statusEffect" on käytössä.

Pisteille on hyvä myös asetta ominaisuus "halign" tilaan "Left", koska 963-valvomossa kaikki objektit on vakiona suunnattu vasemmalle. Kun nämä asiat on muutettu kaikista yksittäisistä objekteista, päästään jo pidemmälle migraatiossa, mutta tämä on todella

työläs ja raskas tapa tehdä muutoksia, koska pisteitä voi olla kymmeniä tuhansia.

Bound Label	
background	(default)
blink	false
border	0.0 none black
enabled	true
font	bold 9.0pt Arial
foreground	blue
halign	Left
image	null
layer	
layout	374.0,339.0,84.0,19.0
mouseOver	None
padding	0.0
text	station: slot:/Drivers/TREND/SITE00001/L001/O073/points/S0001
textIconGap	4, 00
textToIconAlign	Right
valign	Center
visible	true
wordWrapEnabled	false
Popup Binding	
ord	history:/SITE00001_L001_O073/P0023
degradeBehavior	None
title	Pop up
position	100.0,100.0
size	800.0,600.0
modal	false
Galileo Bound Label Binding	
ord	station: slot:/Drivers/TREND/SITE00001/L001/O073/points/S0001
degradeBehavior	None
hyperlink	null
summary	%displayName?typeDisplayName% = %.%
popupEnabled	false
statusEffect	None
visibilityPin	-1
actionPin	0

Kuva 13 Sensorin ominaisuus -sivu, jossa kaikki muutokset on tehty.

Seuraavaksi muutoksia vaativat painikkeet. Migraation jälkeen painike ei toimi, koska Visionissa sen toimintaan vaaditaan ominaisuuksissa erillinen osio nimeltä "Action Binding". Tämän takia jokainen painike on käytävä läpi ja tehtävä vaadittavat muutokset ominaisuuksissa. Myös itse painike on muutettava "BoundLabel"-objektista "ImageButton"-objektiksi. Tällä hetkellä nopein tapa tehdä tämä on luoda yksi oikein tehty painike ja liittää siihen erikseen jokaisen pisteen osoitetieto yksitellen.

Galileo Bound Label Binding		X
ord	station: slot:/Drivers/TREND/SITE00001/L001/O074/points/K0007	...
degradeBehav	None	▼
hyperlink	null	...
summary	%displayName?typeDisplayName% = %.%	...
popupEnable	false	▼
statusEffect	None	▼
visibilityPin	-1	
actionPin	0	
Action Binding		X
ord	station: slot:/Drivers/TREND/SITE00001/L001/O074/points/K0007/set	...
degradeBehav	None	▼
widgetEvent	actionPerformed	▼
actionArg		...
<input type="button" value="OK"/> <input type="button" value="Cancel"/>		

Kuva 14 Toimivan "ImageButton"-objektin ominaisuudet.

Tämän jälkeen jäljelle jää hyperlinkkien korjaus. Migraatiotyökalu kääntää monet linkit toimimaan väärällä tavalla, ja tämä pitää korjata. Historiatiedot toimivat jo valmiiksi oikein, mutta esimerkiksi asetusarvojen ja käsikäyttöjen ikkunat avautuvat väärin. Näiden sivujen pitäisi aueta uuteen ikkunaan, mutta ne avautuvat samalle sivulle. Tämä korjataan siirtämällä ja poistamalla linkki "hyperlink"-sarakeesta ja luodaan "Popup Binding"-ominaisuus objektille. Tämän jälkeen linkki toimii oikein. Väärin toimivia linkkejä löytyy aina tuloilmakoneiden ja jäähdytysjärjestelmien sivuilta.





```

<BoundLabel layout="1018.0,412.0,84.0,19.0" font="bold 9.0pt Arial" fore-
ground="green" halign="left">
  <GalileoValueBinding ord="station:|slot:/Driv-
ers/TREND/SITE00001/L001/O035/points/USR_F63_0" popupEnabled="false" ac-
tionPin="0">
    <GalileoStatusToSimple name="text">
      <String name="normal" value="%out.value%  "/>
    </GalileoStatusToSimple>
  </GalileoValueBinding>
  <PopupBinding ord="history:SITE00001_L001_O035/P0020"/>
</BoundLabel>

```

Esimerkkikoodi 2. Objekti tekstieditorissa, kun "popupEnabled" on asetettu "false"-tilaan.

Tästä voidaan huomata, että popupEnabled="false" ilmestyy tekstitiedostossa "actionPin"-tekstin ja objektin pistetiedon väliin.

Seuraavaksi tarkastellaan, mitä muutoksia tapahtuu, kun objektista muutetaan "statusEffect"-ominaisuutta.

```

<BoundLabel layout="1018.0,427.0,84.0,19.0" font="bold 9.0pt Arial" fore-
ground="blue" halign="left">
  <PopupBinding ord="history:/SITE00001_L001_O035/P0081"/>
  <GalileoBoundLabelBinding ord="station:|slot:/Driv-
ers/TREND/SITE00001/L001/O035/points/S0011" actionPin="0">
    <GalileoStatusToSimple name="text">
      <String name="normal" value="%out.value%  "/>
      <String name="override" value="%out.value%  "/>
      <String name="alarm" value="%out.value%  "/>
      <String name="down" value="%out.value% (DN)"/>
      <String name="fault" value="%out.value% (F)"/>
      <String name="disabled" value="%out.value% (D)"/>
    </GalileoStatusToSimple>
  </GalileoBoundLabelBinding>
</BoundLabel>

```

Esimerkkikoodi 3. Tekstinäkymä, kun statusEffect="color and blink" ja popupEnabled="true"

```

<BoundLabel layout="1018.0,427.0,84.0,19.0" font="bold 9.0pt Arial" fore-
ground="blue" halign="left">
  <PopupBinding ord="history:/SITE00001_L001_O035/P0081"/>
  <GalileoBoundLabelBinding ord="station:|slot:/Driv-
ers/TREND/SITE00001/L001/O035/points/S0011" popupEnabled="false" sta-
tusEffect="none" actionPin="0">
    <GalileoStatusToSimple name="text">
      <String name="normal" value="%out.value%  "/>
      <String name="override" value="%out.value%  "/>
      <String name="alarm" value="%out.value%  "/>
      <String name="down" value="%out.value% (DN)"/>
      <String name="fault" value="%out.value% (F)"/>
      <String name="disabled" value="%out.value% (D)"/>
    </GalileoStatusToSimple>
  </GalileoBoundLabelBinding>
</BoundLabel>

```

Esimerkkikoodi 4. Tekstinäkymä, kun "statusEffect" ja "popupEnabled" on muutettu.

Esimerkkikoodeista voidaan huomata, että objektin ollessa vakiotilanteessa, ei tekstieditorissa näy ominaisuuksia ollenkaan. Tämä aiheuttaa vaikeuksia, kun halutaan lisätä ominaisuuksia ”etsi ja korvaa” -menetelmällä. Myös ”halign=left”-ominaisuus pitäisi pystyä lisäämään samalla menetelmällä, mutta sekin on vakiona poissa tekstieditorista.

Seuraavaksi tutkitaan painikkeen eroavaisuuksia käännön jälkeen ja toimivan painikkeen välillä.

```
<BoundLabel layout="180.0,55.0,81.0,17.0" font="bold 9.0pt Arial"
halign="left" foreground="#0000a0" background="silver">
  <GalileoBoundLabelBinding ord="station:|slot:/Drivers/TREND/SITE00001/L001/O035/points/W0016" popupEnabled="false" statusEffect="none" actionPin="0">
    <GalileoBooleanToSimple name="text">
      <String name="on" value="K&#xe4;sik&#xe4;yt&#xf6;ll&#xe4;" />
      <String name="off" value="Automaatilla" />
      <String name="overrideOn"
value="K&#xe4;sik&#xe4;yt&#xf6;ll&#xe4;" />
      <String name="overrideOff" value="Automaatilla" />
      <String name="alarmOn" value="%out.value%" />
      <String name="alarmOff" value="" />
      <String name="down" value="%out.value% (DN)" />
      <String name="disabled" value="%out.value% (D)" />
      <String name="fault" value="%out.value% (F)" />
    </GalileoBooleanToSimple>
  </GalileoBoundLabelBinding>
</BoundLabel>
```

#### Esimerkkikoodi 5. Painike migration jälkeen.

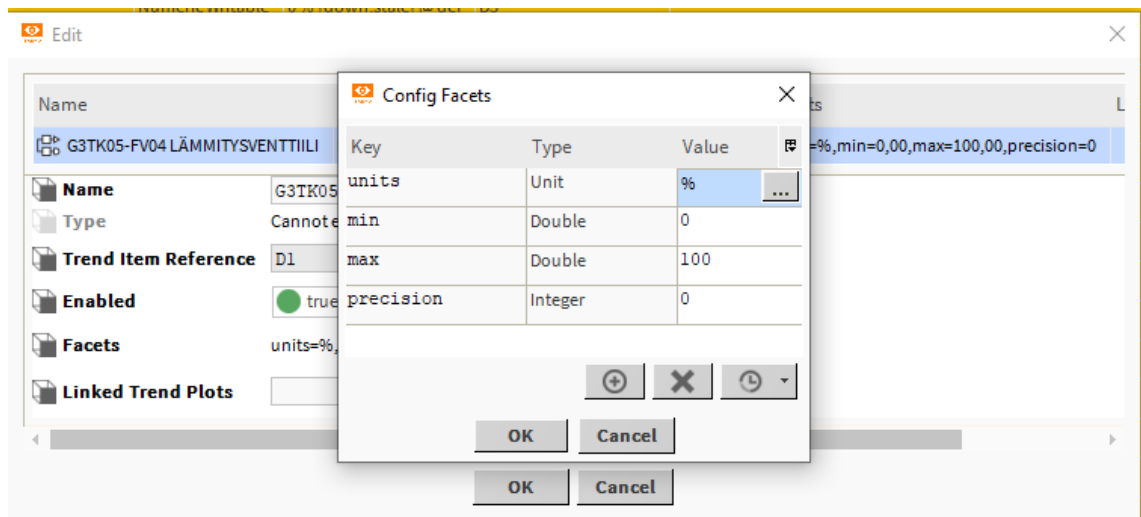
```
<ImageButton layout="180.0,55.0,81.0,17.0" font="bold 9.0pt Arial"
halign="left" foreground="#0000a0" background="silver">
  <GalileoBoundLabelBinding ord="station:|slot:/Drivers/TREND/SITE00001/L001/O035/points/W0016" popupEnabled="false" statusEffect="none" actionPin="0">
    <GalileoBooleanToSimple name="text">
      <String name="on" value="K&#xe4;sik&#xe4;yt&#xf6;ll&#xe4;" />
      <String name="off" value="Automaatilla" />
      <String name="overrideOn"
value="K&#xe4;sik&#xe4;yt&#xf6;ll&#xe4;" />
      <String name="overrideOff" value="Automaatilla" />
      <String name="alarmOn" value="%out.value%" />
      <String name="alarmOff" value="" />
      <String name="down" value="%out.value% (DN)" />
      <String name="disabled" value="%out.value% (D)" />
      <String name="fault" value="%out.value% (F)" />
    </GalileoBooleanToSimple>
  </GalileoBoundLabelBinding>
  <ActionBinding ord="station:|slot:/Drivers/TREND/SITE00001/L001/O035/points/W0016/set" widgetEvent="actionPerformed"/>
</ImageButton>
```

#### Esimerkkikoodi 6. Painike tarvittavien muutoksien jälkeen.

Huomataan, että koko objektin kuvaus ei ole enää "BoundLabel" vaan "ImageButton". Tämä muuttaa grafiikalla painikkeen napin näköiseksi. Seuraava huomattava muutos on toiseksi viimeisellä rivillä. Se on kokonaan uusi rivi verrattaessa migraation jälkeiseen painikkeeseen. Tämä rivi määrittää painikkeen toiminnan ja on siis välttämätön osa toimintaa ajatellen.

#### 4.8 Pisteiden muokkaukset

Pisteiden muokkaus tapahtuu Visionin pistetietokannassa, missä pisteitä muokataan yksitellen tai pientä määrää kerrallaan. Kun halutaan muuttaa tai antaa pisteelle yksikkö, se tapahtuu klikkaamalla kyseistä pistettä ja menemällä sen ominaisuuksiin ja etsiä valikosta haluttu yksikkö. Sama prosessi pitää tehdä myös pisteille, joissa halutaan poistaa tai lisätä desimaaleja. Tämä on todella työläs ja hidas tapa, jos pisteitä on paljon.



Kuva 16 Pisteiden ominaisuus -sivu.

#### 4.9 Trendipisteet

Trendipisteet, eli historia- tai "plot"-pisteet, löytyvät samasta hakemistosta pistetietojen kanssa. Pisteiden historiatietojen merkitsemisen lisäksi, niillä on tärkeä osa "custom"-pisteitä luodessa, koska trendipisteistä näkee tarvittavan pisteen luomista varten. Koska

trendipiste saattoi olla normaali piste vanhassa 963-valvomossa, on tämä todella tärkeä ja helpottava tapa etsiä korvaava ja toimiva piste vanhan ”plot”-pisteen tilalle.

Database 96 obj

Name	History Id	Status	State	Last Success	Plot Number	Plot Source	Plot Interval	Plot Capacity	Record Value Size (bits)
🚩 (G3TK05-TE10 TUOLOILMA)[G3TK05-TE10 TUOLOILMA][P1][S1V][Synchronized@300]	/SITE0001_L001_O011/P0001	[fault_down]	Pending	24.marra.2020 03:00 EET	1	S1	5 minutes	1000	32
🚩 (G3TK05-ME19 POISTOKOSTEUS)[G3TK05-ME19 POISTOKOSTEUS][P2][S2V][Synchronized@300]	/SITE0001_L001_O011/P0002	[fault_down]	Pending	24.marra.2020 02:31 EET	2	S2	5 minutes	1000	32
🚩 (G3TK05-TE18.1 POISTOILMA)[G3TK05-TE18.1 POISTOILMA][P3][S3V][Synchronized@300]	/SITE0001_L001_O011/P0003	[fault_down]	Pending	24.marra.2020 03:00 EET	3	S3	5 minutes	1000	32
🚩 (G3TK05-TE04.1 PALLUUVESI)[G3TK05-TE04.1 PALLUUVESI][P4][S4V][Synchronized@300]	/SITE0001_L001_O011/P0004	[fault_down]	Pending	24.marra.2020 02:02 EET	4	S4	5 minutes	1000	32
🚩 (G3TK05-TE16.1 HUONELMA)[G3TK05-TE16.1 HUONELMA][P5][S5V][Synchronized@300]	/SITE0001_L001_O011/P0005	[fault_down]	Pending	24.marra.2020 03:00 EET	5	S5	5 minutes	1000	32
🚩 (G3TK05-TE16.2 HUONELMA)[G3TK05-TE16.2 HUONELMA][P6][S6V][Synchronized@300]	/SITE0001_L001_O011/P0006	[fault_down]	Pending	24.marra.2020 02:31 EET	6	S6	5 minutes	1000	32
🚩 (G3TK05-TE02.2 TULO LTO JÄLKEEN)[G3TK05-TE02.2 TULO LTO JÄLKEEN][P7][S7V][Synchronized@300]	/SITE0001_L001_O011/P0007	[fault_down]	Pending	24.marra.2020 03:01 EET	7	S7	5 minutes	1000	32
🚩 (G3TK05-PE02 LTO-PAINE)[G3TK05-PE02 LTO-PAINE][P8][S8V][Synchronized@300]	/SITE0001_L001_O011/P0008	[fault_down]	Pending	24.marra.2020 11:05 EET	8	S8	5 minutes	1000	32
🚩 (G3TK05-TE04 TULO LPN JÄLKEEN)[G3TK05-TE04 TULO LPN JÄLKEEN][P9][S9V][Synchronized@300]	/SITE0001_L001_O011/P0009	[fault_down]	Pending	24.marra.2020 02:31 EET	9	S9	5 minutes	1000	32
🚩 (G3TK05-TE05 TULO JÄÄHD.JÄLKEEN)[G3TK05-TE05 TULO JÄÄHD.JÄLKEEN][P10][S10V][Synchronized@300]	/SITE0001_L001_O011/P0010	[fault_down]	Pending	24.marra.2020 02:31 EET	10	S10	5 minutes	1000	32
🚩 (G3TK05-PDE01 TULOSUODATTIN)[G3TK05-PDE01 TULOSUODATTIN][P11][S11V][Synchronized@300]	/SITE0001_L001_O011/P0011	[fault_down]	Pending	24.marra.2020 03:00 EET	11	S11	5 minutes	1000	32
🚩 (G3TK05-TE18.2 POISTOILMA)[G3TK05-TE18.2 POISTOILMA][P12][S12V][Synchronized@300]	/SITE0001_L001_O011/P0012	[fault_down]	Pending	24.marra.2020 03:00 EET	12	S12	5 minutes	1000	32
🚩 (G3TK05-TE02.1 LTO-NESTE)[G3TK05-TE02.1 LTO-NESTE][P13][S13V][Synchronized@300]	/SITE0001_L001_O011/P0013	[fault_down]	Pending	24.marra.2020 03:00 EET	13	S13	5 minutes	1000	32
🚩 (G4JJ06-TE51.2 JÄÄHDYTYSMENO)[G4JJ06-TE51.2 JÄÄHDYTYSMENO][P14][S14V][Synchronized@300]	/SITE0001_L001_O011/P0014	[fault_down]	Pending	24.marra.2020 03:00 EET	14	S14	5 minutes	1000	32
🚩 (G4JJ06-TE52.3 JÄÄHDYTYSPALUU)[G4JJ06-TE52.3 JÄÄHDYTYSPALUU][P15][S15V][Synchronized@300]	/SITE0001_L001_O011/P0015	[fault_down]	Pending	24.marra.2020 03:00 EET	15	S15	5 minutes	1000	32
🚩 (G4JJ06-PE01 VERKOSTONPAINE)[G4JJ06-PE01 VERKOSTONPAINE][P16][S16V][Synchronized@300]	/SITE0001_L001_O011/P0016	[fault_down]	Pending	24.marra.2020 03:00 EET	16	S16	5 minutes	1000	32

Kuva 17 Alakeskuksen historiapistetietokanta.

#### 4.10 ”Custom”-pisteen luominen

”Custom”-piste tarkoittaa usein G- tai F-kirjaimella alkavaa ohjelmallista pistettä. Näitä pisteitä käytetään, jos halutaan luoda piste, joka vaatii matemaattista yhtälöä muista pisteistä. Migraatiossa ”custom”-pisteitä joudutaan tekemään, koska ”plot”-pisteitä ei pysyntyä uudessa valvomossa käyttämään. ”Custom”-pisteiden määrä vaihtelee vanhan valvomon tekotavan mukaan. Jossain valvomoissa näitä pisteitä on vähän ja jossain niitä on paljon verrattuna normaaleihin pisteisiin.

Kuva 18 "Custom"-pisteen historiatieto.

"Custom"-piste luodaan pistetietokantaan "plot"-pisteen tietojen perusteella. "Plot"-pisteestä saadaan selville "custom"-pisteen nimi ja pistetieto. "Custom"-pisteessä tarvitaan numerojen perään vielä jokin kirjain, joka on usein D tai O. Kuva 18 esimerkissä nähdään, että pisteen F146 perään tulee vielä kirjain D, koska trendinpisteen nimessä näkyy tarkka "custom"-pistetieto. Piste luodaan pistetietokannan alla olevalla "New"-napilla. Tämän jälkeen määritellään, onko piste "Booleanwritable", eli "true"- tai "false"-piste, vai "Numericwritable", eli numeerinen piste. Tämän jälkeen lisätään pisteen tiedot ja tarvittava kirjain sulkeiden sisään, jonka jälkeen piste on valmis.

? G3TK05-TE04.1 ASETUSARVO	NumericWritable	0,0 °C [down_state] @ def	F120(D)
? G3TK05-TE05 ASETUSARVO	NumericWritable	0,0 °C [down_state] @ def	F150(D)
? Asetus	NumericWritable	0,0 °C [down_state] @ def	F93(D)

Kuva 19 "Custom"-pisteet pistetietokannassa lisättyinä.

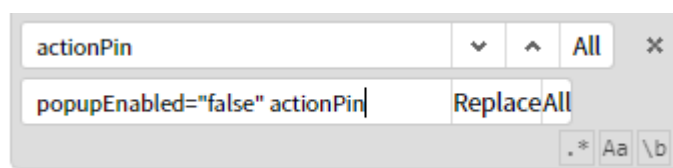
## 5 Migraation automatisointi

### 5.1 Grafiikoiden muokkaus ”etsi ja korvaa” -tyyllillä

Grafiikoita tarkastellessa huomattiin, että manuaalisen työn määrä on iso ja virheiden mahdollisuus kasvaa suuremman työmäärän mukaan. Tämän takia ”etsi ja korvaa” -tapa on hyvä keino virhemarginaalin pienentämiseen, ja toistuvan työn tekeminen onnistuu kerralla.

Tekstieditoria tutkiessa huomattiin, että objektien ominaisuudet ilmestyivät tekstitiedoston vasta muokkauksia tehdessä. Tämä aiheuttaa vaikeuksia, jos halutaan keksiä tapoja lisätä tekstiä tiettyihin väleihin. Jos esimerkiksi ”popupEnabled”-teksti olisi vakiotilanteessa tekstieditorissa, olisi tämän muokkaus helppoa massana, mutta vakiotilanteessa sitä ei näy ollenkaan. Monien muidenkin ominaisuuksien kanssa on sama tilanne, joten on keksittävä kaikille teksteille yhteinen tekijä tekstieditorissa.

Aloitetaan ”popupEnabled”-tekstin selvityksestä. Kuten esimerkkikoodi 2:ssa esitettiin, voidaan huomata, että ”popupEnabled=”false”-teksti ilmestyy aina ”actionPin”-tekstin eteen. Myös muissakin objekteissa on olemassa ”popupEnabled”-ominaisuus ilman ”actionPin”-ominaisuutta, mutta näissä objekteissa ei hiiren oikeaa näppäintä painamalla tapahdu kuva 11:stä esitettyä tilannetta. Tästä voidaan siis tehdä päätelmä, että ”etsi ja korvaa” tavalla voidaan korjaus tehdä etsimällä tekstiä ”actionPin” ja korvata se tekstillä ”popupEnabled=”false actionPin”.



Kuva 20 ”Etsi ja korvaa” -työkalu IQ Visionissa.

Seuraavaksi tarkastellaan ”statusEffect”-ominaisuutta. Tämä ominaisuus määrittää pisteen värin sen tilan mukaan. Kun migraatio tehdään ”offlinessa”, on tärkeää saada tämä ominaisuus pois käytöstä ennen grafiikkasivujen tarkastelua. Ominaisuus on vakiona ”Color And Blink” -tilassa, joka ei tekstieditorissa näy. Kun ”statusEffect” -ominaisuus

asetetaan "None"-tilaan, voidaan huomata, että se ilmestyy "popupEnabled"-ominaisuuden kanssa samalle riville. Nämä muutokset voidaan siis yhdistää hakemalla tekstiä "actionPin" ja korvataan tekstillä " popupEnabled="false" statusEffect="none" actionPin". Tämän muutoksen jälkeen grafiikoiden väritiedot jäävät tallelle eikä niitä tarvitse muuttaa käsin, mikä säästää aikaa migraatiossa merkittävästi. Muutos pitää silti palauttaa takaisin normaalitilaan, kun migraatio on saatu tehtyä ja Vision on siirretty kohteen toimivaan valvomoon, koska pistetietojen värit ovat tärkeä osa valvomon toimintaa. Tämä onnistuu hakemalla "statusEffect="none""-tekstiä ja korvaamalla sen tyhjällä haulilla.

Viimeisenä muutoksena Visionin tekstieditorissa on tekstien suuntaus vasemmalle. Tämä tapahtuu asettamalla "halign"-ominaisuus "Center"-tilasta "Left"-tilaan. Vakiotilanteessa ei ominaisuutta näy tekstieditorissa, joten pitää löytää paikka, mihin ominaisuus ilmestyy muutoksen jälkeen. Eri objekteissa ominaisuus on eri kohdissa tekstiä, mutta iso osa teksteistä on "Arial"-fontissa, minkä kanssa samalle riville "halign"-ominaisuus tulee. Tämän näkee esimerkkikoodeissa ensimmäisellä rivillä. Visionin tekstieditori ei aseta objektin ominaisuuksia tarkkaan järjestykseen vaan lukee ne, jos ominaisuudet on oikein kirjoitettu ja oikeilla riveillä. Tämän ansiosta komento "halign="left"" voidaan asettaa tekstin "Arial" jälkeen. Eli etsitään teksti "Arial" ja korvataan se tekstillä ""Arial" halign="left"".

Edellä mainitulla "etsi ja korvaa" -menetelmällä voidaan monien objektien muutoksia tehdä "massana" ja tämä nopeuttaa migraatioprosessia sekä pienentää virheiden mahdollisuuksia. Edellä mainittujen esimerkkien lisäksi tätä menetelmää voidaan hyödyntää ja soveltaa moneen eri grafiikan osioon Visionissa, mutta nämä muutokset ovat tärkein osa migraation nopeutumisen edesauttamiseksi. Visionin oma text editor -tila on hyvä työkalu yksittäisten grafiikkasivujen muokkaukseen, mutta jos kohteessa on satoja grafiikkasivuja, ei työkalu nopeuta prosessia parhaalla mahdollisella tavalla. Text editor -tilassa ei voi myöskään tehdä makroja, eli käyttäjän nauhoittamia tehtäviä esimerkiksi näppäimistöllä. Vaikeammat muutokset, kuten painikkeet vaativat makroja, jotta ne saataisiin muutettua automaattisesti. Tätä varten tarvitaan erillinen sovellus, jolla muutoksia voitaisiin tehdä. (6.)

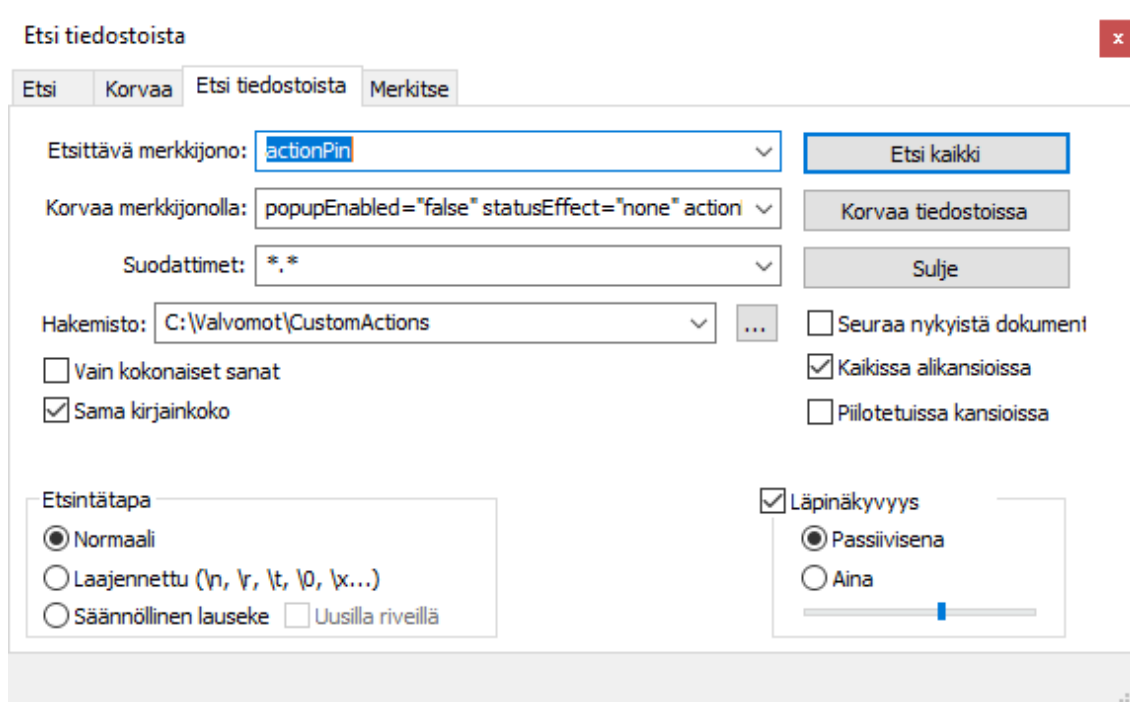
## 5.2 Notepad++

Notepad++ on ilmainen tekstieditoriohjelma, joka tukee monia eri ohjelmointikieliä. Notepad++ on tehokas ja kevyt ohjelma, jossa voi myös luoda ja suorittaa makroja, eli se on sopiva työkalu Visionin tekstitiedostojen muokkaamiseen. (7.)

Tarkastellaan Notepad++:n, tästä eteenpäin Notepad, ominaisuuksia ja sopivuutta Visionin tekstitiedostojen kanssa. Visionin tekstitiedostoja voi käsitellä kahdella tavalla. Niitä voi kopioida sivu kerrallaan ja tehdä muutokset jokaiselle sivulle yksitellen, mutta tämä on hidas tapa, ja se onnistuu myös Visionin omassakin editorissa. Toinen tapa on varmuuskopioida migraation jälkeen koko valvomo Visionista, jonka jälkeen siitä tulee yksi kansio, jota Notepad voi käsitellä.

Notepadissa on työkalu nimeltä ”Etsi tiedostoista”. Tämän avulla voidaan valita haluttu kansio, mistä Notepad hakee merkkijonon ja korvaa sen halutulla merkkijonolla. Tämän avulla kaikki muutokset voidaan tehdä koko valvomoon kerralla ja migraatioprosessi nopeutuu. Tämä myös helpottaa ”statusEffect=’none’”-ominaisuuden poistamista valmiissa valvomossa, koska se voidaan tehdä yhdellä komennolla koko valvomoon. Kaikki edellä mainitut ”etsi ja korvaa” -tyylillä tehdyt muutokset voidaan suorittaa tämän työkalun avulla paljon nopeammin.





Kuva 21 Notepad++ Etsi tiedostoista -näkyvä.

Notepadin makro-ominaisuuden avulla voidaan automatisoida vaikeammatkin tehtävät ja pienentää virheiden mahdollisuutta. Esimerkiksi aikaisemmin esitetyt painikkeille käsittehdyt muutokset voidaan nyt suorittaa makron avulla. Makroja ei voi suorittaa kokonaisille kansioille, joten ne joudutaan suorittamaan sivu kerrallaan. Tämä ei nopeuta prosessia maksimaalisesti, mutta kuitenkin todella paljon. Yhden painikkeen muutostyö käsin vie noin 10 sekuntia, ja painikkeiden määrä sivuilla voi vaihdella parista kappaleesta satoihin kappaleisiin. Makrojen avulla muutos kestää enintään 10 sekuntia jokaista sivua kohden. Makrojen hyödyntäminen migraatioissa on ensisijaisen tärkeää, koska virheen mahdollisuus katoaa lähes kokonaan verrattessa manuaaliseen työhön.

### 5.3 Pistetietokannan muutokset Notepad ++-ohjelman avulla

Visionin pistetietokanta on bog-muotoisena tiedostona valvomon kansiossa. Pienten muutosten jälkeen pistetietokannan saa avattua ja siihen pääsee käsiksi. Tekstitiedosto sisältää koko valvomon konfiguraatitiedot. Nämä tiedot sisältävät kaikki piste-, historia-



## 5.4 Pistetietokannan muutokset IQ Visionin omien työkalujen avulla

Vision tarjoaa Niagara-alustan ansiosta monia työkaluja pistemassan muutoksien nopeuttamiseen, mutta useat työkalut ovat vielä kehitysvaiheessa eivätkä toimi halutulla tavalla. Esimerkiksi Program Service -työkalulla voi etsiä tarkasti haluttuja pisteitä ja muuttaa näihin massana kaikki halutut tiedot. Ongelmana tässä on muutoksien lopullisuus, koska muutoksia ei voi kumota ja tällä työkalulla vääriä muutoksia on helppo tehdä. Myös itse pistetietokannoissa voi valita monia pisteitä kerrallaan ja muuttaa näitä, mutta monissa pisteissä on eri ominaisuuksia, jolloin ne katoavat tällaisessa muutoksessa. Tämän takia suurin osa pisteistä joudutaan muuttamaan yksitellen käsin.

## 6 Automatisoinnin hyöty migraatiossa

### 6.1 Säästetty aika

Opinnäytetyössä esitettyjen migraatiota nopeuttavien tekijöiden ansiosta tulevaisuudessa migraatiossa on vähemmän virheitä ja monia kohteita voidaan päivittää nopeamman aikataululla. Tarkkoja lukemia ajansäästöä ei ole, mutta kaikki edellä mainitut tekniikat yhdistettynä voi melko varmasti sanoa migraation olevan työmäärältään alle 50%:n manuaalisella työllä tehtyyn migraatioon verrattaessa. Esimerkiksi yhden objektin ominaisuuksien muuttamiseen saattaa manuaalisesti mennä jopa minuutti pistettä kohden. Samat muutokset saa nyt tehtyä kerralla koko valvomoon, jolloin muutokseen menevä aika pistettä kohden on olematon. Grafiikoiden muokkaaminen tekstitiedostona mahdollistaa myös grafiikalla olevien pisteiden väritysten säilymisen. Yhden pisteen värinmuutos manuaalisesti voi viedä useita sekunteja, mutta automatisoinnin avulla värit säilyvät grafiikalla ennallaan. Edellä mainittujen asioiden ansiosta tulevaisuudessa tarjoukset voidaan laskea vähemmällä työtunneilla, jonka ansiosta yritys saa työt myös itselleen.

Manuaalisesti joudutaan enää tekemään "custom"-pisteet tietokantaan ja objektien lisäys grafiikalle sekä käsikäyttöjen ja asetusten avautuminen uuteen ikkunaan. Nämä kuitenkin pyritään automatisoimaan tulevaisuudessa. Myös vanhassa 963-valvomossa

tehtävä alustustyö joudutaan vielä tekemään manuaalisesti, mutta vanhassa valvomossa tehtävä työmäärä ei ole ajallisesti kovin iso. Siihenkin voidaan tarvittaessa hyödyntää Notepadin etsi tiedostoista -ominaisuutta.

## 6.2 Virhemarginaalin pieneneminen

Automatisoinnin ansiosta ihmisen tekemä virhemarginaali on vähäisempi kuin manuaalisessa työssä. Migraatio ei ole täysin automatisoitu, mutta isoimmat työt saa silti parilla painalluksella tehtyä, vaikka ennen samaan työhön joutui käyttämään paljon aikaa ja joitain ominaisuuksia saattoi huolimattomuuden takia jäädä tekemättä. Tämän ansiosta migraation valmistuttua ei asiakkaalta tule takuun alle meneviä lisätöitä yhtä paljon verrattaessa manuaaliseen työhön. Esimerkiksi koko painikkeen muutos piti ennen tehdä käsin, jolloin virheitä saattoi tulla helposti, jolloin koko painike ei toiminut. Nyt samanlainen virhe on miltei mahdoton. Virheitä silti löytyy, koska vaikeimpia muutoksia ei voinut Notepadin avulla automatisoida.

## 6.3 Pythonin tarjoamat lisämahdollisuudet

Työn alkuvaiheessa oli tarkoituksena sisällyttää ohjelmointikurssin käyminen ja ohjelman luonti migraation helpottamiseksi. Tärkein syy tälle oli "custom"-pisteiden muutostyön osuus migraatioprosessissa. Jokainen piste joudutaan tällä hetkellä käymään manuaalisesti läpi ja lisäämään pistetietokantaan käsin. Tämä on viimeinen askel migraation miltei täydelliseen lopputulokseen. Ohjelman toteuttaminen opinnäytetyöhön sisällytettynä olisi liian työläs ja iso prosessi, koska Visionin valvomosta selviää lisää uusia ominaisuuksia ja sovellus päivittyy koko ajan. Alkuperäinen idea oli luoda "custom"-pisteet pistetietokantaan sen ollessa tekstimuodossa. Tämä kuitenkin edellyttää melko paljon tietämystä ohjelmoinnista sekä itse Visionin konfiguraatitiedostosta, joka voi isoissa kohteissa olla satoja tuhansia riviä pitkä.

Ohjelmointitaito tarjoaa myös nopeutusta nykyisten painikkeiden makroihin. Makrot ovat lopulta yksinkertaisia käskyjä, minkä takia itse painikkeiden muutostyö on vielä työläs verrattuna "etsi ja korvaa" -tyyliin. Tämän takia kaikki sivut joudutaan käymään yksitellen läpi painikkeita luodessa makrojen avulla. Pythonilla tuo vaihe saadaan mahdollisesti poistettua ja painikkeet luodaan koko valvomoon kerralla.

Python mahdollistaisi kaikkien työssä mainittujen toimintojen yhdistämisen yhdeksi ohjelmaksi, jolloin päivittäminen olisi automatisoitu niin pitkälle, että 963-valvomo menisi aluksi Trendin oman migraatiotyökalun läpi ja tämän jälkeen sille suoritettaisiin Pythonilla tehty ohjelma. Tämän jälkeen grafiikkasivut käytäisiin Visionissa läpi, ja tässä vaiheessa niihin täytyisi tehdä enää pieniä korjauksia. Pythonin tuoma hyöty automatisointiin on iso, joten edellä mainitut ideat pyritään toteuttamaan tulevaisuudessa, mutta ei opinnäyte-työssä.

## 7 Yhteenveto

Insinööriyön tavoitteena oli automatisoida ja helpottaa nykyistä migraatioprosessia uuteen Trendin valvomoon. Työ alkoi tutustumalla ja yrittämällä saada mahdollisimman laaja kuva Trendin uusimmasta IQ Vision -valvomosta ja ymmärtää sen eroja vanhaan 963-valvomoon verrattaessa. Eroavaisuuksien tutkimisen jälkeen siirryttiin migraatioprosessiin, joka sisälsi 96x Schematic Export -työkaluun tutustumista sekä grafiikkakuvien ja pistetietokantojen toimintaan perehtymistä Visionissa ja 963:ssa. Kun migraation isoimmat ja aikaa vievät virheet oli löydetty, alkoi migraation automatisointiin perehtyminen.

Migraation automatisointia varten täytyi Visionin grafiikkakuvia tutkia tekstimuodossa. Muutoksien löytäminen oli melko helppoa, mutta muutoksien tekeminen ”etsi ja korvaa” -tavalla vaati monia erilaisia kokeiluja sekä epäonnistumisia. Lopulta oikeat korvaustavat löytyivät ja grafiikoita pystyi muokkaamaan melko nopeasti. Kun oikeat korjaustyyli oli löydetty, täytyi löytää sovellus tai työkalu, jonka avulla korjauksia voisi tehdä koko valvomoon kerralla. Tähän sopiva työkalu oli Notepad++.

Notepad++ vei migraation nopeuttamisen uudelle tasolle. Ennen grafiikkakuvia editoitiin yksi kuva kerrallaan, mutta Notepad++ mahdollisti koko valvomoon yhdellä kertaa tehtävät muutokset. Tämä nopeutti migraatiota ja pienensi myös virheiden mahdollisuuksia, koska kaikki muutokset tehtiin koko valvomoon. Tämän ansiosta mikään grafiikkakuva ei jäänyt huomioimatta, ja muutokset tulivat voimaan jokaiseen kuvaan.

Notepad++:n makrot tekivät vaikeammat muokkaukset mahdollisiksi. Tämän ansiosta painikkeiden teko helpottui, ja virheiden mahdollisuus pieneni entisestään. Valitettavasti

makroja ei voinut suorittaa kokonaisuvalvomolle kerrallaan, mutta ajansäästö vanhaan tekotapaan verrattaessa oli iso. Lisäksi makrojen ja ”etsi ja korvaa” -menetelmän ansiosta myös koko valvomon pistetietokantaa pystyi muokkaamaan kerralla. Esimerkiksi desimaalit ja yksiköt sai muutettua koko valvomoon tätä kautta.

Automatisoinnista jäi enää pois vaikeammat syvällisempää ohjelmointitaitoa vaativat muutokset, kuten ”custom”-pisteiden teko. Niitä varten olisi pitänyt tehdä erillinen ohjelma, joka osaisi käsitellä isoa tekstitiedostoa sekä ymmärtämään Visionin toimintaa entistäkin paremmin. Ohjelman tekeminen insinööriyön jälkeen on tarkoituksena, koska saatu hyöty on iso. Ohjelman avulla pystyttäisiin suorittamaan viimeisetkin hiomista vaativat tehtävät ja yhdistämään kaikki työssä mainitut toiminnot yhdeksi toimivaksi ohjelmaksi.

Työn isoin tavoite kuitenkin suoritettiin, koska ajansäästöä migraatioprosessiin tuotiin huomasti. Manuaalisella työllä tehtävät muutokset saattoivat viedä jopa kymmeniä minuutteja sivua kohden, ja nyt tämä kaikki saadaan suoritettua muutamalla ”etsi ja korvaa” -toiminnolla sekä makroilla. Voidaan varmasti sanoa, että työhön kuluva aika puolittui ja samalla myös virheiden mahdollisuus pieneni vähintään saman verran. Asiakkaalle tämä tarkoittaa halvempaa päivittämistä uuteen IQ Vision -valvomoon sekä laadukkaampaa työn jälkeä.

## Lähteet

- 1 Yrityksestä. 2020. Verkkoaineisto. Assemblin Intranet. Luettu 18.11.2020.
- 2 About Trend Control Systems Limited. 2020. Verkkoaineisto. <<https://www.trendcontrols.com/en-GB/About/Pages/AboutTrend.aspx>>. Luettu 5.12.2020.
- 3 HVAC Controls. 2020. Verkkoaineisto. <<https://americanenergycare.com/energy-efficiency-solutions/energy-controls/hvac-controls/>> Luettu 5.12.2020.
- 4 Why upgrade to IQVISION from 963? 2020. Verkkoaineisto. <<https://www.trendcontrols.com/en-GB/Pages/IQVISIONvs963.aspx>>. Luettu 11.12.2020.
- 5 IQVISION v2.40[R] Configuration Manual. 2020. PDF-dokumentti. Luettu 12.12.2020.
- 6 Task automation with macros. 2020. Verkkoaineisto. <<https://npp-user-manual.org/docs/macros/>> Luettu 16.12.2020.
- 7 What is Notepad++. 2020. Verkkoaineisto <<https://notepad-plus-plus.org/>> Luettu 16.12.2020.