

**This is an electronic reprint of the original article.  
This reprint *may differ* from the original in pagination and typographic detail.**

**Author(s):** Lassila, Matti; Markkanen, Hannu; Häkkinen, Veli-Matti

**Title:** Koha-kirjastojärjestelmän testaus Jyväskylän yliopiston kirjastossa

**Year:** 2015

**Version:**

**Please cite the original version:**

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

# Koha-kirjastojärjestelmän testaus Jyväskylän yliopiston kirjastossa

Matti Lassila, Hannu Markkanen, Veli-Matti Häkkinen

25.11.2015

# Sisällys

1. Johdanto .....	1
2. Datan migraatio .....	1
3. Integroinnit .....	2
3.1. Melinda-replikointi .....	3
3.2. Finna .....	4
4. Testauskokemuksia .....	4
4.1. Hankintatoimisto .....	4
4.2. Lainaustoimisto .....	5
4.3. Kokoelmatoimisto .....	7
5. Kohan kehitysnäkymiä .....	8
Liite A: Asetukset .....	10
A.1. Kirjastot .....	10
A.2. Aineistotyytit .....	10
A.3. Kokoelmakoodit .....	11
A.4. Asiakkaiden tilastointiryhmät .....	12
A.5. Asiakasryhmät .....	13
A.6. Linaussäännöt .....	14
A.7. Asiakasryhmien määreet .....	14

# 1. Johdanto

Suomalaisten korkeakoulukirjastojen käyttämä Voyager-kirjastojärjestelmä alkaa olla elinkaarensa päässä. Teknisiltä ratkaisuiltaan monin paikoin vanhentunut järjestelmä on toistaiseksi toiminut luotettavasti siitä huolimatta, että yhä useampi korkeakoulukirjasto on siirtynyt käyttämään asiakaskäyttöliittymänään Finnaa, joka kuormittaa kirjastojärjestelmää merkittävästi enemmän kuin aiemmin käytössä ollut WebVoyage-käyttöliittymä.

Keväällä 2015 Jyväskylän yliopiston kirjastossa päätettiin käynnistää Koha-kirjastojärjestelmän testausprojekti, jonka tarkoituksena oli selvittää kirjaston valmiutta siirtyä uuteen kirjastojärjestelmään asiakkaille tällä hetkellä tarjotut palvelut säilyttäen.

Projektin alussa tutustuimme Kohaan Vaara-kirjastojen käyttämän ohjelmistoversion avulla, mutta kirjaston oma testausprojekti toteutettiin Kohan virallisella kehitysversiolla [1: Testauksessa käytettiin Kohan versiota 3.19.00.026] huomioimatta Vaara-kirjastoissa tehtyjä muutoksia.

Selvitystyötä ohjasi kolme kysymystä:

1. Miltä osin kirjasto kykenee itse suoriutumaan järjestelmävaihdoksesta ilman järjestelmätoimittajan tai muun ulkopuolisen tahon apua?
2. Miltä osin Koha täyttää asiakkaille tällä hetkellä tarjottujen palvelujen asettamat vaatimukset?
3. Mitä nykyisten Voyager-järjestelmäintegrointeja vastaavien integrointien toteuttaminen Kohaan vaatii; huomioiden erityisesti Finna ja Melinda?

Sen lisäksi, että tässä raportissa kuvataan testausprojektin aikana tehtyjä ratkaisuja, raporttiin on sisällytetty myös toteuttamatta jääneiden ratkaisuvaihtoehtojen hahmotelmia ja yleisiä Kohaan liittyviä huomioita.

Jyväskylän yliopiston kirjasto tulee siirtymään seuraavaan Voyager-kirjastojärjestelmän versioon yhdessä muiden suomalaisten korkeakoulukirjastojen kanssa vuoden 2016 aikana. Testausprojektin myötä kirjastolla on kuitenkin valmius siirtyä Kohaan vaihtoehtoisena järjestelmänä, mikäli Voyager-kirjastojärjestelmän tuki lakkaa tai eteen tulee teknisiä ongelmia tai palvelutarpeita, joita Voyagerin puitteissa ei voida ratkaista. Työnkulkuja kehittämällä ja joitain lisätoiminnallisuuksia hankkimalla Koha voi korvata Voyager-kirjastojärjestelmän.

## 2. Datan migraatio

Migraatiossa käytettiin apuna skriptejä, joiden alkuperäiskehittäjä on [Bywater Solutions](#). Skriptit eivät olleet sellaisenaan käyttövalmiita, vaan niihin tehtiin muutoksia omien tarpeidemme mukaan. Bywaterin julkaisemista skripteistä voi päätellä, että paikalliset muutokset ovat tyypillisiä myös heidän hoitamissaan migraatioissa—joihinkin työvaiheisiin voi tuottaa yleiskäyttöisiä työkaluja, mutta koska lähtödatan sisältö vaihtelee eri järjestelmien välillä suuresti, paikalliset muutokset skripteihin ovat välttämättömiä. Olemme julkaisseet omiin tarpeisiimme muokatut skriptit [Githubissa](#).

Migraatiossa siirrettiin bibliografiset tietueet, nidetiedot, varaukset, lainaustilanne sekä anonymisoidut asiakastiedot. Sarjajulkaisujen tiedot siirrettiin vain osittain. Voyagerin kokoelma-

ja sijaintiperusteiset aineistotyyppit (*Item Type*) muutettiin migraation yhteydessä laina-ajan pituuteen perustuviksi aineistotyypeiksi (kts. [\[Aineistotyyppit-taulukko\]](#)). Kokoelmatyössä tärkeät sijaintitiedot (Voyagerin *Location Code* ja *Location Name*) muunnettiin suoraan Kohan kokoelmakoodeiksi (CCODE). Mutta mahdollisessa Kohan tuotantokäyttöön tähtäävässä migraatiossa osa Voyagerin sijaintitiedoista voisi olla järkevää tallentaa Kohan sijaintitiedoksi (LOC) ja osa kokoelmakoodeiksi (CCODE), sillä osa nykyisistä sijaintitiedoista perustuu aineiston sijaintiin ja osa aineiston käsittelemään tieteenalaan tai aihepiiriin.

Sarjajulkaisujen varastotiedot — Voyagerin hankintamoduulin kautta hallittavien MARC-kenttien (852, 863, 866 ja 994) sisällöt — siirrettiin migraatiossa Kohan sarjajulkaisumoduulin *subscriptionhistory*-tietokantataulun kenttiin. Kohassa sarjajulkaisujen tiedoista vain bibliografiset tiedot löytyvät MARC-tietueista kun taas [kattavuus](#), [huomautukset puuttuvista numeroista yms.](#) ovat ainoastaan sarjajulkaisumoduulin tietokantatauluissa.

Vufind-integrointia varten olisi hyödyllistä, jos sarjajulkaisu-tietokantataulujen lisäksi varastotiedot löytyisivät myös MARC-tietueista standardinmukaisessa muodossa. Tästä syystä, sen lisäksi että, kattavuuskenttien perusteella muodostetaan Kohaan niteitä joiden fyysinen varastoyksikkö — vuosikerta, vuosikerran osa, yksittäinen numero jne. — vaihtelee julkaisusta riippuen, sarjajulkaisujen tietojen osalta voisi olla järkevää tutkia mahdollisuutta sisällyttää Voyagerin hankintamoduulin tuottamat MARC-kentät Kohaan tuotaviin tietueisiin sellaisenaan.

JYKDOKissa on osakohteita, jotka jätettiin siirtämättä testimigraatiossa ajan säästämiseksi. [Osakohde-emo-linkitys](#) perustuu Kohassa MARC-kenttään 773 tallennettuihin tietoihin. Osakohteet tuodaan järjestelmään *bulkmarcimport.pl*-skriptillä kuten muukin aineisto, jonka jälkeen *create\_analytical\_rel.pl* -skriptin avulla tietueiden välille muodostetaan linkitys. Linkityksessä käytetään tuotujen osakohdetietueiden 773\$o-osakenttaan tallennettua emotietueen aineiston viivakoodia, jonka perusteella osakohdetietueen 773\$0-osakenttään kirjoitetaan emotietueen tietuetunnus (Kohan biblionumber, joka toisin kuin Voyagerissa on oletuksena eri kuin emotietueen 001-kentän arvo) ja 773\$9-osakenttään Kohan nidetunnus. Tuontitiedostoa muodostettaessa osakohteen 773\$w -kenttään voidaan tallentaa emotietueen 001-kentän arvo, jolloin tietueiden mukana saadaan säilymään Kohan tietokantarakenteesta riippumaton yhteys.

Migraatiossa aikaavievin vaihe oli Voyagerista siirrettyjen bibliografisten tietojen ja nidetietojen yhdistäminen Kohan *bulkmarcimport*-skriptin syötteeksi [sopivaan muotoon](#). Käytettäessä Githubista löytyvää *bibliomasher*-skriptin [versiota](#) tuontitiedoston muodostaminen etenee noin 400 tietueen minuuttinopeudella. [2: Tuontitiedostot muodostettiin Mac OS X 10.9.5 käyttöjärjestelmällä varustetulla koneella, jossa oli 2,5GHz Intel Core i5 (I5-3210M)-prosessori, 8GB keskusmuistia ja SSD-massamuisti.]

### 3. Integroinnit

Testausprojektin aikana tutkittiin, miten nykyiset Voyager-integroinnit muihin järjestelmiin voidaan toteuttaa Koha-ympäristössä. Integrointeja ei toteutettu käytännössä, vaan ratkaisuvaihtoehtojen toimivuutta punnittiin järjestelmien dokumentaation ja lähdekoodiin tutustumisen perusteella.

## 3.1. Melinda-replikointi

Melindassa luetteloidut tietueet siirtyvät kirjastojen paikallisiin tietokantoihin Kansalliskirjastossa kehitettyjen replikointiskriptien avulla. Skriptit eivät ole järjestelmäriippumattomia, vaan ne on suunniteltu toimimaan nykyisessä Alehp (Melinda) - Voyager (paikallistietokanta) -ympäristössä. Kohaa varten replikointiskripteihin tulee tehdä muutoksia, jotka alustavan selvitystyön perusteella vaikuttaisivat olevan verrattain helppoja toteuttaa.

Paikalliskannan osalta replikointi perustuu nykyisellään Voyager-tietokannassa ajettaviin SQL-kyselyihin. Kohan tapauksessa SQL-kyselyt on kansallisbibliografiatunnuksella tapahtuvien hakujen osalta — eli hakujen, jotka kohdistuvat paikallisen kannan 035-kenttään — välttämätöntä korvata Kohan hakurajapintoihin perustuvalla toteutuksella. Kohassa MARC-kenttien arvoja ei tallenneta haettavassa muodossa SQL-tietokantaan vaan erilliseen hakuindeksiin, joka nykyisessä julkaisuversiossa on [Zebra](#) ja lähitulevaisuudessa [Elastic Search](#).

Kohassa 035 kenttä ei ole Zebrassa, eikä tulevassa ElasticSearch-haussa oletusarvoisesti haettavissa, mutta on helposti määriteltävissä haettavaksi järjestelmäasetuksista. ElasticSearchin haettavissa olevat kentät tulevat olemaan määriteltävissä selainkäyttöliittymän kautta.

Mikäli Kohan käytössä on Zebra-hakumoottori, tietueen tarkistus voidaan toteuttaa Z39.50- tai SRU-haulla, kunhan kenttä on määritetty haettavaksi Zebran asetustiedostossa. Käytettäessä ElasticSearchia Z39.50 voidaan tarvittaessa ohittaa kokonaan. Paikallisen tietueen olemassaolon tarkistukseen liittyvät SQL-kyselyt voidaan korvata HTTP-pyyntöillä ElasticSearchin hakurajapintaan. HTTP-pyyntöt voivat olla esimerkiksi muotoa:

```
curl -XGET 'http://elasticsearch-server.lib.jyu.fi:9200/koha/biblios/_search?q=other-control-number:648704'
```

haettaessa Melindan tunnisteella 648704 olevaa tietuetta Kohasta silloin kun MARC-kenttä 035 on määritetty haettavaksi nimellä `other-control-number` hakuasetuksissa.

Paikallisen MARC-tietueen nouto voidaan toteuttaa joko Z39.50:n, SRU-haun tai OAI-PMH-rajapinnan kautta. Suunnitteilla on myös sisältöneuvotteluun ([content negotiation](#)) perustuva mahdollisuus pyytää tietue aina kulloiseenkin käyttötarkoitukseen sopivassa muodossa kunkin tietueen yksilöivän URL-osoitteen kautta.

Paikallisen tietueen päivittäminen tehdään replikointiskriptin nykytoteutuksessa `Pbulkimport`-skriptillä, joka on Voyagerin komentorivityökalu tietueiden tuontia tai massamuutoksia varten. Kohassa ei ole vastaavaa, kaikki replikoinnissa tarvittavat toiminnot tarjoavaa työkalua, mutta toiminnot voidaan kuitenkin toteuttaa Kohan nykyisten komentorivityökalujen yhdistelmällä. Kohan `bulkmarcimport.pl`-skriptiä voidaan hyödyntää replikoinnin lisäys- ja muutostoiminnoissa, kun taas tietueen poisto saadaan hoidettua `batchdeletebiblio.pl`-skriptillä. `Pbulkimportin` käyttämät Voyagerin `Bulk Import Rules`-muunnossäännöt saadaan Kohassa toteutettua joko `bulkmarcimportin` parametreillä tai laajentamalla `bulkmarcimportin` toiminnallisuutta Perl-alirutiineilla, joita kutsutaan kunkin tuotavan tiedoston kohdalla. Mahdollisuus laajentaa `bulkmarcimportin` toiminnallisuutta on sisäänrakennettu skriptiin, joten tarvittavat laajennokset saadaan toteutettua hallitusti.

## 3.2. Finna

Kohaa varten on saatavissa ajantasainen [Vufind-ajuri](#), joka on tarkoitettu Vufind 2.4 -versiolle. Ajuri on todennäköisesti muokattavissa yhteensopivaksi Finnan lähtökohtana olleen varhaisemman Vufind 2-version kanssa, mutta testausprojektin yhteydessä muutostyöhön ei kuitenkaan ryhdytty.

Ajurin toteutus perustuu Kohan [ILS-DI](#)-rajapintaan sekä nykyisen Voyager-ajurin tapaan suoriin SQL-kyselyihin Kohan tietokantaan. ILS-DI:n kautta toteutettavissa olevia toimintoja ovat muun muassa aineiston varaaminen, varauksen peruminen ja lainojen uusiminen.

ILS-DI on toteutukseltaan vanhanaikainen rajapinta— sen [pohjatyö](#) on tehty 2007 ja standardimäärittelyn [viimeisin versio](#) (1.1) on julkaistu vuonna 2008.

Kohaan ollaan toteuttamassa teknisesti ajanmukainen [REST-tyyppinen rajapinta](#), joka tulee tarjoamaan nykyisen ILS-DI-rajapinnan tarjoamat toiminnot sekä muun muassa sarjajulkaisujen tietojen käsittelyyn liittyviä toiminnallisuuksia.

Aineistotietojen siirto Finnaan voidaan toteuttaa nykyiseen tapaan OAI-PMH-rajapinnan yli.

## 4. Testauskokemuksia

Kohan testaus jaettiin osiin kirjaston eri toiminnoista — lainaus, hankinta, kokoelmat — vastaavien toimistojen kesken. Näin pyrimme selvittämään miten Koha nykymuodossaan soveltuu toimistojen tämänhetkisiin toimintatapoihin ja mitä kehittämistarpeita nousee esiin, mikäli toimintatavat ja asiakkaille tarjotut palvelut halutaan säilyttää nykyisenlaisena myös järjestelmävaihdoksen jälkeen.

### 4.1. Hankintatoimisto

#### 4.1.1. Kausijulkaisut

- Kausijulkaisujen ilmestymiskaavojen laatiminen saapumisvalvontaa varten on monimutkaista
- tilikarttaa yksinkertaistetaan uuteen kirjastojärjestelmään siirryttäessä
- tietojen tulisi siirtyä saumattomasti Kohan, LibNet-tilaushallintapalvelun ja tulevan ERM-järjestelmän välillä
- Kohan raportointityökaluun on luotavissa hankinnassa tarvittavia raportteja lomakekäyttöliittymän tai [SQL-kyselyiden](#) avulla

#### 4.1.2. Monografiat

- Nykyiselle Voyagerin hankintamoduulin laajuiselle hankintatoiminnallisuudelle ei ole Kohassa tarvetta, sillä rahankäytön yms. seurantaraportit saadaan yliopiston taloushallinnosta
- Kirjatilaukset tulnaisiin viemään Kohaan minimitiedoilla jotta hankinnan tila saadaan näkyviin asiakkaille, muut tiedot ERMiin

## 4.2. Lainaustoimisto

### 4.2.1. Varastotilaukset

Kirjastossa on asiakkailta suljettu varasto, jossa oleviin teoksiin kohdistuvat aineistopyynnöt käsitellään tällä hetkellä Voyagerin *Call Slip* -toiminnon kautta. Nykyisessä Kohan julkaisuversiossa ei ole olemassa vastaavaa toimintoa, joten testauksessa selvitettiin voisiko varastotilauksissa tarvittavat toiminnallisuudet toteuttaa määrittelemällä käyttöön sopivat asetukset. Asetuskombinaatioiden osalta testauksessa kokeiltiin seuraavaa:

- Luotiin lainaussääntö, jonka mukaan hyllyvarausten teko varastokokoelmaan on sallittu
- Pakotettiin varastokokoelmaan nidekohtainen varaus, eli varaus on pakko tehdä aina tiettyyn niteeseen
- Otettiin varastokokoelmassa käyttöön `StaticHoldsQueueWeight` -asetus
- Määritettiin `Transfer`-asetuksissa, ettei aineistoja siirretä pääkirjaston sekä pääkirjaston varaston välillä. Tarkoitus oli kokeilla, estääkö tämä myös varausten siirtymisen em. kirjastojen kokoelmien välillä.

Näillä asetuksilla asiakas saa tehtyä varauksen varastokokoelman niteeseen. Ongelmaksi tulee kuitenkin se, että nimekevarausten tarttumista ei voi rajoittaa aineistotyyppikohtaisesti siten, että asiakkaan itsensä tekemät nidevaraukset olisivat kuitenkin mahdollisia.

Nidevaraustoimintoa voisi käyttää kuitenkin siihen, että kirjastojärjestelmän ulkoista järjestelmää — esimerkiksi tikettijärjestelmää — käyttäen tehdyt varaukset saadaan näkyviin asiakaskäyttöliittymän puolella. Nimekevarausten tarttumisen saa estymään määrittelemällä varastolle omat lainaussäännöt ja estämällä varaukset (*Hold Policy Holds not Allowed*). Silloin kun lisäksi on sallittu henkilökunnan ohittaa asetettu `Hold Policy` (*AllowHoldPolicyOverride*), henkilökunta voi tehdä noudon jälkeen ko. niteeseen varauksen, mutta asiakkaiden nimekevaraukset eivät kuitenkaan tartu näihin varastokappaleisiin. Näin ulkoista järjestelmää voitaisiin käyttää tilausten tekoon sekä käsittelyyn ja samalla asiakaskäyttöliittymään saataisiin näkyville varastotilauksen tilanne. Mahdollisia ulkoisia järjestelmiä voisivat olla esimerkiksi:

#### Viola

- Tukholman yliopiston kirjastossa kehitetty [varastopyyntöjen käsittelyjärjestelmä](#)
- on Tukholmassa tuotantokäytössä, tulossa avoimella lisenssillä julkiseen jakoon
- käyttää Microsoftin palvelinteknologioita, joita yliopistolla ei muuten ole käytössä. Tukea voi olla vaikea saada

#### Tikettijärjestelmä

- käytetään varastopyyntöjen käsittelyyn tikettijärjestelmää (kuten [OTRS](#))
- pyynnöt tehdään [rajapinnan kautta](#) järjestelmään, varaus luodaan asiakkaalle noudon jälkeen, joten näkyy Kohassa
- muokattavissa oleva työnkulku
- tikettijärjestelmälle voi olla tarvetta muissakin [kirjaston työkuluissa](#)



## 4.2.2. Varastopyyntöominaisuuden kehittäminen

Katso kohta Kohan [\[kehitysnäkymistä\]](#).

## 4.2.3. Varausten käsittely

Tällä hetkellä hyllyvarauksen teko tulee mahdolliseksi heti kun ensimmäinen lainakappale kyseiseen aineistotyyppiin kuuluvasta teoksesta on lainattu. Tarjolla on ollut korjaus, joka muuttaa toiminnan siten, että vasta kun kaikki lainakappaleet ovat lainattu, varauksen tekeminen tulee mahdolliseksi. [Tätä korjausta](#) ei kuitenkaan ole vielä otettu osaksi Kohan julkaisuversion koodia

## 4.2.4. Pääkirjaston varasto omana kirjastona

Testausasennuksessa pääkirjaston varasto on omana kirjastonaan. Tarkoituksena on testata, voiko varastotilaukset hoitaa tämän avulla.

Palautuksen kannalta on helpompi, jos varasto on samaa kirjastoa pääkirjaston kanssa. Jos varasto olisi erillinen kirjasto:

- varastokirja jouduttaisiin palauttamaan kahteen kertaan. Ensimmäisen palautuksen (yleiskokoelma automaattilla, Fennicat käsin) jälkeen kirja jäisi Kuljetuksessa-tilaan. Toisessa palautuksessa kirjastoksi vaihdettaisiin Pääkirjaston varasto, jolloin kirja muuttuisi Saatavana-tilaan.

TAI

- joku palautuspisteen kone tai selainikkuna pidettäisiin auki niin, että kirjastona on Pääkirjaston varasto. Kaikki varastokirjat palautettaisiin sen kautta (myös automaattilla palautettu varaston yleiskokoelma).

## 4.2.5. Recall-varaukset

Nykyisessä Kohan julkaisuversiossa ei ole recall-mahdollisuutta. Jyväskylän yliopiston kirjastossa recall-toimintoa käytetään pääasiassa laitoslainassa oleviin Fennica-kokoelman teoksiin. Recallia käytetään myös esim. Kanavuoren etävaraston lainassa olevien gradujen varaamiseen.

Prosessi, jos recallia ei olisi ja asiakas kysyy laitoslainassa olevaa Fennicaa:

- ostetaan teos meille lainakappaleeksi
- jos teos on loppuunmyyty (tai lehti), tilataan Varastokirjastosta. Teettää työtä ja aiheuttaa kuluja (tilauksen teko ja käsittely, postitukset), mutta yleensä toimii nopeasti.
- jos teos ei löydy Varastokirjastosta, tilataan muualta kaukolainana. Voiko asiakas joutua odottamaan pitkään? Entä maksut? Asiakasta tuskin ilahduttaa maksaa kirjasta, joka kuitenkin olisi meidän omissa kokoelmissa.
- entä jos meillä on ainoa kappale, jota ei löydy mistään muualta. Harvinainen tilanne, mutta mahdollinen. Pyydetäänkö asiakasta odottamaan vai otetaanko lainaajaan yhteyttä esim. sähköpostitse tai puhelimitse ja pyydetään palauttamaan kirja? Hidas ja työläs tapa. Kirjan lainaaja voi vitkastella palautuksen kanssa. Laina-ajan lyhentäminen ja mahdollinen sanktio voivat jouduttaa palauttamista, mutta niiden asettaminen käsipelissä hidasta.

Recall-tyyppisten varausten määrä on n. 200 kpl vuodessa, eli toiminnolle olisi käyttöä, mutta toisaalta määrä on sen verran pieni, että on syytä miettiä, pärjätäänkö ilmankin? Lyhennetäänkö henkilökunnalla lainassa olevien Fennicoiden laina-aikaa esim. 28 vrk ja laitetaan ne normaalisti varattaviksi?

#### 4.2.6. Chydenius-instituutin kirjasto

Pitäisikö Chydeniuksen/Kokkolan kirjoilla olla oma nidetyyppi? Jos Kokkolan kirjoilla oma nidetyyppi, voisi asetuksilla estää meidän asiakkaiden sinne tekemät varaukset ja päinvastoin. Tällä hetkellä varaukset tarttuvat myös Kokkolan aineistoihin riippumatta asiakkaan kotikirjastosta.

Mikäli varausten tarttuminen estettäisiin omalla nidetyypillä, opiskelija tai henkilökunnan jäsen, jonka kotikirjasto on Jyväskylässä mutta asioi myös Kokkolassa, ei voisi varata aineistoa omatoimisesti. Henkilökunta voisi kuitenkin ohittaa varauseston.

#### 4.2.7. Kalenteri

Eräpäivien laskentaan vaikuttavaan [kalenteriin](#) ei saa laitettua kirjaston aukiolopäivien kellonaikoja, jolloin eräpäivätiedoissa tai lainauskuiteissa ei näy kirjaston sulkemisaikaa. (=palautusaika) Olisi tarpeen erityisesti lyhytlainoissa. Kalenteriin liittyviä parannuksia on esillä Kohan kehitystikettijärjestelmässä (tiketit [#8133](#) ja [#11211](#), mutta niiden toteutusaikataulusta ei ole tietoa.

#### 4.2.8. Myöhästymismaksut

Voi olla tarpeen miettiä, halutaanko henkilökunnalla pitää erilaiset maksu- ja lainakieltokäytännöt vai yhtenäistää opiskelijoiden ja muiden asiakkaiden kanssa?

### 4.3. Kokoelmatoimisto

Testauksen aikana ei päätetty, tulisiko Koha olemaan myös luetteloinnin työväline vai tapahtuisiko luettelointi ensisijaisesti Melindassa, jolloin tietueisiin liitetään vain nidetiedot.

#### 4.3.1. Sijainti / Niteen luokan pituus

Kohan nidentietojen hyllysijainti ([952\\$o](#)) -kenttä voidaan haluttaessa muodostaa [automaattisesti](#) bibliografisen tietueen kentän, kuten esimerkiksi [082](#) tai [092](#) -kentän tietojen pohjalta. Hyllysijainnin pituudella ei ole rajoitusta.

#### 4.3.2. RDA-luettelointi

Koha [tukee RDA:ta](#) mutta RDA-luettelointia ei testattu lukuun ottamatta [RDA-luettelointipohjan](#) asennusta, joka onnistui ongelmitta.

#### 4.3.3. Tietojen poiminta muista järjestelmistä

Mikäli Kohaa tullaan käyttämään luettelointityökaluna, tietueiden poiminta muista järjestelmistä voidaan tehdä Z39.50-rajapinnan kautta. Myös tietueiden siirto eräajona [selainkäyttöliittymästä](#) on

mahdollista. USEMARCON-tyyppiset muutokset ovat osin mahdollisia suoraan Kohan eräajotyökalussa.

#### 4.3.4. Osakohteet

Tällä hetkellä musiikin osakohteet luetteloidaan Alephilla Melindaan ja monografiaosakohteet Voyagerilla JYKDOKiin. Kohassa on osakohdeluettelointimahdollisuus

- Onnistuuko eräajo ARTOon?
- Jos luettelointi tehdään vain Alephilla, pitää varmistaa monografiaosakohteiden luettelointimahdollisuus. Vai siirrytäänkö luetteloimaan ARTOon? Rajapinta TUTKAan?

## 5. Kohan kehitysnäkymiä

Koha-kehitystyössä keskeisessä asemassa oleva [ByWater Solutions](#) ylläpitää [palvelua](#), jolla kerätään joukkorahoitusperiaatteella resursseja Kohan ominaisuuksien kehittämistä varten. Palveluun voi jättää kehitysehdotuksia, joita Bywater harkintansa mukaan julkaisee sivustolla. Palvelun hyödyntäminen voisi olla harkitsemisen arvoinen vaihtoehto myös suomalaisten Koha-kirjastojen tapauksessa, jos havaitaan, että Kohasta puuttuu ominaisuus, joka voisi olla hyödyllinen laajalti, eikä ominaisuuden kehittämiseen pelkästään kotimaisin resurssin ole mahdollista. Esimerkiksi varastopyyntöjen käsittely voisi olla tällainen yhdessä rahoitettava, laajasti hyödyllinen kehityskohde.

Kohan kehitystyön suuntaa ja uusien ominaisuuksien ideointia on mahdollista seurata kehittäjäyhteisön ylläpitämältä [Request for Comments](#)-wikisivulta. RFC-dokumenttien tarkoituksena on koota yhteen erilaisia näkemyksiä Kohaan halutuista ominaisuuksista, hakea konsensusta ja mahdollistaa kehitystyöhön osallistuminen mahdollisimman varhaisessa vaiheessa. RFC-dokumenttien lisäksi kehitystyön kokonaiskuvan muodostamisessa ovat avuksi kehittäjäyhteisön [verkkokokousmuistiinpanot](#) ja Kohan [tikettijärjestelmä](#).

Keväällä 2015 EBSCO [ilmoitti](#) tukevansa merkittävällä summalla Kohan kehitystyötä. Tuki kanavoidaan [Koha Gruppo Italianon](#) kautta Kohan kehittämisessä mukana oleville ohjelmistoyhtiöille. Tukea on käytetty muun muassa Kohan [sisäisesti käyttämän hakuindeksin uudistustyöhön](#) — uudistetulla haulalla varustettu Koha on tulossa julki jo syksyllä 2015 — ja muita EBSCO:n tuella tehtäviä parannuksia ovat muun muassa:

- Linkitetyn datan tuki
- Rajapinta asiakastoiminnallisuuksiin
- Laajennettu metadataskeemojen tuki; mahdollisuus sisällyttää hakuun myös muuta kuin MARC21-dataa
- Nopeusoptimoinnit
- Laajennettu [Ebscon EDS-palvelun tuki](#)

Näiden kehityskohteiden lisäksi käynnissä tai suunnitteilla on myös muita Jyväskylän yliopiston kirjaston kannalta kiinnostavia kehityskohteita, kuten varastotilausten tuki. Tällä hetkellä Kohasta puuttuu mahdollisuus tehdä ja käsitellä yleisöltä suljettuihin kokoelmiin kohdistuvia

aineistopyyntöjä. Vuonna 2010 julkaistuun Kohan 3.2-versioon on kuitenkin ollut saatavilla ranskalaiselle [BULAC-kirjastolle](#) kehitetty [aineistopyyntö-laajennos](#), joka BULAC-kirjaston on ollut tarkoitus muokata yhteensopivaksi Kohan uudempien versioiden kanssa. Varastotilaustoimintoon ja recall-tyyppisiin varauksiin tarvittavien muutosten kehittäminen tai rahoittaminen voisi olla myös suomalaisille korkeakoulukirjastoille hyödyllinen yhteistyön muoto, mikäli Kohaan siirtyminen tulee ajankohtaiseksi tulevina vuosina.

# Liite A: Asetukset

Seuraavat taulukot sisältävät testiasennuksessamme käytössä olleet asetukset.

## A.1. Kirjastot

Sijainti	sijaintikoodi
Kanavuoren varasto	kanavuori
Kaukopalvelu	ILL
Laitoskirjastot	UNKNOWN
Mattilanniemen kirjasto	mattila
Pääkirjasto	main
Pääkirjaston varasto	stock
Ylistönrinteen kirjasto	ylisto

## A.2. Aineistotyyppit

Aineistotyyppi	Kuvaus
EIKAYTTOA	Aineisto ei käytössä
LAINA0	Ei lainata
LAINA35	Kaukopalvelulainat
LAINA14	Laina-aika 14 vrk
LAINA28	Laina-aika 28 vrk
TABLETIT14	Lainattavat tablettitietokoneet
SP	Säilytyspaikka
VARASTO28L	Varasto, lukusalilaina
VARASTO180	Varastolaina 180 vrk
VARASTO28	Varastolaina 28 vrk
LAINA1	Vuorokauden/viikonlopun lyhytlaina

## A.3. Kokoelmakoodit

Voyagerin kokoelmatunnukset mapattiin Kohan CCODE-sallittuihin arvoihin. Yhteensä kokoelmakoodeja oli 259.

Arvo	Kuvaus	Pitkä kuvaus
Plainaus	.Pääkirjasto. Lainaus	Pääkirjasto. Lainaus 1. krs
0121b	0121bTesti	0121b
121	0121Testi	121
660	Agora Center	Agora Center (Agora). Ei kotilainaan
902	Ambiotica BIO Lainattavat	Bio- ja ymp.tiet. kirjasto (Ambiotica). Laina-aika 28 vrk.
900	Ambiotica BIO LsKirja	Bio- ja ymp.tiet.kirjasto (Ambiotica) Kirjat. Lukusalikäyttö
658	ATK-keskus Lehdet	ATK-keskus Lehdet
700	AVO Lehdet	AVO Lehdet
952	BIO Bio-ja ymp. lehdet	Bio- ja ymp.tiet.kirjasto (Ambiotica). Lehdet. Ei kotil.
5200	P Kurssikirja	Pääkirjasto. Kurssikirjat
116	P KäsikKurssik	Pääkirjasto. Lukusalikurssikirjat. 2.krs. Käsikirjat
111	P Lainattava kurssikirja	Pääkirjasto. Kurssikirjat. 2.krs. Laina-aika 14 vrk.
109	P Lehdet 2. krs	Pääkirjasto. Lehdet. 2.krs. Ei kotilainaan
104	P Lehdet 3. krs	Pääkirjasto. Lehdet. 3.krs. Ei kotilainaan
303	P Lehdet Tupa	Pääkirjasto. Lehdet. Tupa
126	P LsEU-kokoelma. 3. krs	Pääkirjasto. EU-lukusalikokoelma. 3.krs. Ei kotilainaan
130	P LsMultimedia 1. krs	Pääkirjasto. Multimediat. Lainaustoimisto. Ei kotilainaan
115	P Lyhytlainakurssikirjat	Pääkirjasto. Kurssikirjat. 2.krs. Lyhytlaina.

## A.4. Asiakkaiden tilastointiryhmät

Koodi	Kuvaus
G	Ammattikorkeakoulut
BT	Avoimen yliopiston henkilökunta
ET	Avoimen yliopiston opiskelijat
R	Elinkeinoelämä
BE	Erillislaitosten henkilökunta
BH	HTDK henkilökunta
EH	HTDK opiskelijat
BI	ITDK henkilökunta
EI	ITDK opiskelijat
S	Julkishallinto
N	Kotimaiset kirjastot
BS	KTDK henkilökunta
ES	KTDK opiskelijat
BL	LTDK henkilökunta
EL	LTDK opiskelijat
BM	MTDK henkilökunta
EM	MTDK opiskelijat
I	Muiden yliopistojen henkilökunta
J	Muiden yliopistojen opiskelijat
U	Muut asiakkaat
P	Muut opiskelijat
BN	TTDK henkilökunta

## A.5. Asiakasryhmät

Tunnus	Luokkatyyppin nimi	Tyyppi	Voimassaoloaika
HEN	JY:n henkilökunta	Aikuinen	60 kuukautta
STUDENT	JY:n opiskelijat	Aikuinen	60 kuukautta
TEMP	JY:n tilapäinenhenkilökunta	Aikuinen	30 kuukautta
LIBRARY	Kirjastot	Aikuinen	60 kuukautta
OTHER	Muut asiakkaat	Aikuinen	60 kuukautta
TEST	Testiryhmä	Aikuinen	01/01/2025 saakka.
STAFF	Kirjaston henkilökunta	Henkilökunta	31/05/2025 saakka.



## A.6. Lainaussäännöt

Asiakastyppi	Aineistolaji	Lainojasallittu	Laina-aika	Yksikkö	Eräpäivä	Maksun määrä	Item level holds
JY:n henkilökuunta	Laina-aika 14 vrk	Rajoittamaton	14	days	Ei määritely	10	Allow
JY:n henkilökuunta	Laina-aika 28 vrk	Rajoittamaton	28	days	Ei määritely	10	Allow
JY:n opiskelijat	Laina-aika 14 vrk	Rajoittamaton	14	days	Ei määritely	0.3	Allow
JY:n opiskelijat	Laina-aika 28 vrk	Rajoittamaton	28	days	Ei määritely	0.3	Allow
JY:n opiskelijat	Lainattavat tablettitietokoneet	1	14	days	Ei määritely	10	Don't allow
JY:n opiskelijat	Vuorokauden/viikon lopun lyhytlaina	Rajoittamaton	1	days	Ei määritely	10	Don't allow
Kirjaston henkilökuunta	Laina-aika 14 vrk	Rajoittamaton	14	days	Ei määritely	0	Force
Testiryhmä	Ei lainata	Rajoittamaton	0	days	Ei määritely	10	Allow
Kaikki	Ei lainata	Rajoittamaton	1	days	Ei määritely	10	Don't allow
Kaikki	Säilytyspaikka	Rajoittamaton	0	days	pvm 25/09/2016	0	Don't allow

## A.7. Asiakasryhmien määreet

Tunnus	Kuvaus
INST_ID	Yksilöivä tunniste
STAT	Tilastointiryhmä