

Visma Nova SDK2.0
1.11.2019

Visma NovaSDK 2.0

Tämä dokumentti sisältää ohjeita ja esimerkkejä Visma NovaSDK:n käyttöön liittyen. Esimerkit on tehty vb.net kielellä.

Dokumentissa ei suinkaan käydä läpi kaikkia luokkia tai niiden propertyja. Lisää Luokista/propertyista löytyy erillisestä NovaSDK API dokumentaatiosta.

NovaSDK on fyysiseltä nimeltään Visma.Nova.Sdk.dll ja kirjaston version voi katsoa ko. tiedoston product version –kohdasta esim. 2015.11.25.1555

NovaSDK on jaettu eri osa-alueisiin:

- Nova.CustomerMgmtLibrary : Nova asiakaskortistoon liittyvät toiminnot
- Nova.Common : yleisiä Novassa käytettyjä listoja tms.
- Nova.InvoiceMgmtLibrary : Nova laskutukseen liittyvät toiminnot
- Nova.MaterialFlowMgmtLibrary : Novan materiaalihallintaan liittyvät toiminnot
- Nova.PayrollMgmtLibrary : Novan palkanlaskentaan liittyvät toiminnot
- Nova.PricelistMgmtLibrary : Novan hinnastojen käsittely
- Nova.ProductionMgmtLibrary : Novan tuotantoon liittyvät toiminnot
- Nova.PurchaseLedgerMgmtLibrary : Novan ostoreskontraan liittyvät toiminnot
- Nova.PurchaseOrderMgmtLibrary : Novan ostotilauksiin liittyvät toiminnot
- Nova.SalesOrderMgmtLibrary : Novan myyntitilauksiin liittyvät toiminnot
- Nova.WarehouseMgmtLibrary : tuotekoodiin liittyvät toiminnot

Tietokantayhteys

NovaSDK tarvitsee toimiakseen Novan tietokantaa. Tietokantayhteys määritellään SDK:lle context –olion avulla ja lähes jokainen NovaSDK:n funktio tarvitsee context -olion toimiakseen. Näin ollen context -olio kannattaa alustaa pääluokan ylätasolle. Tässä esimerkki context olion alustamisesta:

```
private context As NovaContext = New NovaContext("sqlpalvelin", 1000)
```

Esimerkissä otetaan sqlpalvelimen Novan yritystietokantaan 1000. Yhteys voidaan muodostaa myös SQL connectionstringin avulla tai tietyinä käyttäjänä.

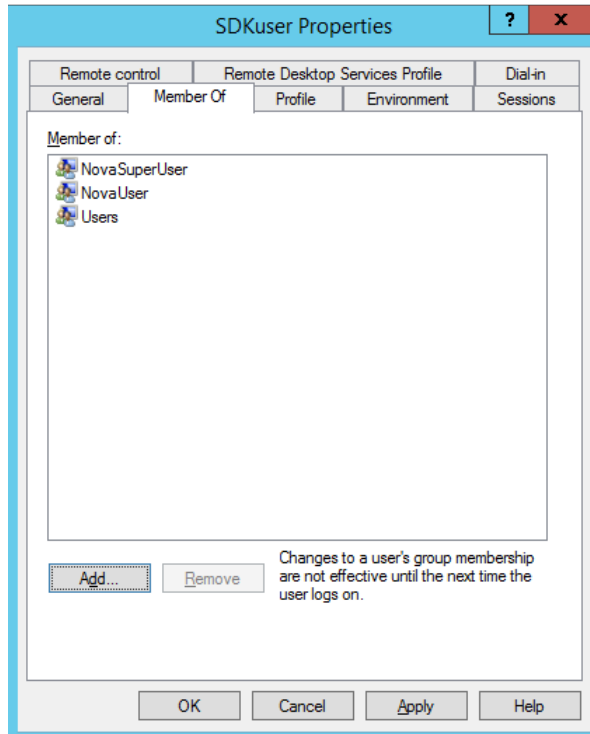
```
Dim context As NovaContext = New  
NovaContext("Server=SQLpalvelin;Database=YR1000;Trusted_Connection=True;")
```

Visma Nova SDK2.0

1.11.2019

Nova SDK:n käyttämä käyttäjätunnus täytyy määrittellä Visma Novan käyttäjäryhmiin (Windows käyttäjäryhmät). Pakollisia ryhmiä ovat NovaSuperUser ja NovaUser. Lisää Novan käyttäjäryhmistä voit lukea Novan käsikirjoista.

Kuvassa näkyy SDKuser –käyttäjälle määritellyt ryhmät:



Visma Nova SDK2.0
1.11.2019

Luokkien rakenteesta

NovaSDK:ssa yleisesti kaikilla luokilla on CreateNew –funktio, jonka avulla luodaan uusi ilmentymä luokasta. Read –funktiolla luetaan haluttu tietue olioon ja vastaavasti Save –funktio tallentaa tiedot kantaan tai suorittaa määritellyt toimenpiteet. Varsinaisen pääluokan aliluokissa uuden tietueen lisääminen tapahtuu yleensä AddNew –funktiolla. Näitä ovat esimerkiksi SalesOrder.rows (myyntilauksen rivit) sekä Customer.contacts (asiakkaan yhteyshenkilöt). Nimiavaruuksista löytyy myös Info -päätteisiä luokkia, joista listat koostuvat. Nämä info –luokat ovat usein paljon suppeampia kuin niitä vastaavat pääluokat. Esimerkeissä ei ole käyty läpi luokkien kaikkia propertyja eikä nimiavaruuksien kaikkia luokkia. Lisää luokista/propertyista löytyy NovaSDK API dokumentista.

Nova.CustomerMgmtLibrary : asiakaskortistoon liittyvät toiminnot

Tässä luvussa käydään läpi SDK:n tarjoamat asiakkaaseen liittyvät toiminnot. Asiakkaalla tarkoitetaan tässä tapauksessa Visma Novan asiakaskortiston asiakas –tietuetta. Uuden asiakkaan perustaminen, lukeminen, päivittäminen löytyvät Customer –luokasta. Toimittajakortiston vastaavat toiminnot löytyvät Supplier –luokasta. Supplier –luokkaa ei tässä erikseen käydä läpi, koska se noudattelee samaa rakennetta Customer –luokan kanssa.

Nova.CustomerMgmtLibrary –nimiavaruuden sisältämät tärkeimmät luokat:

- **Customer** : asiakkaan luonti, luku ja tallennus.
- **CustomerList** : asiakaslista, joka koostuu CustomerInfo –luokista.
- **CustomerInfo** : suppea asiakasluokka (luku)
- **Contacts** : asiakkaan yhteyshenkilölista, joka koostuu Contact luokista.

Esimerkkejä varten pääluokalle on lisätty seuraava nimiavaruus:

```
Imports Nova.CustomerMgmtLibrary
```

Uuden asiakkaan perustaminen Customer –luokan avulla:

```
Public Sub UusiAsiakas()  
    Dim asiakas As CustomerMgmtLibrary.Customer 'esitellään asiakas olio  
    'perustetaan uusi asiakas:  
    asiakas = Customer.CreateNew(context)  
    asiakas.Name1 = "SDK koulutus" 'asiakkaan nimi  
    asiakas.Name2 = "2013" 'asiakkaan lisänimi  
    asiakas.StreetAddress = "Lähiosoite" 'lähiosoite  
    asiakas.City = "65100 VAASA" 'Osoite  
    asiakas = asiakas.Save() 'tallennetaan tietue kantaan.  
End Sub
```

Visma Nova SDK2.0
1.11.2019

Asiakkaan lukeminen Customer.Read –funktion avulla:

```
Public Sub AsiakkaanLuku()  
    'luetaan asiakastiedot olio on asiakasnumerolla 13450  
    Dim cust As CustomerMgmtLibrary.Customer = Customer.Read(context, 13450)  
    cust.YCode = "1234567-8" 'asetetaan asiakkaalle Y-tunnus  
    cust.Save() 'tallennetaan muutokset kantaan.  
End Sub
```

Asiakaslistan haku Customerlist.Read –funktion avulla:

```
Public Sub Asiakaslista()  
    Dim asiakaslista As CustomerList = CustomerList.Read(context) 'haetaan kaikki asiakkaat listaan  
    For Each custInfo As CustomerInfo In asiakaslista  
        Debug.Print(custInfo.Number) 'tulostetaan asiakasnumero  
        Debug.Print(custInfo.Name) 'asiakkaan nimi  
    Next  
  
    'asiakaslista voidaan hakea myös siten, että joissain näistä kentistä: nimi, nimi2, osoite tai kaupunki  
    'löytyy sana VAASA:  
    Dim hakulista As List(Of String) = New List(Of String)()  
    hakulista.Add("VAASA")  
    asiakaslista = CustomerList.GetCustomersWithCriteria(context, hakulista)  
    'ja tarkennetaan vielä että haetaan listasta vain ne jossa kaupunki on VAASA  
    Dim vaasalista = From asiakasinfo As CustomerInfo In asiakaslista Where  
    asiakasinfo.City.Contains("VAASA")  
    'palautetaan esimerkiksi näiden asiakkaiden lukumäärä:  
    Debug.Print(vaasalista.Count)  
End Sub
```

Esimerkki asiakkaan yhteyshenkilön lisäämisestä ja yhteyshenkilöiden listaaminen:

```
Public Sub AsiakkaanYhteyshenkilot()  
    Dim asiakas = Customer.Read(context, 13450)  
    'lisätään asiakkaalle uusi yhteyshenkilö, joka tapahtuu contacts.addnew metodilla:  
    Dim person As Contact = asiakas.Contacts.AddNew  
    person.Name = "Erkki Esimerkki"  
    person.Email = "esimerkki@asiakas.fi"  
    'tallennetaan ns. isäntäolio eli tässä tapauksessa asiakas -olio.  
    asiakas = asiakas.Save()  
  
    'Listataan asiakkaan kaikki yhteyshenkilöt  
    For Each person In asiakas.Contacts  
        Debug.Print(person.Name) 'yhteyshenkilön nimi  
        Debug.Print(person.Title) 'henkilön titteli  
        Debug.Print(person.Email) 'henkilön sposti  
    Next  
End Sub
```

Nova.Common : Yleisiä Novan listoja yms

Nova.Common –nimiavaruus sisältää Novan listoja esim. myyjälistä, maksuehdot yms. tässä oppaassa ei tarkemmin käydä Nova.Common -nimiavaruuteen liittyviä toimintoja, vaan osassa muista esimerkeissä on Nova.Commonin tarjoamia luokkia käytetty. Kysy tarvittaessa lisätietoja.

Visma Nova SDK2.0
1.11.2019

Nova.InvoiceMgmtLibrary : Nova laskutukseen liittyvät toiminnot

Tässä luvussa käydään läpi SDK:n tarjoamat laskutukseen liittyvät toiminnot. Uuden laskun perustaminen, lukeminen ja päivittäminen löytyvät Nova.InvoiceMgmtLibrary.Invoice –luokasta.

Nova.InvoiceMgmtLibrary nimiavaruuden sisältämät tärkeimmät luokat

- **Invoice** : Laskun lukeminen, luonti, muokkaus ja tallennus
- **InvoiceInfo** : Suppea lasku –luokka (luku)
- **InvoiceList** : lista, joka koostuu InvoiceInfo –luokista.
- **InvoiceRow** : Laskun rivi -luokka
- **ManualPaymentTransaction** : Laskun manuaalisuoritus

Esimerkkejä varten pääluokalle on lisätty seuraava nimiavaruus:

```
Imports Nova.InvoiceMgmtLibrary
```

Laskun muodostus Invoice.CreateNew –funktion avulla:

```
Public Sub LaskunMuodostus()  
    'alustetaan lasku olio:  
    Dim lasku As InvoiceMgmtLibrary.Invoice = Invoice.CreateNew(context, 1) 'tehdään lasku laskulajiin 1.  
    lasku.SetCustomer(13450) 'asetetaan laskulle asiakas 13450  
    lasku.SetDeliveryCustomer(13450) 'asetetaan toimitusasiakkaaksi 13450  
    lasku.InvoiceDate = Date.Today 'laskun pvm  
    lasku.TermsOfPaymentId = 1 'maksuehto  
    lasku.Language = "FIN" 'laskun kielikoodi  
    'lisätään laskulle rivi  
    Dim rivi As InvoiceRow = lasku.InvoiceRows.AddNew()  
    rivi.SetProduct("tuote1") 'asetetaan riville tuote: tuote1  
    rivi.Amount = 1 'määräksi 1  
    lasku = lasku.Save() 'tallennetaan lasku kantaan  
    Debug.Print(lasku.Number)  
End Sub
```

Laskun suorittaminen InvoicePaymentServicen avulla:

```
Public Sub LaskunSuorittaminen()  
    Dim lasku As Invoice = Invoice.Read(context, 32902279) 'luetaan lasku numero 32902279 olioon  
    'manuaalisuoritusolio, joka luodaan luetulle laskulle  
    Dim suoritus As ManualPaymentTransaction = New ManualPaymentTransaction(context, lasku.Number)  
    Dim opensum As Decimal = 0  
  
    suoritus.Account = "1900" 'suoritustili  
    suoritus.DifferenceAccount = "1901" 'erotili  
    suoritus.Difference = 0 'ero  
  
    opensum = suoritus.PaymentSum 'avoin summa  
    Debug.Print(opensum)  
    InvoicePaymentService.FilePayment(context, suoritus) 'tehdään suoritus InvoicePaymentServicella  
End Sub
```

Visma Nova SDK2.0

1.11.2019

Laskujen haku listalle hakuehdoilla:

```
Public Sub ListaaLaskut()  
    'kun halutaan listaan tietyt kentät, voidaan tässä tapauksessa käyttää NovaFieldManager -luokkaa  
    Dim fields As New Nova.Common.NovaFieldManager  
  
    'otetaan listaan seuraavat kentät:  
    fields.Add(InvoiceProperties.Number)           'numero  
    fields.Add(InvoiceProperties.CustomerNumber)   'asiakasnumero  
    fields.Add(InvoiceProperties.Type)             'laji  
  
    Dim builder As Nova.Common.QueryBuilder = New Nova.Common.QueryBuilder  
    builder.AddAndCondition(New NovaFieldQueryCondition("TILA=0")) 'otetaan kaikki laskut joiden tila on  
kesken  
    builder.AddAndCondition(New NovaFieldQueryCondition("LAJI=1")) 'lajista 0 ja  
    builder.AddAndCondition(New NovaFieldQueryCondition("ANUMERO=13428")) 'asiakasnumero = 13428  
    'luodaan laskulista, joka käyttää yllä mainittuja ehtoja  
    Dim laskut As InvoiceList = InvoiceList.Read(context, fields, builder)  
  
    'esimerkiksi ensimmäinen löydetty lasku voidaan lukea seuraavasti:  
    Dim laskuinfo As InvoiceInfo = laskut(0)  
    Dim laskunro As Integer = laskuinfo.GetValue(InvoiceProperties.Number)  
  
    Dim lasku As Invoice = InvoiceMgmtLibrary.Invoice.Read(context, laskunro)  
    'ja laskun rivit löytyy Invoicerows -listasta  
    ' lasku.InvoiceRows 'listasta  
End Sub
```

Visma Nova SDK2.0
1.11.2019

Nova.MaterialFlowMgmtLibrary : Materiaalihallintaan liittyvät toiminnot

Tässä luvussa käydään läpi SDK:n tarjoamat materiaalihallintaan liittyvät toiminnot, kuten varastosiirto, varasto-otto, varasto-tulo, tuotantoajo sekä inventointi.

Nova.MaterialFlowMgmtLibrary nimiavaruuden sisältämät tärkeimmät luokat

- **ArticleArrival** : varastotulo
- **ArticleInventory** : Inventointi
- **ArticleWarehouseTransfer** : varastosiirto
- **ArticleWithdraw** : varasto-otto
- **ProductionRun** : tuotantoajo

Esimerkkejä varten pääluokalle on lisätty seuraava nimiavaruus:

```
Imports Nova.MaterialFlowMgmtLibrary
```

Varaston tulokirjaus ArticleArrival -luokan avulla:

```
Public Sub Tulo()  
    'luodaan tulo -olio käyttämällä ArticleArrival.CreateNew -metodia  
    Dim tulo As Nova.MaterialFlowMgmtLibrary.ArticleArrival = ArticleArrival.CreateNew(context)  
    tulo.ArticleCode = "sarjanrotuote" 'tuotekoodi  
    tulo.Amount = 100 'tuodaan 100kpl  
    tulo.Clarification = "selite"  
    tulo.Warehouse = 1 'varastopaikka  
    tulo.TransactionDate = Now  
    For i = 1 To tulo.Amount 'generoidaan tarvittaessa sarjanumerot:  
        tulo.SerialNumbers.Add("A" & 10000 + i)  
    Next  
    tulo.Save() 'tallennetaan tulotapahtuma  
End Sub
```

Tuotteen inventointi ArticleInventory -luokan avulla:

```
Public Sub Inventointi()  
    'luodaan inventointi -olio käyttämällä ArticleInventory.CreateNew -metodia  
    Dim inventointi As ArticleInventory = ArticleInventory.CreateNew(context)  
    inventointi.ArticleCode = "SDK_testituote" 'inventoitava tuotekoodi  
    inventointi.Warehouse = 1 'varastopaikka  
    inventointi.InventoryAmount = 99 'inventoitu määrä  
    inventointi.InventoryDate = Date.Today 'inventointipvm  
    inventointi.Shelf = "A1" 'hyllypaikka  
    inventointi.Save() 'suoritetaan inventointi  
End Sub
```

Visma Nova SDK2.0
1.11.2019

Varastosiirto ArticleWarehouseTransfer –luokan avulla:

```
Public Sub Varastosiirto()  
    Dim varastosiirto As ArticleWarehouseTransfer = ArticleWarehouseTransfer.CreateNew(context)  
    varastosiirto.ArticleCode = "sdk_testituote"  
    varastosiirto.SourceWarehouse = 1 'varastopaikka, josta siirretään  
    varastosiirto.DestinationWarehouse = 2 'varastopaikka, johon siirretään  
    varastosiirto.Amount = 12 'siirrettävä määrä  
    varastosiirto.Clarification = "Varastosiirto (SDK)" 'selite  
    varastosiirto.Save() 'suoritetaan varastosiirto  
End Sub
```

Varasto-otto tehdään tuotekoodille *SDK_testituote* seuraavasti:

```
Public Sub Otto()  
    Dim otto As ArticleWithdraw  
    'varasto-otto tuotteelle SDK_testituote: varastopaikasta 1 ja kpl määrä 100:  
    otto = ArticleWithdraw.CreateNew(context, "SDK_testituote", 1, 100)  
    otto.Clarification = "Otto (SDK)"  
    otto.Save() 'suoritetaan otto  
End Sub
```

Tuotantoajo:

```
Public Sub Tuotantoajo()  
    'luodaan tuotantoajo-olio ProductRun.CreateNew -metodilla  
    Dim tuotantoajo As ProductionRun = MaterialFlowMgmtLibrary.ProductionRun.CreateNew(context)  
    tuotantoajo.OnlyFirstLevel = True 'käsitellään vain tuoterakenteen ensimmäinen osa  
    tuotantoajo.AllLevels = False ' aikaisemmin True 23.5.2013  
    tuotantoajo.Clarification = "sdk tuotantoajo" 'selite  
    tuotantoajo.Warehouse = 2 'varastopaikka  
    tuotantoajo.ArticleCode = "akra" 'tuotekoodi  
    tuotantoajo.Amount = 100 'valmitettu määrä  
    tuotantoajo.Save() 'suoritetaan tuotantoajo  
End Sub
```


Visma Nova SDK2.0
1.11.2019

Nova.PayrollMgmtLibrary : Palkanlaskentaan liittyvät toiminnot

Tässä luvussa käydään läpi SDK:n tarjoamat palkkakauteen ja palkkatapahtumaan liittyvät toiminnot. SDK:n avulla voidaan luoda palkkakausia ja tapahtumia palkkakausille.

Nova.PayrollMgmtLibrary nimiavaruuden sisältämät tärkeimmät luokat:

- **PayrollEvent** : palkkatapahtuma
- **PayrollEvents** : palkkakauden palkkatapahtumat (listaluokka)
- **SalaryType** : palkkalaji -luokka
- **WagePeriod** : Palkkakauden luonti

Esimerkkejä varten pääluokalle on lisätty seuraava nimiavaruus:

```
Imports Nova.PayrollMgmtLibrary
```

Esimerkki palkkakauden luonnista ja sille palkkakausin lisäämisestä:

```
Public Sub UusiPalkkakausiajaTapahtuma()  
    'haetaan henkilö työntekijän numeron avulla  
    Dim henkilo As Common.Employee = Employee.Read(context, 4)  
    'perustetaan uusi palkkakausi työntekijälle  
    'käyttämällä WagePeriod.CreateNew -metodia  
    Dim plkausi As WagePeriod = WagePeriod.CreateNew(context, henkilo.EmployeeCode)  
    plkausi.Txt = "plkauden teksti " & DateTime.Now.ToString 'palkkakauden vapaa teksti  
    'avataan palkkatapahtumat kyseiselle kaudelle:  
    Dim pltapauhtumat As PayrollEvents = plkausi.PayrollEventsList  
    'ensimmäinen tapahtuma:  
    Dim pltapauhtuma As PayrollEvent = pltapauhtumat.First  
  
    Dim p1 As SalaryType = SalaryType.Read(context, 1) 'luetaan palkkalaji 1. tuntipalkka  
    p1.EmployeeNumber = henkilo.EmployeeCode 'annetaan palkkalajille työntekijän numero  
    'asetetaan palkkatapahtuman tiedot:  
    With pltapauhtuma  
        .EmployeeNumber = henkilo.EmployeeCode 'henkilönumero  
        .SalaryTypeNumber = p1.SalaryTypeNumber 'palkkalajin numero  
        .UnitPrice = p1.PriceFormula 'palkkajin antama yksikköhinta  
        .Amount = 160 'määrä  
    End With  
  
    'toinen palkkatapahtuma samalle kaudelle  
    Dim pltapauhtuma2 As PayrollEvent = pltapauhtumat.AddNew  
    Dim p3 As SalaryType = SalaryType.Read(context, 3) 'luetaan palkkalaji 3 p3-olioon  
    p3.EmployeeNumber = henkilo.EmployeeCode 'asetetaan palkkalajille työnumero  
    pltapauhtuma2.EmployeeNumber = henkilo.EmployeeCode  
    pltapauhtuma2.SalaryTypeNumber = p3.SalaryTypeNumber  
    pltapauhtuma2.UnitPrice = 100  
    pltapauhtuma2.Amount = 1  
    plkausi.Save() 'tallennetaan palkkatapahtumat palkkakaudelle  
End Sub
```

Visma Nova SDK2.0
1.11.2019

Nova.PricelistMgmtLibrary : hinnastojen käsittelyyn ja hinnan hakuun liittyvät toiminnot

Tässä luvussa käydään läpi SDK:n tarjoamat hinnastojen ja hinnan hakuun liittyvät toiminnot.

Nova.PricelistMgmtLibrary –nimiavaruuden tärkeimmät luokat:

- **CustomerPriceList** : asiakaskohtainen hinnasto.
- **GeneralPriceList** : yleishinnasto
- **GeneralPriceLists** : yleishinnastot
- **PriceListItem** : hintatietue luokka

Esimerkkejä varten pääluokalle on lisätty seuraava nimiavaruus:

```
Imports Nova.PricelistMgmtLibrary
```

Asiakaskohtaisen hinnaston lukeminen:

```
Public Sub AsiakashinnastonLuku()  
    'Luetaan asiakkaan 13428 asiakaskohtainen hinnasto  
    Dim hinnasto As CustomerPricelist = PricelistMgmtLibrary.CustomerPricelist.Read(context, 13428)  
    For Each hinta As PricelistItem In hinnasto  
        Debug.Print(hinta.ProductCode) 'tuotekoodi  
        Debug.Print(hinta.ContractPrice) 'hinta  
        Debug.Print(hinta.ContractDiscount) 'ale %  
    Next  
End Sub  
  
Public Sub YleishinnastonLuku()  
    'Luetaan yleishinnasto nro 3  
    Dim hinnasto As GeneralPricelist = PricelistMgmtLibrary.GeneralPricelist.Read(context, 3)  
    For Each hinta As PricelistItem In hinnasto  
        Debug.Print(hinta.ProductCode) 'tuotekoodi  
        Debug.Print(hinta.ContractPrice) 'hinta  
        Debug.Print(hinta.ContractDiscount) 'ale %  
    Next  
End Sub
```

Hinnan haku tuotteelle onnistuu Nova.Common.Price –luokan GetPriceInfo –funktiolla, joka palauttaa priceInfo –olion. Tässä esimerkissä haetaan hintaa asiakkaan, tuotekoodin ja määrän mukaan. Näiden lisäksi funktiolle voidaan vielä kertoa, jos hintaa halutaan hakea tietystä hinnastosta ja/tai jos halutaan käyttää hinnan hakemiseen Novan aletaulukoita.

```
Public Sub Hinnanhaku()  
    Dim hinta As PriceInfo  
    'haetaan hinta tuotekoodille, asiakkaan (1234) ja määrän mukaan (10)  
    hinta = Nova.Common.Price.GetPriceInfo(context, "tuotekoodi", 1234, 10)  
    Debug.Print(hinta.Price) 'hinta  
    Debug.Print(hinta.Discount) 'mahd. alennusprosentti  
End Sub
```

Visma Nova SDK2.0
1.11.2019

Nova.ProductionMgmtLibrary : Tuotantoon liittyvät toiminnot

Tässä luvussa käydään läpi SDK:n tarjoamat tuotantoon liittyvät toiminnot kuten työn lukeminen Novan tuotannosta ja työn muokkaaminen.

Nova.ProductionMgmtLibrary nimiavaruuden sisältämät tärkeimmät luokat:

- **Production** : tuotantotyön muokkaus, luku ja tallennus.
- **Workphase** : työvaiheen ottaminen käsittelyyn, aloitus ja lopetus.
- **Workphases** : työvaiheiden haku, lisäys ja muokkaus.

Esimerkkejä varten päälluokalle on lisätty seuraava nimiavaruus:

```
Imports Nova.ProductionMgmtLibrary
```

Tuotantoajo ja työn lukeminen:

```
Public Sub Tuotantoajo2()  
    Dim tyo As Production = Production.Read(context, 886) 'luetaan työn numeron 886 tiedot tuotanto-olioon.  
    If tyo.WorkNumber <> 0 Then 'työnumero on olemassa ja on asetettu tuotantoon (type=2)  
        Dim tuotajo As ProductionRun = MaterialFlowMgmtLibrary.ProductionRun.CreateNew(context, tyo)  
        tuotajo.Amount = tyo.OrderedAmount 'valmistettava määrä  
        tuotajo.Warehouse = tyo.StockId 'varastopaikka  
        tuotajo.Clarification = "Tuotantoajo (SDK)" 'selite  
        tuotajo.OnlyFirstLevel = True  
        tuotajo.RemoveWork = True 'poistetaan tuotannosta mikäli määrä tulee täyteen  
        tuotajo.Save() 'suoritetaan tuotantoajo  
    End If  
End Sub
```

Työvaiheiden lukeminen ja muokkaaminen:

```
Public Sub LuetaanTyovaiheet()  
    Dim tuotanto As Production  
    tuotanto = Production.Read(context, 683) 'luetaan tuotantotyö 683  
    'luetaan tuotantotyön työvaiheet:  
    For Each tyovaihe As Nova.ProductionMgmtLibrary.WorkPhase In tuotanto.WorkPhases  
        Debug.Print(tyovaihe.PhaseNumber) 'vaihenumero  
        Debug.Print(tyovaihe.PhaseDescription) 'vaiheen selite  
    Next  
End Sub  
  
Public Sub LuetaanVaiheetJaMuokataan()  
    Dim tuotanto As Production = Production.Read(context, 690) 'read phases of production work no: 690  
    Dim tyovaiheet As Nova.ProductionMgmtLibrary.WorkPhases = tuotanto.WorkPhases  
  
    For Each vaihe As Nova.ProductionMgmtLibrary.WorkPhase In tyovaiheet  
        If vaihe.RunNumber = 3 Then 'muokataan kolmatta työvaihetta:  
            vaihe.StartDate = DateAdd(DateInterval.Day, 1, vaihe.StartDate) 'muokataan vaiheen aloituspäivä  
        End If  
    Next  
    tuotanto.Save() 'tallennetaan muutokset  
End Sub
```

Visma Nova SDK2.0
1.11.2019

Työvaiheiden käsittely; aloitus ja lopetus:

```
Public Sub TyoVaiheidenKasittely()
    'Luetaan tuotantotyö 691
    Dim tuotantotyo As Production = Production.Read(context, 691)
    For Each p As ProductionMgmtLibrary.WorkPhase In tuotantotyo.WorkPhases 'käydään työvaiheet läpi
        If p.PhaseNumber = 3 Then
            p.AddToHandling() 'asetetaan työvaihe käsittelyyn
            p.StartWorkPhase(4, Now) 'työntekijä nro 4 aloittaa työvaiheen
            p.FinishWorkPhase(4, DateAdd(DateInterval.Day, 1, Now), 1, 0, "SDK lopetusleimaus") 'työntekijä
nro 4 saa työn valmiiksi
        End If
    Next
End Sub
```

Nova.PurchaseOrderMgmtLibrary : Ostotilauksiin liittyvät toiminnot

Tässä luvussa käydään läpi SDK:n tarjoamat Novan ostotilauksiin liittyvät toiminnot, kuten tilauksen lukeminen, muokkaus, luonti ja saavuttaminen.

Nova.PurchaseOrderMgmtLibrary nimiavaruuden sisältämät tärkeimmät luokat:

- **PurchaseOrder** : ostotilauksen luonti, muokkaus ja tallennus
- **PurchaseOrderArrival** : ostotilauksen saavuttaminen (toimittaminen)
- **PurchaseOrderDeliveredRowList** : tilauksen toimitusrivit
- **PurchaseOrderInfo** : suppea luokka (luku)
- **PurchaseOrderList** : lista luokka
- **PurchaseOrderRow** : tilausrivi luokka

Esimerkkejä varten pääluokalle on lisätty seuraava nimiavaruus:

```
Imports Nova.PurchaseOrderMgmtLibrary
```

Tilausrivin saavuttaminen (toimittaminen):

```
Public Sub Erätuotesaapuminen()
    'Luodaan uusi olio käyttämällä PurchaseOrderArrival.CreateNew -funktiota
    Dim poa As PurchaseOrderArrival = PurchaseOrderMgmtLibrary.PurchaseOrderArrival.CreateNew(context)
    poa.OrderNumber = 1290463 'ostotilausnumero
    poa.ArticleCode = "erätuote" 'tuotekoodi
    poa.ArrivalAmount = 100 'saapuvamäärä
    poa.RowRunNumber = 1 'rivin juoksu nro
    poa.SupplierBatchId = "toimerä1_123456" 'toimittajan erän tunnistus
    poa.Save() 'suoritetaan rivi saapuneeksi
End Sub
```

Tilauksen luonti ja usean rivin saavuttaminen (toimittaminen):

Visma Nova SDK2.0

1.11.2019

```

Public Sub ostotilausLuontijaSaavuttaminen()
    Dim ot As PurchaseOrder = PurchaseOrder.CreateNew(context, 3) 'tehdään uusi myyntitilaus lajiin 6.
    Dim i As Long
    'toimittajatiedot
    Dim toimittaja As Supplier = Supplier.Read(context, 50001)
    ot.SupplierNumber = toimittaja.Number
    ot.SupplierName = toimittaja.Name1
    ot.SupplierName2 = toimittaja.Name2
    ot.SupplierStreetAddress = toimittaja.StreetAddress
    ot.SupplierCity = toimittaja.City
    'Toimitustiedot
    ot.DeliveryCustomerNumber = 1 'asettaak
    ot.DeliveryCustomerName = "SDK asiakas"
    'muut tiedot
    ot.Deliverydate = Date.Now 'toimitusaika
    'Lisätään tilaukselle 3 riviö
    For i = 1 To 3
        Dim poRow As PurchaseOrderRow
        poRow = ot.OrderRows.AddNewRow
        'luetaan tuote article luokan avulla:
        Dim art As Nova.WarehouseMgmtLibrary.Article = Article.Read(context, "Tuote" & i.ToString)
        poRow.ArticleCode = art.ArticleCode 'tuotekoodi
        poRow.ArticleName = art.ArticleName 'nimike
        poRow.Amount = 10 * i 'määrä
        poRow.ArrivalDate = DateAdd(DateInterval.Day, 2, DateAndTime.Now)
        poRow.Unit = art.Unit
        poRow.UnitPrice = 99 + i 'A-hinta
    Next
    ot = ot.Save() 'tallennetaan tilaus
    i = ot.Number 'otetaan numero talteen

    'toimitetaan rivit
    Dim saavuttaminen As PurchaseOrderArrival
    Dim lahetenro As String
    'ensimmäinen rivi, ensimmäinen toimitus
    saavuttaminen = PurchaseOrderArrival.CreateNew(context)
    saavuttaminen.OrderNumber = Convert.ToInt32(i) 'tilausnumero
    saavuttaminen.ArticleCode = "tuote1" 'rivin tuotekoodi
    saavuttaminen.RowRunNumber = 1 'rivin juoksu
    saavuttaminen.ArrivalAmount = 5 'toimitettava määrä
    saavuttaminen.ArrivalDate = DateAndTime.Today 'pvm
    saavuttaminen.Save()
    lahetenro = saavuttaminen.CoveringLetterId 'otetaan toimituksen lehetennumero talteen,
    'koska sitä tarvitaan seuraavissa toimituksissa
    'ensimmäinen rivi, toinen toimitus
    saavuttaminen = PurchaseOrderArrival.CreateNew(context)
    saavuttaminen.OrderNumber = Convert.ToInt32(i)
    saavuttaminen.ArticleCode = "tuote1"
    saavuttaminen.RowRunNumber = 1
    saavuttaminen.ArrivalAmount = 5
    saavuttaminen.ArrivalDate = DateAndTime.Today
    'HUOM! asetetaan aiemmin saatu lähetennumero saman saapumiskerran toimituksille.
    saavuttaminen.CoveringLetterId = lahetenro
    saavuttaminen.Save()
    'toinen rivi
    saavuttaminen = PurchaseOrderArrival.CreateNew(context)
    saavuttaminen.OrderNumber = Convert.ToInt32(i)
    saavuttaminen.ArticleCode = "tuote2"
    saavuttaminen.RowRunNumber = 11
    saavuttaminen.ArrivalAmount = 10
    saavuttaminen.ArrivalDate = DateAndTime.Today
    saavuttaminen.CoveringLetterId = lahetenro
    saavuttaminen.Save() 'tallentaa toimitukset
End Sub

```

Nova.SalesOrderMgmtLibrary : Myyntitilausiiin liittyvät toiminnot

Visma Nova SDK2.0

1.11.2019

Tässä luvussa käydään läpi SDK:n tarjoamat Novan myyntitilauksiin liittyvät toiminnot, kuten tilauksen lukeminen, muokkaus, luonti ja saavuttaminen.

Nova.SalesOrderMgmtLibrary nimiavaruuden sisältämät tärkeimmät luokat:

- **SalesOrder** : myyntitilauksen luonti, muokkaus ja tallennus
- **SalerOrderDelivery** : myyntitilauksen toimittaminen
- **SalesOrderInfo** : suppea luokka (luku)
- **SalerOrderList** : lista luokka
- **SalesOrderRow** : tilausrivi luokka

Esimerkkejä varten pääluokalle on lisätty seuraava nimiavaruus:

`Imports Nova.SalesOrderMgmtLibrary`

Tilauksen luominen SalesOrder luokan avulla:

```
Public Sub UusiMyyntitilaus()
    'luodaan uusi tilaus lajiin 6 SalesOrder.CreateNew funktion avulla
    Dim tilaus As SalesOrder = SalesOrder.CreateNew(context, 6)
    'asetetaan tilaukselle laskutusasiakas:
    tilaus.SetCustomer(13450) 'setCustomer funktio täyttää asiakkaan nimi ja osoitetiedot tilaus oliolle
    tilaus.OrderDate = Now 'tilauksen päiväys
    tilaus.DeliveryDate = DateAdd(DateInterval.Day, 14, Now) 'toimitusaika
    'käytetään common.Getweek luokkaa, viikon hakemiseen
    tilaus.DeliveryWeek = (New Nova.Common.GetWeek).WeekAndYear(tilauks.DeliveryDate)
    tilaus.DeliveryNumber = 100 'toimitustiedot
    tilaus.DeliveryName = "Toimitusasiakas"
    tilaus.DeliveryStreetAddress = "Katuosoite"
    tilaus.DeliveryCity = "00000 KAUPUNKI"
    tilaus.OrderLanguage = "FIN" 'tilauksen kielikoodi
    'lisätään kolme riviä tilaukselle SalesOrderRows.Addnew funktion avulla
    For i = 1 To 3
        Dim rivi As SalesOrderRow = tilaus.SalesorderRows.AddNew
        'haetaan tuotetta varastosta
        Dim tuote As Nova.WarehouseMgmtLibrary.Article = Article.Read(context, "Tuote" & i.ToString)
        rivi.SetProduct(tuote.ArticleCode) 'asetetaan tuote riville
        rivi.Position = i.ToString 'positio
        rivi.Amount = 10 * i 'määrä
        rivi.DeliveryStart = DateAdd(DateInterval.Day, 14, DateAndTime.Now) 'toimitusaika
        rivi.UnitPrice = 99 + i
        'jos kyseessä rakennetuote nimikelajilla 4, puretaan se riveille
        If tuote.ArticleType = 4 Then
            Dim rakenne As ArticleStructureList = ArticleStructureList.Read(context, tuote.ArticleCode, 0)
            'käydään tuotteen rakenne läpi
            For Each rakennerivi As ArticleStructureInfo In rakenne
                If rakennerivi.Code <> tuote.ArticleCode Then
                    Dim soRow_sub As SalesOrderRow = tilaus.SalesorderRows.AddNewItem
                    soRow_sub.SetProduct(rakennerivi.Code)
                    soRow_sub.Amount = rakennerivi.Demand * rivi.Amount
                    soRow_sub.UnitPrice = 0
                    soRow_sub.TotalSum = 0
                    soRow_sub.InternalRowCode = 2 'rivikoodi 2 = purettu rakennerivi
                End If
            Next
        End If
        Next
    Next
    tilaus.Save() 'tallennetaan tilaus
End Sub
```

Tilausrivin toimittaminen:

Visma Nova SDK2.0

1.11.2019

```

Public Sub TilausrivinToimitus()
    'tehdään toimitusta tilaukselle 12901515 ja sen riville, jonka juoksu (mtrivi.juoksu) on 21.
    Dim toimitus As SalesOrderDelivery = SalesOrderDelivery.CreateNew(context)
    toimitus.OrderNumber = 12901515 'tilausnumero
    toimitus.ProductCode = "tuoteY" 'koodi
    toimitus.RowRunNumber = 21 'rivin juoksu
    toimitus.DeliveryAmount = 1 'toimitettu määrä
    toimitus.CoveringLetterDate = DateAndTime.Today
    toimitus.Save() 'tallentaa toimitus novaan
    Dim i As Integer = toimitus.CoveringLetterNumber 'otetaan lähete numero talteen

    'ja jos samassa toimituksessa toimitetaan useampi rivi
    toimitus = SalesOrderDelivery.CreateNew(context)
    toimitus.OrderNumber = 12901515 'tilausnumero
    toimitus.ProductCode = "tuoteX" 'koodi
    toimitus.RowRunNumber = 181 'rivin juoksu
    toimitus.DeliveryAmount = 500 'toimitettu määrä
    toimitus.CoveringLetterDate = DateAndTime.Today
    toimitus.CoveringLetterNumber = i 'asetetaan tähän lähetenumero, näin saadaan niputettua toimitukset
novassa samaan läheteeseen
    toimitus.Save() 'tallentaa toimitus novaan
End Sub

```

Tilausen luku ja muokkaus:

```

Public Sub TilauksenLukuJaMuokkaus()
    Dim tilaus As Nova.SalesOrderMgmtLibrary.SalesOrder
    Dim asiakas As CustomerMgmtLibrary.Customer

    tilaus = SalesOrder.Read(context, 42900094) 'luetaan myyntitilaus nro 42900094
    'haetaan asiakkaalta toimitusehto ja päivitetään se tilauselle:
    asiakas = Customer.Read(context, tilaus.CustomerNumber)
    tilaus.TermsOfDeliveryId = asiakas.TermOfDelivery
    tilaus.Save() 'tallennetaan muutokset
End Sub

```

SSCC toimitus:

```

Public Sub Toimittaminen()
    Dim sd As SalesOrderDelivery = SalesOrderDelivery.CreateNew(context)
    sd.OrderNumber = 8432
    sd.ProductCode = "tuote2"
    sd.RowRunNumber = 1
    sd.DeliveryAmount = 20
    sd.CoveringLetterDate = DateAndTime.Today
    sd.SSCC.Code = "0016412345670000014"
    sd.SSCC.Amount = sd.DeliveryAmount
    sd.SSCC.Weight = 111
    sd.SSCC.Volume = 222
    ''SSCC:lle on käytössä viisi info -kenttää ja viisi infopvm kenttää ja niitä käytetään seuraavasti:
    'sd.SSCC.Info(1) = "ABC-123" 'autonumero, joka halutaan tallentaa SSCC_delivery.info1 kenttään
    'sd.SSCC.Info(3) = "Lisäinformaatiota" 'lisäinformaatio, joka halutaan tallentaa SSCC_delivery.info3
kenttään
    'sd.SSCC.InfoDate(1) = Date.Now 'esim. parasta ennen, joka tallennetaan SSCC_delivery.infopvm1 kenttään
    'sd.SSCC.InfoDate(4) = Date.Now 'esim. valmistuspvm, joka halutaan tallentaa SSCC_delivery.infopvm4
kenttään
    sd.Save()
End Sub

```

Toimitetun tai osatoimitetun myyntitilauksen voi SDK:n avulla laskuttaa seuraavasti:

Visma Nova SDK2.0

1.11.2019

```
Public Sub LaskutaTilaus()  
    Dim i As Integer  
    Dim so As SalesOrder = SalesOrder.Read(context, 8276) 'toimitettu tilaus  
    Dim message As String = ""  
  
    i = Nova.SalesOrderMgmtLibrary.SalesOrder.InvoiceSalesorder(context, so, 7, 1, message) 'laskutetaan  
    tilaus. Huom. Tilauksen täytyy olla toimitettu/osatoimittu tilassa.  
    ' Esimerkissä 7 = novan laskulaji, 1=viitteenmuodostustapa, ellei sitä ole novan numerosarjoissa  
    määritelty  
    'InvoiceSalesorder -funktio palauttaa muodostetun laskun numeron  
  
    If i > 0 Then  
        Dim inv As InvoiceMgmtLibrary.Invoice = InvoiceMgmtLibrary.Invoice.Read(context, i)  
        'inv.BankReference 'viitenumero  
        'inv.TotalSum 'laskun summa  
        MsgBox(inv.BankReference)  
    Else  
        MsgBox(message)  
    End If  
End Sub
```


Visma Nova SDK2.0
1.11.2019

Nova.WarehouseMgmtLibrary : Tuotekoodiin liittyvät toiminnot

Tässä luvussa käydään läpi SDK:n tarjoamat tuotekoodiin liittyvät toiminnot, kuten tuotteen perustaminen, tuotteen hakeminen ja tuotteen muokkaaminen.

Nova.WarehouseMgmtLibrary nimiavaruuden sisältämät tärkeimmät luokat:

- **Article** : tuotteen luonti, luku ja tallennus.
- **ArticleList** : tuotelista, joka koostuu ArticleInfo -luokista.
- **ArticleStructureList** : tuotteen tuoterakenne, joka koostuu ArticleStructureInfo – luokista.
- **SerialNumbers** : tuotteen sarjanumerolista.
- **Shelves** : tuotteen hyllypaikalista.
- **Sizes** : tuotteen kokolajitelma.
- **Warehouses** : tuotteen varastopaikat.
- **WorkphaseList** : tuotteen työvaiheet.

Esimerkkejä varten pääluokalle on lisätty seuraava nimiavaruus:

`Imports` Nova.WarehouseMgmtLibrary

Uuden tuotteen perustaminen:

```
Public Sub Uusituote()  
    Dim tuote As Nova.WarehouseMgmtLibrary.Article = Article.CreateNew(context) 'luodaan uusi tuote:  
    tuote.ArticleCode = "tuotekoodi" & Today.ToShortDateString.Replace(".", "") 'tuotekoodi (pakollinen)  
    tuote.ArticleName = "tuotteen nimi" 'tuotteen nimike  
    tuote.ArticleNameAlias = "tuotteenlisänimike" 'lisänimike  
    tuote.Unit = "kpl" 'yksikkö  
    tuote.Price1 = 100 'ovh hinta  
    tuote.SalesAccount = "3000" 'myyntitili  
    tuote.Supplier = "50001" 'toimittajanumero  
    tuote.ProductGroup = 1 'tuoteryhmä  
    tuote.ArticleType = 2 'nimikelaji  
    tuote.Save() 'tallennetaan tuote kantaan.  
End Sub
```

Vaihtoehtoinen tapa perustaa tuote:

```
Public Sub UusiTuote2()  
    Dim tuote As Article = Article.CreateNew(context, "tuotekoodi", "nimike1", "lisänimike")  
    tuote.Save()  
End Sub
```

Tuotteen lukeminen ja päivittäminen:

```
Public Sub HaeTuote()  
    'luetaan tuote article -olioon:  
    Dim tuote As Article = Article.Read(context, "tuote1")  
    MsgBox(tuote.ArticleName) 'palautetaan luetun tuotteen nimike msgboxiin.  
    'kun tuote on luettu olioön, voidaan sitä muokata:  
    tuote.ArticleNameAlias = "uusi lisänimike" 'asetetaan tuotteelle uusi arvo lisänimike kenttään  
    tuote.Save() 'tallennetaan olio, jolloin muutokset siirtyvät tietokantaan.  
End Sub
```

Visma Nova SDK2.0

1.11.2019

Tuotetta voidaan etsiä myös esimerkiksi EAN koodin avulla. Kun tuotteita haetaan jollain propertyllä saadaan palautusarvona ArticleList –tyyppinen listaolio, joka sisältää *haku* –olion kaltaisia tietueita (olioita). Tässä esimerkki:

```
Public Sub HaeTuottetta()
    'esitellään ensin haku olio, joka alustetaan tyhjäksi:
    Dim haku As IArticle = ArticleList.GetEmptyArticleForSearch
    haku.EanCode = "6423111000044" 'haku olion EAN -propertylle annetaan haluttu arvo.

    'listataan ne tuotteet, jotka palauttavat kysyisen ehdon.
    Dim varastolista As ArticleList = ArticleList.FetchArticlesLike(context, haku)
    MsgBox(varastolista(0).ArticleCode) 'palautetaan listan ensimmäisen tietueen (olion) tuotekoodi.
End Sub
```

Toinen esimerkki listaolion käytöstä:

```
Public Sub Tuotelista()
    'luetaan varastosta kaikki tuotteet ArticleList -olioon.
    Dim tuotelista As ArticleList = ArticleList.GetArticles(context)
    For Each tuote As ArticleInfo In tuotelista
        'tässä käydään läpi varaston kaikki tuotteet
    Next
End Sub
```

Esimerkki tuotteen tuoterakenteen hakemisesta:

```
Public Sub TuotteenRakenne()
    'haetaan "rakennetuote" -tuotekoodin vakiotyörakenne eli silloin työnumero on 0.
    'jos halutaan hakea työnumerolle kiinnitetty rakenne syötetään nollan tilalle työnumero.
    Dim tuoterakenne As ArticleStructureList = ArticleStructureList.Read(context, "rakennetuote", 0)
    For Each rakenneinfo In tuoterakenne
        'palautetaan tuotekoodi ja tuoterakenteen taso ja määrä:
        Debug.Print("Koodi: " & rakenneinfo.Code & ", Taso: " & rakenneinfo.Level.ToString & ", määrä: "
        & rakenneinfo.Demand)
    Next
End Sub
```

Esimerkki tuotteen työvaiheiden hakemisesta:

```
Public Sub Työvaiheet()
    'haetaan rakennetuote -tuotekoodin työvaiheet
    Dim tyovaiheet As WorkPhaseList = WorkPhaseList.Read(context, "rakennetuote")
    For Each vaihe As WorkPhaseInfo In tyovaiheet
        Debug.Print("vaihenumero: " & vaihe.PhaseNumber)
        Debug.Print("selite: " & vaihe.Description)
        Debug.Print("kpl: " & vaihe.Pcs)
        Debug.Print("Yksikkö: " & vaihe.Unit)
    Next
End Sub
```

Tuotteen saavutaminen (jossa yksikkökäsittely):

Visma Nova SDK2.0

1.11.2019

```
Public Sub ArrivalWithUnits()
    'luetaan ensin tuote
    Dim tuote As Nova.WarehouseMgmtLibrary.Article = WarehouseMgmtLibrary.Article.Read(context, "maali")
    'tällä voidaan listata tuotteen mahdolliset yksiköt:
    Dim yksikot As ProductUnitList = ProductUnitList.GetProductsUnits(context, tuote.ArticleCode)
    MsgBox(yksikot(0).Unit & "," & yksikot(1).Unit & "," & yksikot(2).Unit)

    'tehdään tulo maali yksikön PRK5 mukaisesti:
    'ensin haetaan PRK5 -yksikköä vastaava määrä
    Dim pu As ProductUnit
    Dim apu As Decimal
    pu = ProductUnit.GetUnit(context, "PRK5", tuote.ArticleCode)
    apu = pu.Qty    'tässä kerrotaan montako varastoyksikköä PRK5 yksikkö vastaa.

    'tehdään tulo viidelle viiden litran purkille:
    Dim tulo As ArticleArrival = ArticleArrival.CreateNew(context)
    With tulo
        .ArticleCode = tuote.ArticleCode
        .Warehouse = 1
        .Amount = 5 * apu
        .AmountPrice = 5.75
        .CurrencyID = "EUR"
        .SupplierNumber = 50001
        .UpdateAveragePrice = False
    End With
    tulo = tulo.Save()
End Sub
```

Visma Nova SDK2.0

1.11.2019

Nova.BookkeepingMgmtLibrary : Kirjanpitoon liittyvät toiminnot

Nova.BookkeepingMgmtLibrary nimiavaruuden sisältämät tärkeimmät luokat:

- **Voucher** : tosittien luonti, muokkaus ja tallennus
- **VoucherInfo** : suppea luokka (luku)
- **VoucherList** : lista luokka
- **VoucherRow** : tilausrivi luokka

Esimerkkejä tositte hausta.

Hae kaikki tositteet:

```
Public Sub BookkeepingListReadAll()
    Dim voucherList As BookkeepingMgmtLibrary.VoucherList =
        BookkeepingMgmtLibrary.VoucherList.Read(Context)

    Dim vouchersInLockedState As New List(Of BookkeepingMgmtLibrary.Voucher)
    ' Loop through the voucher info list, get for example all vouchers in normal state.
    For Each item As BookkeepingMgmtLibrary.VoucherInfo In voucherList
        If item.State = BookkeepingMgmtLibrary.Voucher.VoucherStates.normal Then
            vouchersInLockedState.Add(BookkeepingMgmtLibrary.Voucher.Read(Context, item.Number))
        End If
    Next
    ' Do something with the vouchers.
End Sub
```

Hae tositteet kriteerillä:

```
Public Sub BookkeepingListReadwithCriteria()
    ' With querybuilder you can filter the voucher infos based on Nova database fields.

    ' For example get info for all vouchers
    ' with date between 2019-01-01 and 2019-01-31
    Dim builder As Nova.Common.QueryBuilder = New Nova.Common.QueryBuilder
    builder.AddAndCondition(New NovaFieldQueryCondition("PVM>='2019-01-01'"))
    builder.AddAndCondition(New NovaFieldQueryCondition("PVM<='2019-01-31'"))

    Dim voucherList As BookkeepingMgmtLibrary.VoucherList =
        BookkeepingMgmtLibrary.VoucherList.Read(Context, builder)

    ' Another example how to get info for all vouchers
    ' in class(tlaji) = 2 and date from 2019-05-01
    Dim builder2 As Nova.Common.QueryBuilder = New Nova.Common.QueryBuilder
    builder2.AddAndCondition(New NovaFieldQueryCondition("tlaji=2"))
    builder2.AddAndCondition(New NovaFieldQueryCondition("PVM>='2019-05-01'"))

    Dim voucherList2 As BookkeepingMgmtLibrary.VoucherList =
        BookkeepingMgmtLibrary.VoucherList.Read(Context, builder2)

    ' Now you got a list of voucher infos,
    ' you can get the vouchers to do something with them...
End Sub
```

Visma Nova SDK2.0

1.11.2019

Esimerkkejä luonnista ja muokkauksesta.

Luonti:

```
Public Sub BookkeepingVoucherCreateAndAddOneRow()  
    Dim voucherClass As Integer = 2  
    Dim currency As String = "EUR"  
  
    Dim voucher As BookkeepingMgmtLibrary.Voucher =  
        BookkeepingMgmtLibrary.Voucher.CreateNew(Context, voucherClass, currency)  
  
    With voucher  
        .Account = "3000"  
        .CostCenter = "CostCenterID"  
        .Currency = "EUR"  
        .Description = "SDK voucher"  
        .ExchangeRate = 1  
        .Id = "Tunniste"  
        .Note = "Note"  
        .State = BookkeepingMgmtLibrary.Voucher.VoucherStates.normal  
        .Total = 12500  
        .Type = 1  
        .ProjectId = "ProjectId"  
        .VoucherDate = Date.Now.Date  
    End With  
  
    Dim voucherRow As BookkeepingMgmtLibrary.VoucherRow = voucher.VoucherRows.AddNew()  
    With voucherRow  
        .SetBookKeepingAccount = "3000" ' Sets account, accountname and taxbase.  
        .CostCenter = "CostCenterID"  
        .Debit = 00  
        .Credit = 10000  
        .Description = "Description"  
        .ProjectId = "ProjectId"  
    End With  
  
    voucher = voucher.Save()  
End Sub
```

Muokkaus:

```
Public Sub BookkeepingVoucherReadAndAddOneRow()  
    Dim voucherClass As Integer = 2  
    Dim currency As String = "EUR"  
  
    Dim voucher As BookkeepingMgmtLibrary.Voucher =  
        BookkeepingMgmtLibrary.Voucher.Read(Context, 4791)  
  
    Dim voucherRow As BookkeepingMgmtLibrary.VoucherRow = voucher.VoucherRows.AddNew()  
    With voucherRow  
        .SetBookKeepingAccount = "3000" ' Sets account, accountname and taxbase.  
        .CostCenter = "CostCenterID"  
        .Debit = 10000  
        .Credit = 00  
        .Description = "Description 2"  
        .ProjectId = "ProjectId 2"  
    End With  
  
    voucher = voucher.Save()  
End Sub
```