

NAV Customer Center

Version: 1.0
2011.04.15
English

LEGAL INFORMATION

© Copyright 2011 Dynamicweb Software A/S.

All rights reserved. Alteration or reproduction of this document or parts hereof is strictly prohibited, regardless of form or means, unless explicit permission has been acquired from Dynamicweb Software.

Dynamicweb is a registered trademark of Dynamicweb Software.

Company and product names mentioned in this document may be registered trademarks or trademarks of third parties.

CONTENTS

1	NAV Customer Center	5
1.1	Customer Center	5
	• 1.1.1 Settings.....	5
1.2	Dynamicweb.....	6
	• 1.2.1 Default template files	6
	• 1.2.2 Settings.....	6
	• 1.2.3 Application – assemblies	6
	• 1.2.4 Application – Custom module backend	6
	• 1.2.5 Dynamicweb Developer.....	6
1.3	Paragraph module.....	7
	• 1.3.1 Templates.....	10
2	NORRIQ.XML	11
3	NAV Customer Center - Developer	13

1 NAV CUSTOMER CENTER

NAV Customer center is live data from NAV.

The architecture in NAV Customer Center is based on the NAV Integration Module, and employs the use of NORRIQ NAV Manager Service and NAV Application Servers.

It requires that this is setup correctly.

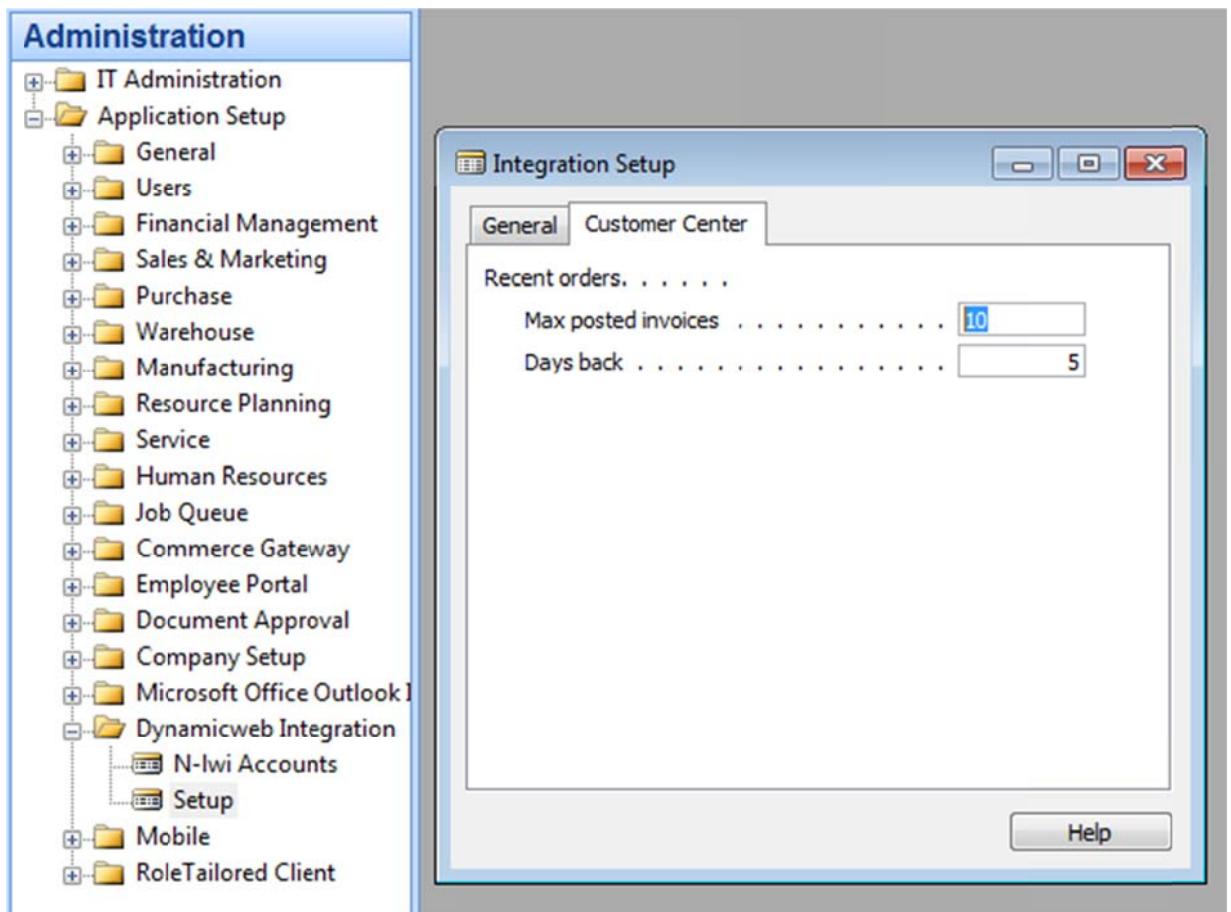
1.1 Customer Center

NAV Integration needs to be installed and setup previous to this.

1.1.1 Settings

In the “Administration – Application Setup – Dynamicweb Intergration – Setup”.

Under Customer Center, you can choose to setup how many max posted invoices the function “Recent orders” should include and how many days back the search should include.



For standard self service there are two different built-in functions that comes with Customer Center.

- Recent orders, that includes sales orders, orders in progress and posted invoices according to settings. See above.

- History Search, that searches in posted invoices by 4 different criteria.
 - Document No.
 - Item No.
 - From Date
 - To Date

These functions are available to NAV in a separate FOB package from NORRIQ that needs to be in-installed in the NLWI enabled NAV.

1.2 Dynamicweb

In Dynamicweb there is a Custom Content Module that needs to be registered in the Dynamicweb Developer. If this is not available to you, Dynamicweb needs to install this for you.

First however you need to deploy the files needed for NAV Customer Center.

The following files are included, and needs to be added to your Dynamicweb solution according to the following paths.

1.2.1 Default template files

```
\Files\Templates\NORRIQ_Service\openordersResult.xslt
\Files\Templates\NORRIQ_Service\historyResult.xslt
\Files\Templates\NORRIQ_Service\reorder.xslt
\Files\Templates\NORRIQ_Service\historySearch-REO.html
\Files\Templates\NORRIQ_Service\showorder.xslt
\Files\Templates\NORRIQ_Service\historySearch.html
```

1.2.2 Settings

```
\Files\NORRIQ.config
```

1.2.3 Application – assemblies

```
\Application\Bin\NORRIQ.Web.Admin.Service.dll
\Application\Bin\NORRIQ.Service.dll
```

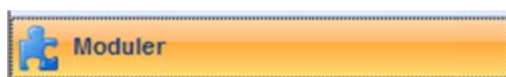
1.2.4 Application – Custom module backend

```
\Application\CustomModules\NORRIQ_Service\NORRIQ_Service_Edit.aspx
```

1.2.5 Dynamicweb Developer

Next step in enabling the solution for NAV Customer Center, is to activate / setup a Custom Module for this can be done by you self if you have enabled Dynamicweb Developer module on the solution.

Go to the Modules section of your Dynamicweb Administration.



If you have enabled Dynamicweb Developer you will find it in the list as below.



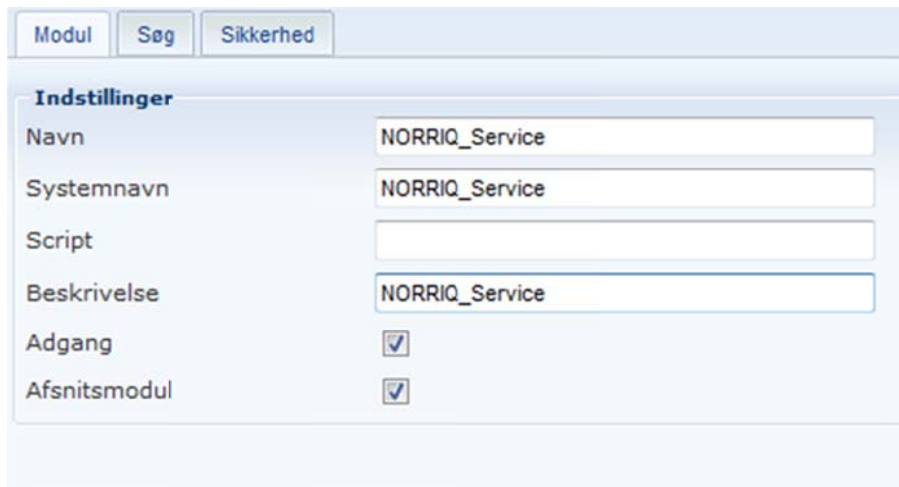
In this page you should choose “New module”.



Navn	Systemnavn	Slet
------	------------	------

Kodegenerering Nyt modul Luk

Next you have to enter the details for the custom module, enter, “NORRIQ_Service” in the Name, Systemname and Description text fields. Set “Access” and “Paragraph module” to yes.



Modul Søg Sikkerhed

Indstillinger

Navn: NORRIQ_Service

Systemnavn: NORRIQ_Service

Script:

Beskrivelse: NORRIQ_Service

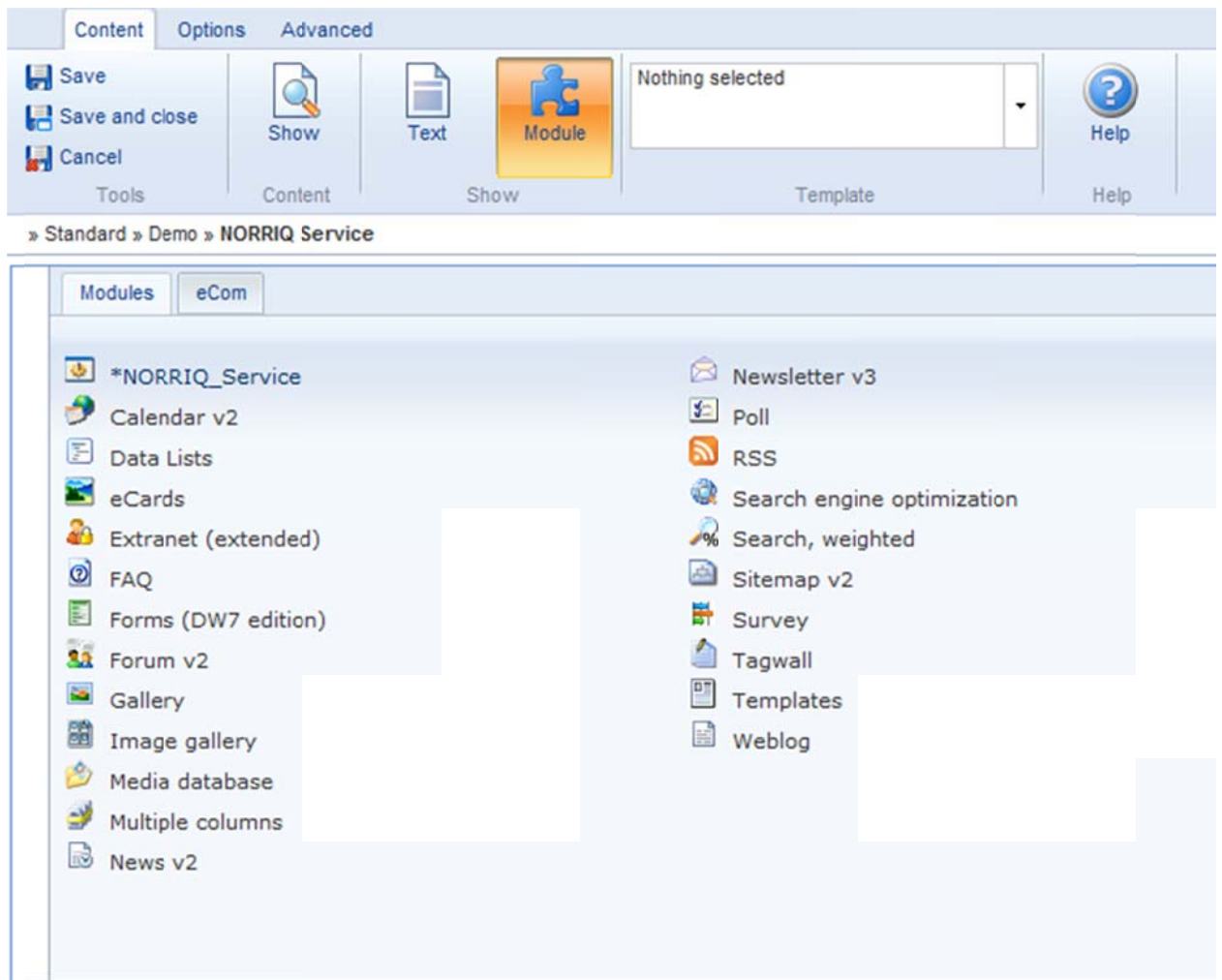
Adgang:

Afsnitsmodul:

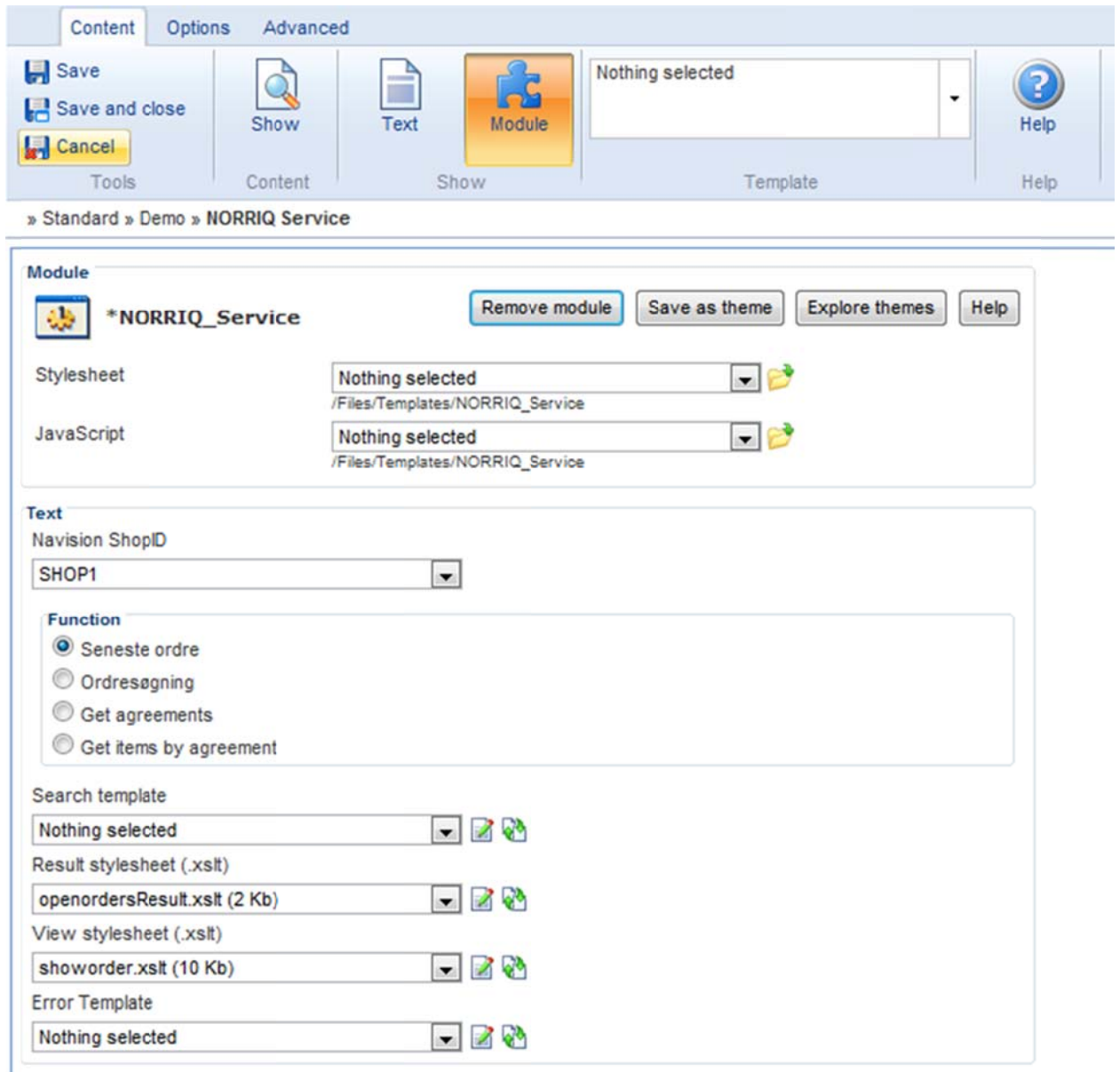
Save these settings. You should now be able to use this module as a paragraph module.

1.3 Paragraph module

To use the module on a paragraph, go to a Page and insert a new paragraph, give this a name, and you will have a dialog as below. Choose the “NORRIQ_Service” module.



If you have set things up right and the files have been deployed, you will get a settings page for the paragraph module.



In this module there are some things to make notice of.

Navision ShopID

You should choose the shop that you have this page setup to correspond to, this is if you have different setups of shops/companies for your customers. You need to setup the module so that users from each company are allowed to use the selfservice pages.

Function

Choose the function available, that you will retrieve data from Navision through. Each function is setup in a settings file, and corresponds to a function created / developed in NAV.

There are only in default setup two functions, "Recent orders", and "History search". Others are created and available by custom setup in the settings file, and development in NAV.

1.3.1 Templates

Search result

This is a template to be displayed as “step” 1 of a selfservice function. Choose this if you have a form to show before getting data from NAV. This is applicable in the “History search” where the users should type in some search criteria before searching NAV.

Included is a “HistorySearch.html”. This is a standard typed Dynamicweb HTML template and includes Dynamicweb Template tags to be used accordingly. Use <!--@DwTemplateTags--> to get an example of these.

For “Recent orders” this is not applicable, as there are no criteria for a recent order list, as this is setup in NAV.

Result stylesheet (.xslt)

This is a template for viewing the resultset from NAV. As results are usually XML from NAV, this is an XSLT template, the data from NAV is then transformed into the viewable HTML by applying this template.

You can choose to change and manipulate data through XSLT.

View stylesheet (.xslt)

When the above resultset lists the headers/rows there are usually a drilldown further to view the actual document. When this is done through default functions this is used.

This is again an XSLT template, and you should dump the complete XML to view what data is available in which structure.

Error template

If for some reason the communication fails or NAV has some kind of error, this will be rendered, so to handle and write out some kind of custom error message to the user. In this template there are tags available for what kind of error happened, by an error number and a message usually generated from NAV.

2 NORRIQ.XML

NORRIQ.XML is an essential file that contains definitions of the functions available from NAV. This file needs to be managed by using some sort of xml editor, usually notepad.

When creating new custom functions in NAV, you need to make sure that both parts have the same definition.

This means that naming and parameters must be setup before you can use a function from Dynamicweb.

Default file

```
<?xml version="1.0" encoding="utf-8" ?>
<data>
  <selfservice>
    <function id="RecentOrders" default="true"
description="Seneste ordre" />
    <function id="HistorySearch" default="true"
description="Ordresøgning" >
...
  </function>
</selfservice>
<developer>
</developer>
</data>
```

In the selfservice there are the built in functions, looks in these as inspiration, the functions created must have the same schema.

There are two options a "selfservice" and "developer", in custom developing you have to put your functions in the "developer", and this will then be handled in NAV in the corresponding Custom Functions handler codeunit.

To add some new functions you create a node named function with an unique id that matches the name that will be handled in NAV.

If there are parameters these also need to be defined.

Ex to make a function that takes two parameters we create an xml node like this:

```
<developer>
  <function id="GetAgreementsByUser" default="false"
description="Get agreements by user name" >
    <param id="agreementtype" datatype="System.String" />
    <param id="username" datatype="System.String" />
  </function>
</developer>
```

As this will be send through XML, all parameters are typically strings, and thus no validation for these, if you need validation you should validate on client side and in NAV.

The above assumes there is a function named "GetAgreementsByUser" in NAV handler, it takes two parameters, agreementtype and username. If the user is logged on with a user that has a customer number this will also be send through to NAV by default handling.

Function handler in NAV could look like this:

```

OK := TRUE;
CASE UPPERCASE(FunctionName) OF
  'GETAGREEMENTBYUSER'      : GetAgreements(XMLDoc, ParamsXML);
ELSE
  OK := FALSE;
END;

```

```

GetAgreements(VAR XMLDoc : Automation '''Microsoft XML, v3.0'.DOMDocument30'';Params :
CustomerNo := Util.GetAttributeValue(Params, 'customerno', 0);
AgreementType := Util.GetAttributeValue(Params, 'agreementtype', 0);
UserName := Util.GetAttributeValue(Params, 'username', 0);
//CREATE RETURN XML
Util.CreateDoc(XMLDoc, XMLStartNode, '', '');
Util.AddNode(XMLDoc, XMLStartNode, XMLDataNode, 'agreements');
// SET SOME FILTERS
IF Agreements.FIND('-') THEN REPEAT
  Util.AddNode(XMLDoc, XMLDataNode, XMLRowNode, 'agreement');
  Util.AddAttribute(XMLDoc, XMLRowNode, 'no', Util.FormatInteger(Agreements."No.");
  Util.AddAttribute(XMLDoc, XMLRowNode, 'status', Agreements.Status);
  Util.AddAttribute(XMLDoc, XMLRowNode, 'date', Util.FormatDate(Agreements.Date));
Until Agreements.Next = 0;
//RETURN STATUS 0 = NO ERRORS
Util.AddStatusNode(XMLDoc, XMLStartNode, 0, 0, '');

```

3 NAV CUSTOMER CENTER - DEVELOPER

To use your own code to get into NAV is also possible through the integration above. The same applies to this as the Service module, you first have to setup the functions in NAV and in the NORRIQ.XML file.

So all you need to do is reference the **NORRIQ.Common.dll**, **NLWI.Com.dll**, **NAVConnector.dll** into your project and if you need any other assemblies from Dynamicweb.

Here is an example of code in C# that uses the above declared method.

```
using Dynamicweb.Frontend;
using System.Collections;

namespace NORRIQ
{
    public class GetAgreementsByUser
    {
        public static string CallNAV(string shopID, string
userName,
        string agreementType, string customerNo)
        {
            //THE PARAMETERS ARE ADDED TO THE FUNCTION BY A
HASHTABLE.
            Hashtable parameters = new Hashtable();
            // MUST MATCH PARAMETER NAMES IN NORRIQ.XML
            parameters.Add("agreementtype", agreementType);
            parameters.Add("userName", userName);
            NORRIQ.Service.CustomFunction navFunction =
                new NORRIQ.Service.CustomFunction();
            int statusNo;
            string xmlResult =
navFunction.CallCustomFunction(shopID,
                customerNo, "GetAgreementsByUser", parameters, out
statusNo);
            // STATUS IS 0 WHEN THERE ARE NO ERRORS
            if (statusNo == 0)
            {
                return xmlResult;
            }
            else
            {
                return string.Format("ERROR OCCURRED");
            }
        }
    }
}
```