Simplified UX

White Paper

Brian Nielsen
Principal Program Manager
Nikola Kukrika
Software Engineer
Jacob Winther
Senior UX Designer

July 2014

Contents

Simplified UX – Design Insights

This whitepaper provides detailed insight into the concepts and principles that support the application simplifications implemented in standard NAV as well as the Simplified UX package that is shipped with Microsoft Dynamics NAV 2015.

The document starts by providing conceptual overviews of the Simplified UX features, and of the underlying principles and themes in the simplification work.

Then it explains the architecture of selected features to show how the various simplifications have been implemented and to show their limitiations.

Finally, the document provides code examples to prepare you for using the simplifications to build or update your own solutions.

Microsoft Dynamics NAV

Introduction to Simplified UX

This section introduces the application simplifications implemented in Microsoft Dynamics NAV 2015, and explains the overall goal for the Simplified UX objects package.

Smaller businesses need simpler accounting software than bigger enterprises, both because their operations are simpler with fewer users, and their users are often not as trained in software as enterprise employees are.

Smaller businesses often do not have dedicated staff for bookkeeping and accounting. Instead, the business owner manages the books. For such simple uses, the standard Microsoft Dynamics NAV product sometimes cause challenges for small business accounting, as it doesn't need all the functionality built to give flexibility in setting up the system for the great variety of larger businesses operations.

The goals for the smaller business are to quickly create a sales invoice for a customer, keep track of account receivables, and get an overview of their business' health. They are likely to outsource bookkeeping and payroll, and they generally do not track their stock beyond rough stock levels.

To help them achieve these goals, Dynamics NAV 2015 comes with a set of features and improvements that support the core small business scenarios. Specifically, we introduce a set of new pages and related functionality that make the user experience much simpler for users that only need Microsoft Dynamics NAV for the basic sales and purchase scenarios of a smaller company.

For these simple scenarios, we have created pages that are simpler, have fewer fields and actions, and are easier to work with. For example, the updated Small Business Role Center contains only the information that most owners of small businesses want to get from their bookkeeping system. This document provides the design insight to the principles and the code behind these features.

Importantly, there are no new *business data tables* added to achieve the simplifications of Microsoft Dynamics NAV. This means that the new pages run off the same business data—and is using the same business processes—that the standard version does. However, a number of new tables have been added to track entity templates, chart definitions, and other types of setup.

The following section explains in more detail the architecture of the simplifications of Microsoft Dynamics NAV.

Architecture of Simplified UX

The simplification of Microsoft Dynamics NAV has happened in two streams of work:
1. Simplification of *existing* pages
2. Creation of a *new* simplified pages for small business accounting

While the two streams of work share a lot of the design and implementation patterns, it is important to understand the differences in their scope of impact on the users' experience with the product.

Existing Pages Simplified

The simplification of *existing pages* will be visible to all users of Microsoft Dynamics NAV, regardless of which profile or role center they are using. However, being a relatively small subset of all NAV's existing pages that have been simplified (primarily sales and purchase documents, master data entities, etc.), there will be users who don't notice the changes.

When simplifying existing pages, it is important to take into account that several *different* user profiles may be using any given page. So even if a set of fields seem unnecessary to one profile, or it would be beneficial to group some fields with others for another profile, this may not be possible while keeping the same page useful to multiple profiles having conflicting needs.

The profile configuration system in Microsoft Dynamics NAV comes to the rescue in a lot of these cases since it provides for different configurations of the same page. Still, there are limitations to what can be achieved via configuration. It is for example currently not possible to change fields between groups via configuration only.

New Simplified Pages

The creation of *new* simplified pages impact a smaller user audience. The new pages will only be experienced by users who are assigned the updated Small Business profile and the associated role center. This is because only the Small Business role center references the different lists that in turn open up the new simplified cards and documents.

However, compared to the simplification of existing pages, the creation of new pages for a specific user audience allows much more freedom to create simplified versions of their counterparts in the standard version of Microsoft Dynamics NAV. It is, for example, straightforward to do a new grouping of fields in a page, place often used parts directly in the page, change captions of lists, etc.

Keep in mind, though, that a *new* set of pages incur costs in training and maintenance, and that the more specific a page is targeting a certain profile only, the fewer users will be able to use it.

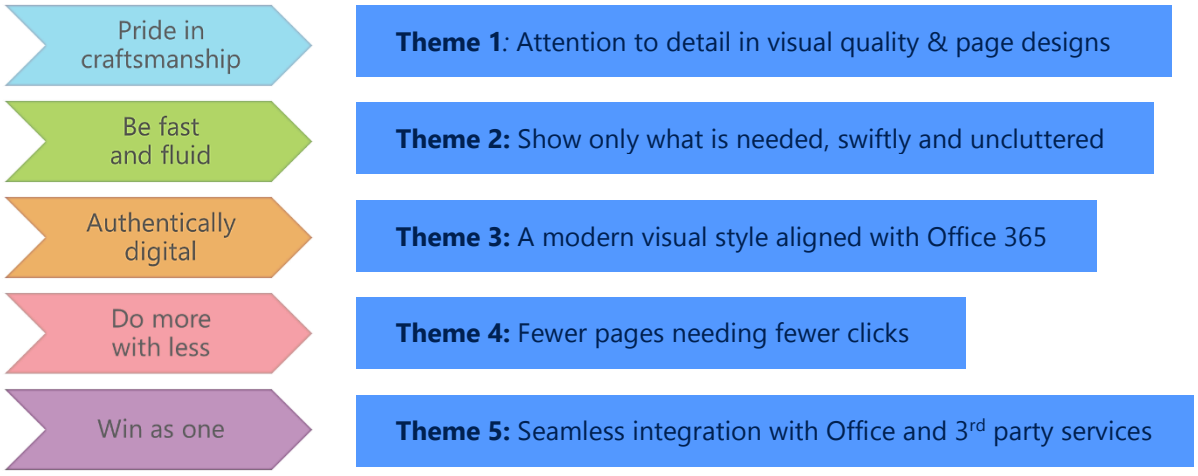
Microsoft Design Principles

The Microsoft Design Principles are the starting point for every product, web site, and even advertising coming from Microsoft. The five principles are: *Pride in craftsmanship, fast and fluid, authentically digital, do more with less,* and *win as one*.

On this common foundation, the Simplified UX for Microsoft Dynamics NAV focuses specifically on the work tasks that business users have. The goal is to provide simple, efficient, and flexible solutions in a well-designed and visually attractive product having seamless integration to other products and services used in the business.

First and foremost, designing simple solutions for end users, in businesses and elsewhere, means reducing the complexity and amount of information that users are faced with. It means designing with people's tasks and pain points in mind, and giving these aspects more priority than the database layout.

But there is more to it. To design a successful business solution, the Microsoft Design Principles bring attention to all the aspects that make up the customer's experience of a product or service. In the design and implementation of Simplified UX for Microsoft Dynamics NAV, the five design principles led us to devise the central themes by which we defined and prioritized ways to give users a simpler experience with the product.



The following sections go into detail with the specific design decisions that were made in connection with Simplified UX for Microsoft Dynamics NAV. Since every simplification typically stretches several of the themes, the presentation of each will not describe the theme or design principle, but focus on the actual design and how it is implemented.

Microsoft Dynamics NAV

Feature Walkthrough

The following sections each present the various features and capabilities that have been implemented to create Simplified UX.

The feature walkthrough will go through:

- Role Center – Setup
- Role Center – KPIs
    - KPI Charts (the carousel)
    - Mini Trial Balance
- Instructions in the UI
    - Welcome Video
    - Getting Started Actions
    - Dismissable Dialogs
    - Task-oriented Page Help
    - Tooltips.
- Auto-fill the No. Field
- Create Customers and Vendor on the Fly
- Create Master Data Records from Templates
- Mandatory Fields
- Item as a Service
- Totals on Sales and Purchase Documents
- Toggle Finance Setup on Master Data Cards
- Prices and Line Discounts on Master Data Cards
- KPIs on Master Data Cards

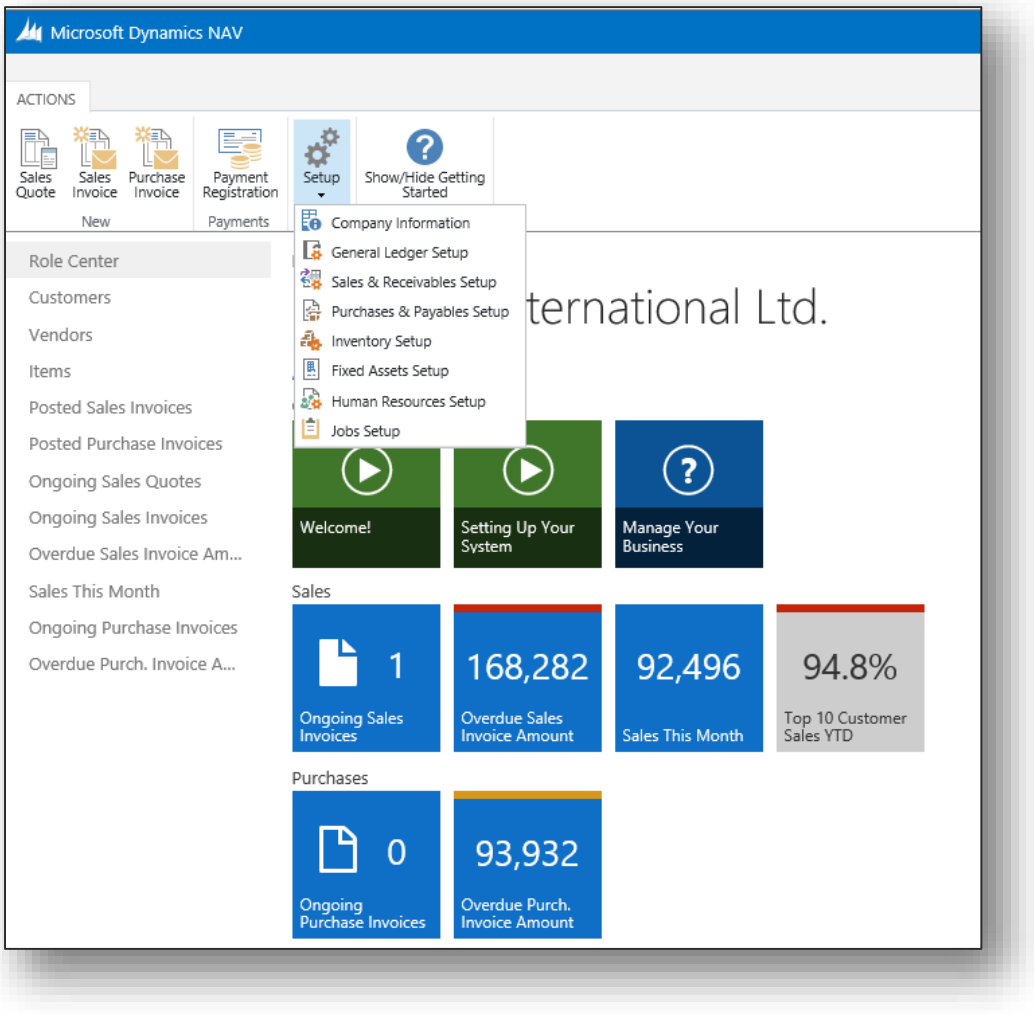
Each section will present the feature, outline the design idea behind it, and then explain in more detail how it was implemented as well as discuss the known limitations and issues related to the feature or capability.

Role Center – Setup

General Principle

In the hosted solution, partners are expected to provide basic setup out of the box. The users may want to do small configurations so the system fits their needs or to update basic setup data, such as company logo, web site, and bank account number.

For small-business users, it is difficult to navigate in the product and use the normal setup pages. We have therefore provided these users with the ability to do all setup from the Role Center by using simplified pages.



The design is based on a pattern where the user is guided through setup on first use. When the user tries

Microsoft Dynamics NAV

to access functionality that is not set up, the system will ask the user to make the setup. This pattern is used, for example, in the OIOUBL functionality (in the Danish version) and in the Payment Reconciliation Journal feature within Cash Management. You can find more information about the pattern here.

Implementation

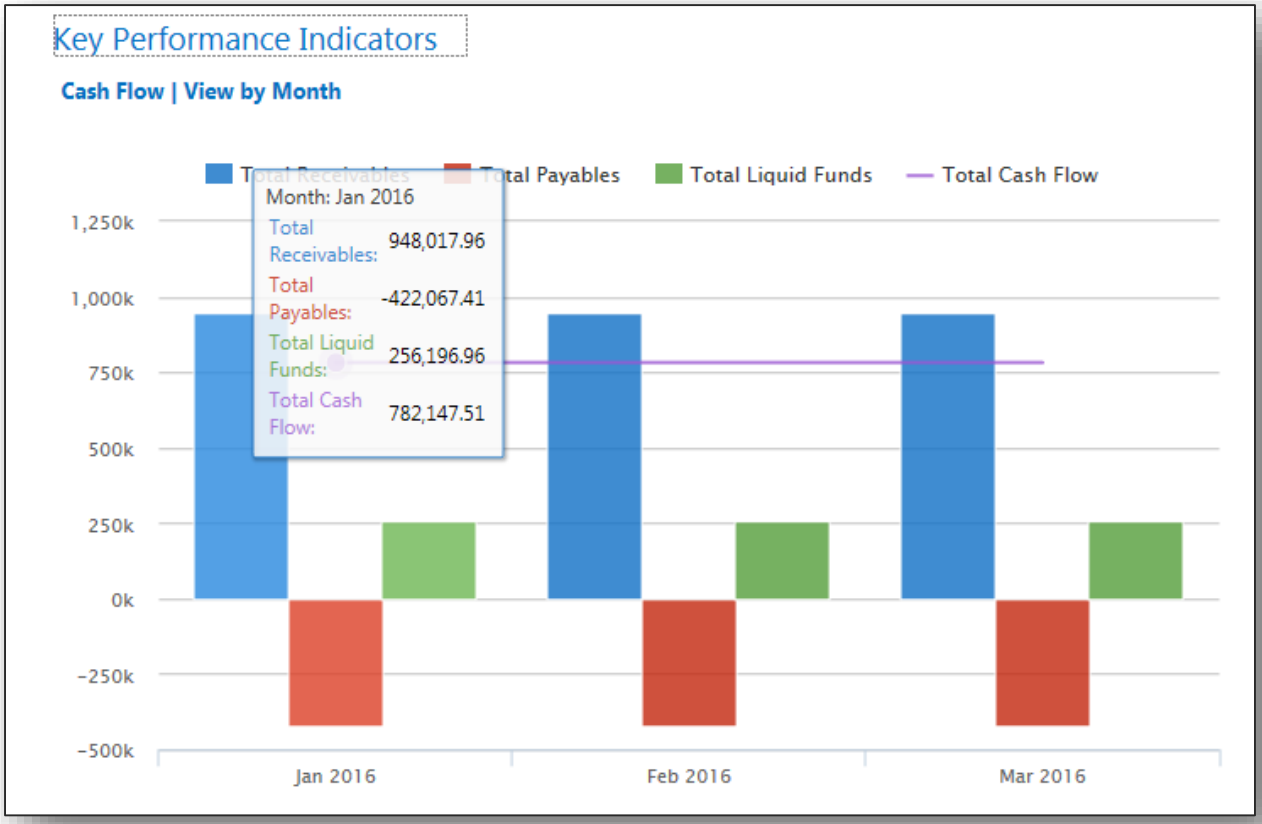
It was decided to implement Simplified UX in a new set of pages. The reason is that the configuration system does not support moving fields between groups or creating new groups. If the configuration system had been used, there would be many groups with few items in, and it would not be possible to provide a nice grouping of information.

Role Center – KPIs

General Principle

The owner of a small business must have an overview of the company in a single place, the Role Center. On the customer card, there is also a chart showing aged accounts receivables. For tablet users, this is shown as expressive tiles in a FactBox. We have implemented three parts for this purpose:

1. Key Performance Indicators Chart – This chart combines the eight different charts in the Role Center. The user is able to go through them one by one or select a chart from the list. It is possible to drill down into the data to investigate the financial data.



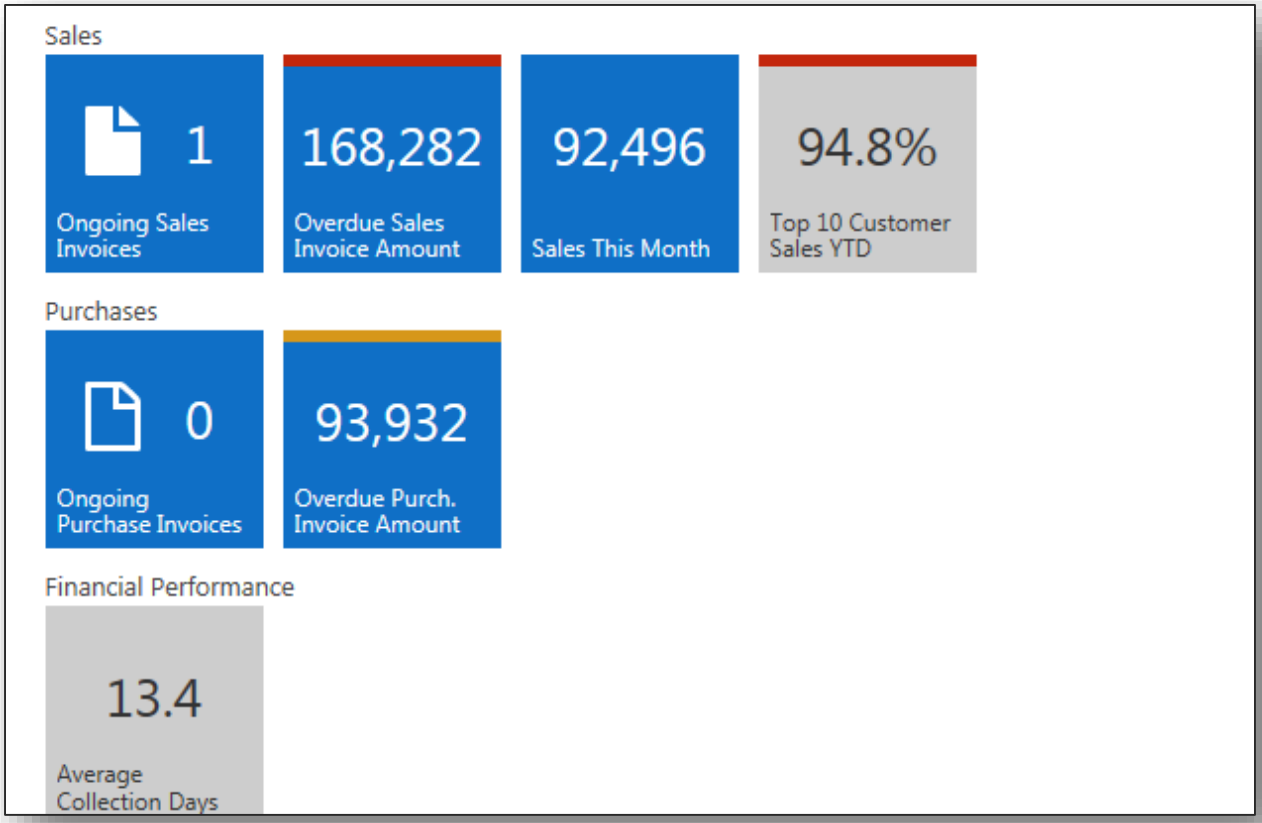
2. Trial Balance Chart – This chart shows the most important financial data so user can see at a glance how well the company operates. All data supports drilldown so the user can take action on the data. It is possible to go to previous/next periods to investigate the trends.

Microsoft Dynamics NAV

Trial Balance

Description	11/01/15..11/30/15	12/01/15..12/31/15	01/01/16..01/31/16
Total Revenue	766,348.65	781,403.91	99,620.27
Total Cost	-286,332.53	-415,526.95	-111,484.68
Gross Margin	480,016.12	365,876.96	-11,864.41
Gross Margin %	**62.64**	**46.82**	***-11.91***
Operating Expenses	212,029.23	469,069.86	2,044.03
Operating Margin	267,986.89	-103,192.90	-13,908.44
Operating Margin %	**34.97**	***-13.21***	***-13.96***
Other Expenses	152.74	90.66	112.33
Income before Interes...	**267,834.15**	***-103,283.56***	***-14,020.77***

3. Expressive Tiles – Instead of just showing the number of documents in the system, it is possible to implement so-called expressive tiles to show the most important information that user view frequently.



Microsoft Dynamics NAV

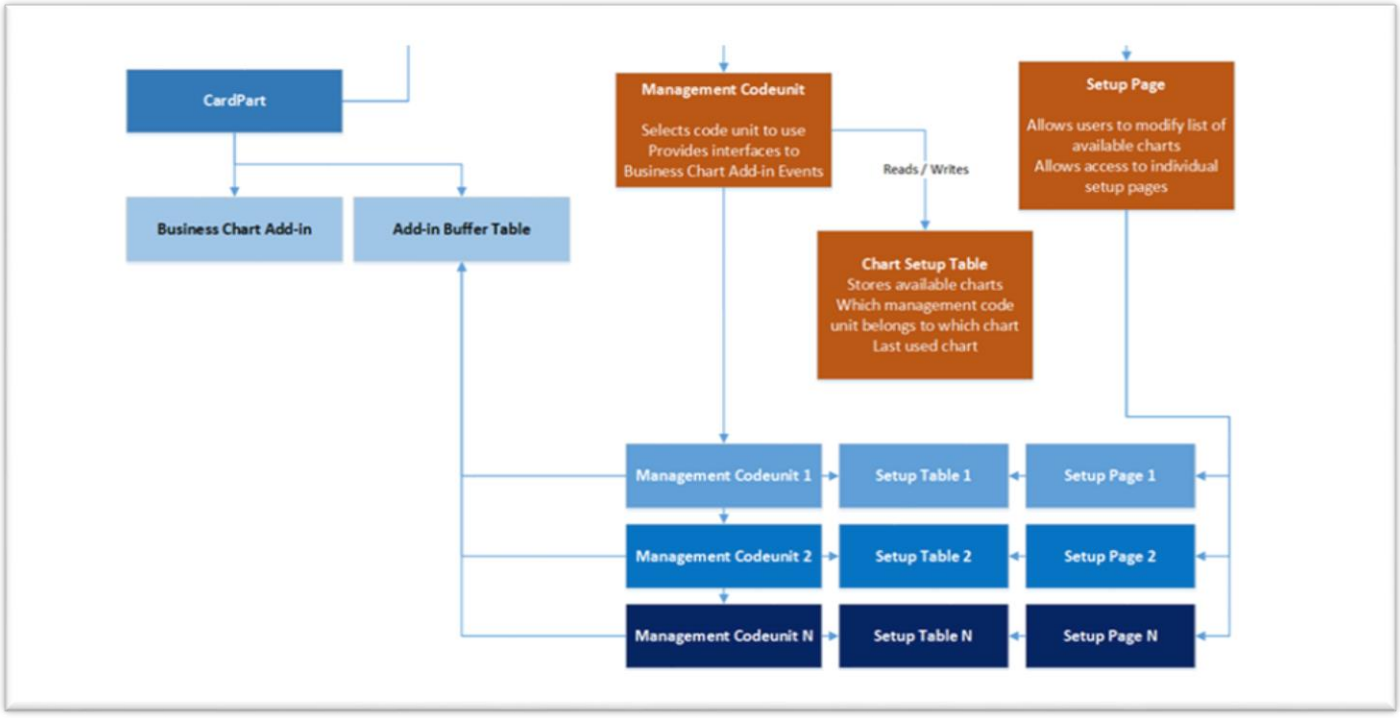
Implementation

Implementation of Key Performance Indicators Chart

This chart uses the following objects:

- Page 1390, Mini Generic Chart – The main page. It is recommended to start investigating the code from here.
- Codeunit 1315, Mini Chart Management – The main code unit that controls the logic.
- Table 1310, Mini Chart Definition – The table that stores the charts that exist.
- Table 1311, Mini Last Used Chart – The table that stores user personalization and which chart has been selected last.

Each chart that is hosted inside the Key Performance Indicators chart has its own management code unit and table to store user personalization. Some of the charts are based on Acccount Schedule records so they share the management objects in a generic way. This is done so new charts can be added with minimal changes to existing code. The relationship is described in the following diagram:



You can find more information about how to implement this chart pattern and custom charts here.

Implementation of Trial Balance Chart

This chart is based on account schedule definitions. It uses the following objects:

- Page 1393, Mini Trial Balance – The page definition. It is implemented as a grid layout to provide better rendering. (Alternative implementation would be to use a repeater, but this would not look as good. Also, at the time of implementation, repeater did not support drilldown, which is crucial to the scenario.)
- Codeunit 1318, MiniTrialBalanceMgt – The codeunit that provides logic for loading the values and interacting with the page.
- Table 1312, Mini Trial Balance Setup – Stores the Account Schedule Name and Column Layout Name values that contain the definitions on how data is calculated. Calculation is done by using the logic from account schedules.

Since it would be too difficult for the owner of a small business to set up account schedules, we have not provided the UI and actions for setup on the chart itself. Advanced users (partners, super users, consultants) can do the setup in the Account Schedules Chart Setup window.

Implementation of Expressive Tiles

Important considerations when implementing tiles is the performance. Each time the Role Center is opened, the values are recalculated if they are not cached. Since the Role Center is the first page that will be opened, this affects the loading of the product, which is very important scenario. We have therefore used queries for some tiles, because the Query object provides better performance than when doing calculation with loops.

Tiles also support user-specified setup that shows a bar over the value to indicate the severity of the data.

Expressive tiles are implemented on page 1310, Mini Activities.

Ideas for Improvement

- A permissions system. All of the KPIs are implemented for small companies, so there were no requirements to set up permissions and access restrictions. It is possible to implement a generic permission system so these parts can be reused by larger companies.

- Two chart parts so users can compare different data. This was not a requirement but it is possible to provide two different chart parts so users can compare and analyze the data from different

periods and/or charts.

- The Mini Trial Balance chart is not configurable. The chart was implemented for small companies in Denmark, so there was no need to provide configuration for this chart. Configuration could be added so the chart can more easily be adopted by other companies and markets.

Instructions in the UI

Instructions in the UI cover welcome video, Getting Started actions, dismissible dialogs, task-oriented page Help, and tooltips.

General Principle

One of the most important requirements for Simplified UX is to make it usable out of the box. Users have the option to try the product for one month and then decide if they want to pay for subscription. Therefore, the ability to complete the tasks without training is very important. In addition, users are not so willing to invest time on training because they expect the product to be simple to use.

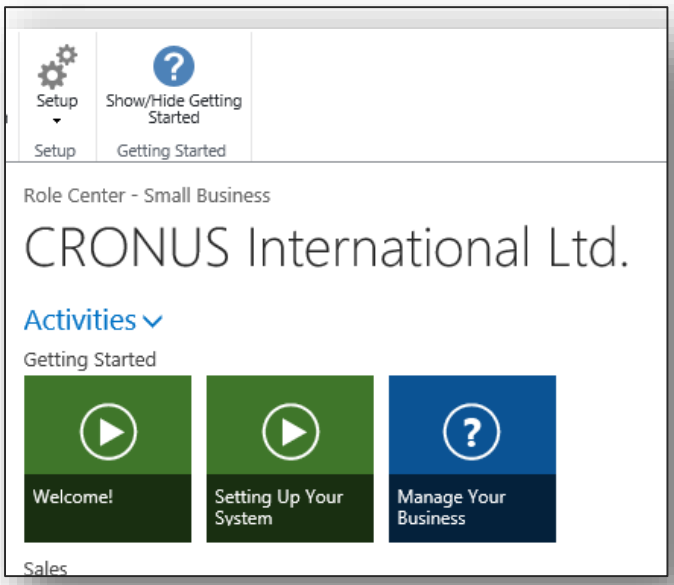
The UI should not require instructions, it must be self-exploratory. All usability issues should be fixed. Unfortunately, this is very expensive and not always possible. However, we can improve the user experience by providing instructions in the UI.

The main guideline when adding instructions in the UI is that it must not get in the way and should be dismissible. The reason is that instructions are very helpful in the beginning, but for experienced users they get in the way and can be annoying. All types of instructions implemented for Simplfied UX follow this guideline.

Welcome Video

On the first logon, a welcome video starts. This is implemented because usability studies show that users that get very short instructions on how to interact with the product perform much better and give the product higher ratings. In simplified UX, a number of short instructions are compiled in the video.

Getting Started Actions



On the Role Center, the first set of actions are the Getting Started tiles:

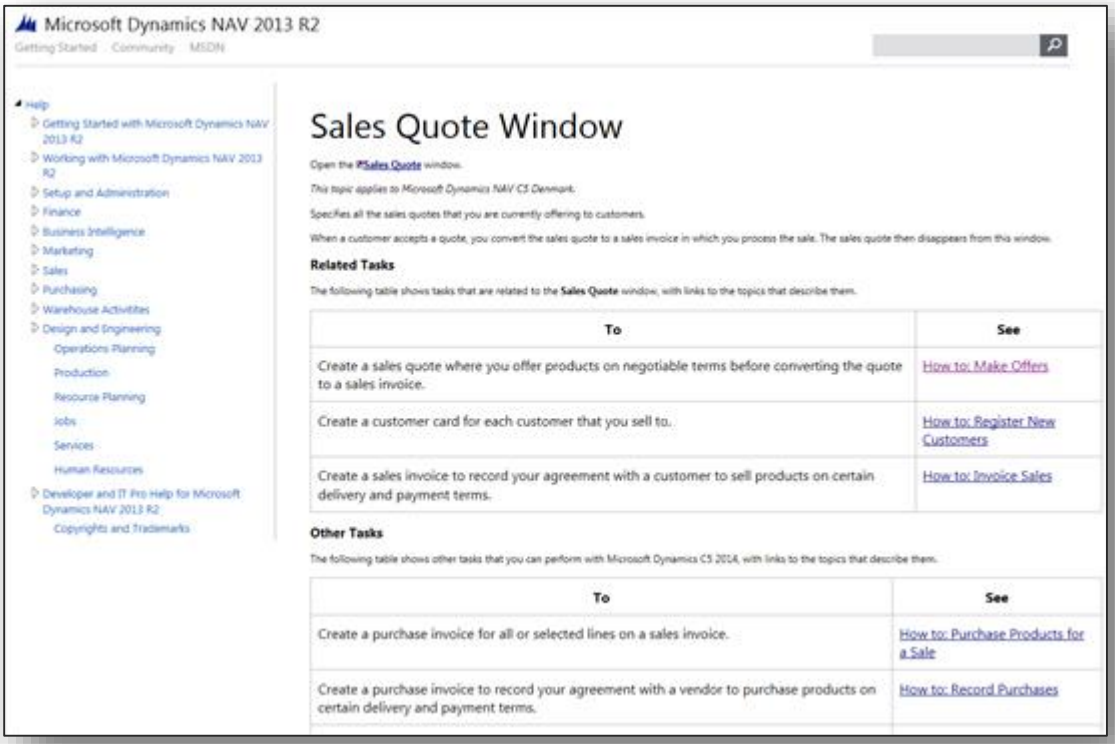
- The first tile, Welcome, is the welcome video, which also starts automatically on the first logon.
- The second tile, Setting Up Your Business, is a video on how to set up the system.
- The third tile, Manage Your Business, is a link to a help topic that lists all the supported business processes and how to perfom them.

The Getting Started part is always shown in a prominent place when users first log on, but the part may be in the way when users become familiar with the product. Therefore, a Show/Hide Getting Started button is implemented so users can hide the part and show it only when they need to take a look at the instructions again.

Task-oriented Page Help and Tooltips

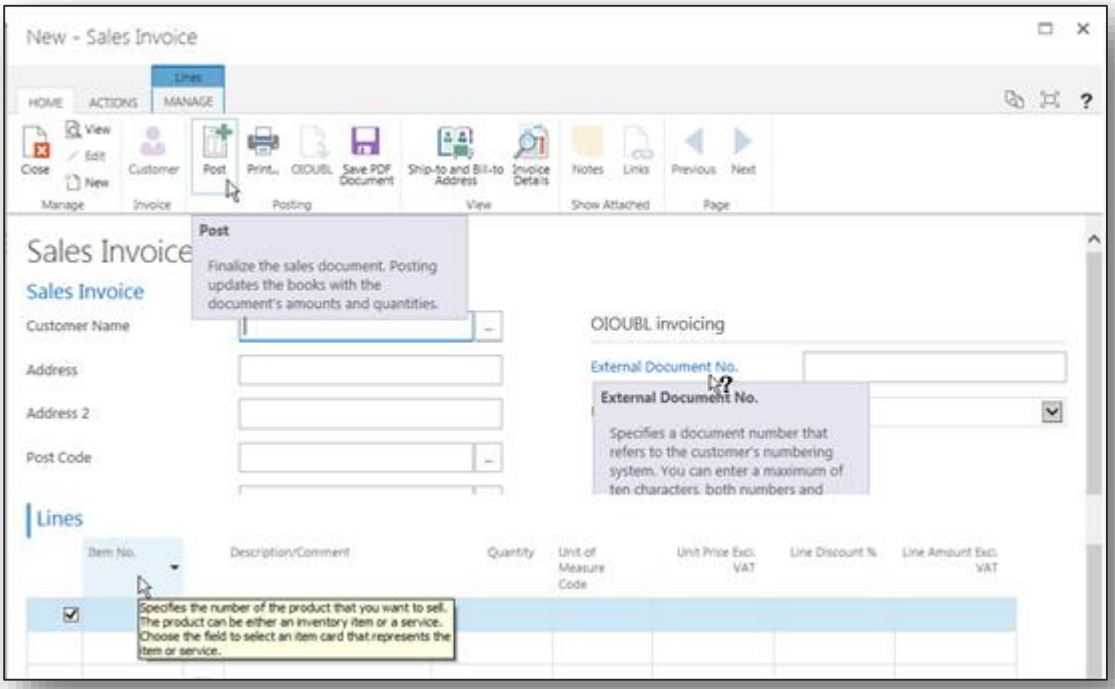
For 37 key pages in Microsoft Dynamivs NAV 2015 with Simplified UX, the Help icon in the top right corner opens a topic that lists tasks related to the page with links to procedural help on how to perform the tasks.

Microsoft Dynamics NAV



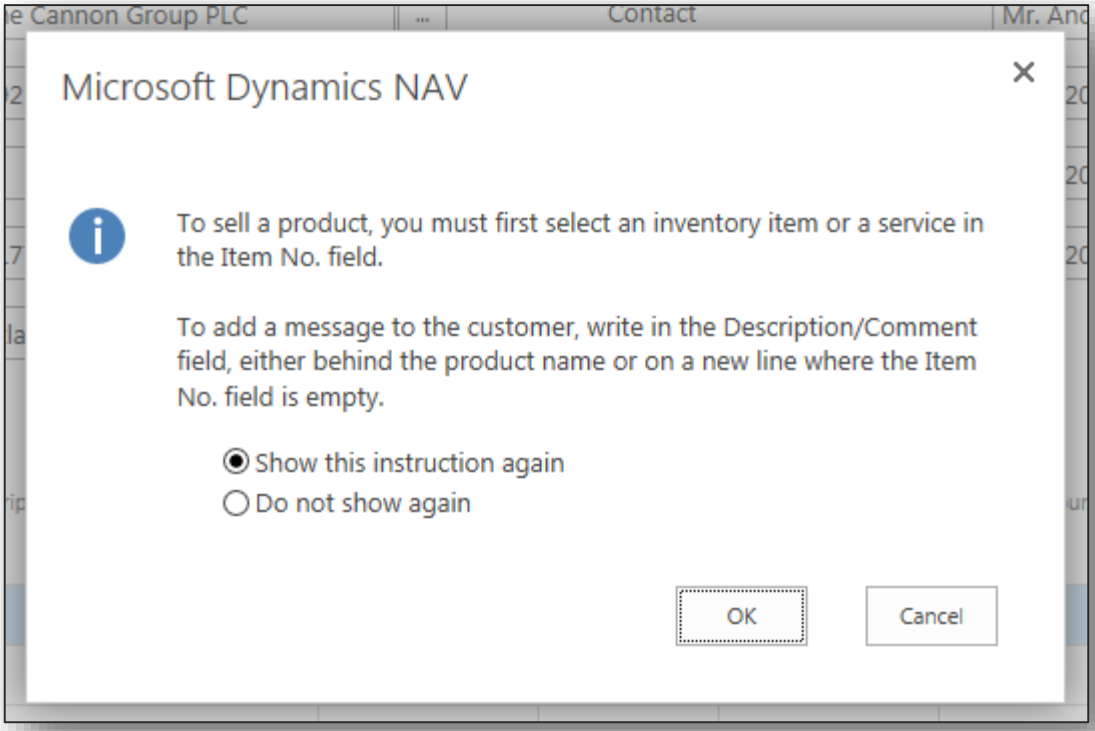
Tooltips are provided for all actions and fields on the Simplfied UX pages. Platform improvements in NAV 2013 R2 provide ability to create tooltips for actions and all kinds of fields in the web client simply by filling the TooltipML property on the page object.

Microsoft Dynamics NAV



Dismissible Dialogs

Dismissible dialogs show an instructional message in connection with “tricky” user interaction, with the user option to not show the message again. This is practical when there is a risk that users enter text in the wrong place or to explain the behavior of a somewhat hidden feature. Such dialogs are highly disruptive so they are only implemented for high-severity UX issues.



Additional Information

All types of instructions described in this document are easy to implement in your solution. You can find detailed implementation information here.

Microsoft Dynamics NAV

Auto-filled and Hidden No. Field

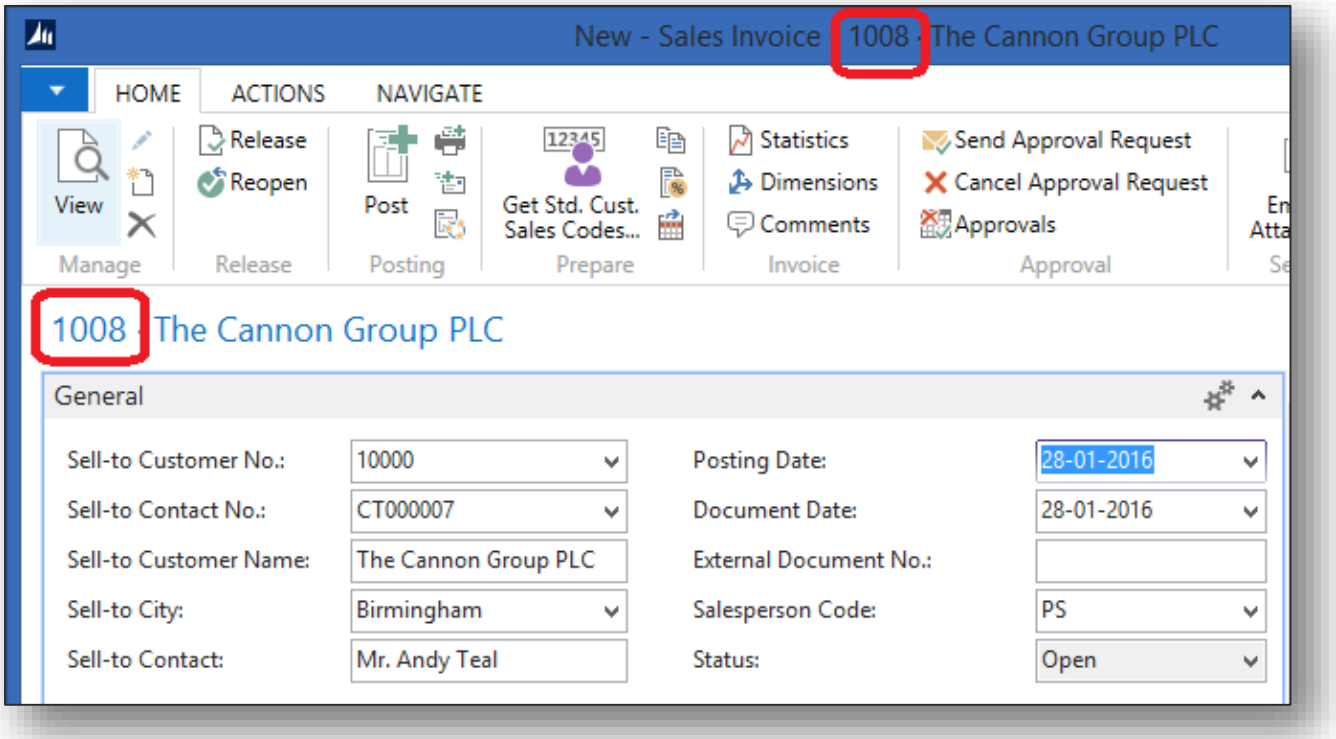
Reasons for Change

From the user's perspective, it was always a bit strange that the No. field is the first editable field because the user cannot continue to another field until it was filled. Users could also be confused about the difference between manual entry and automatic entry based on a number series. In addition, it requires at least two key strokes to get past the No. field even if it is automatically filled.

At the same time, if the numbers can be assigned manually oruser can select other Nos. series (like in on Sales Order in demodata), user still should be able to change No. field.

General Principle

With the change, the No. is not completely visible. Users still can see No. in two places, as shown below:



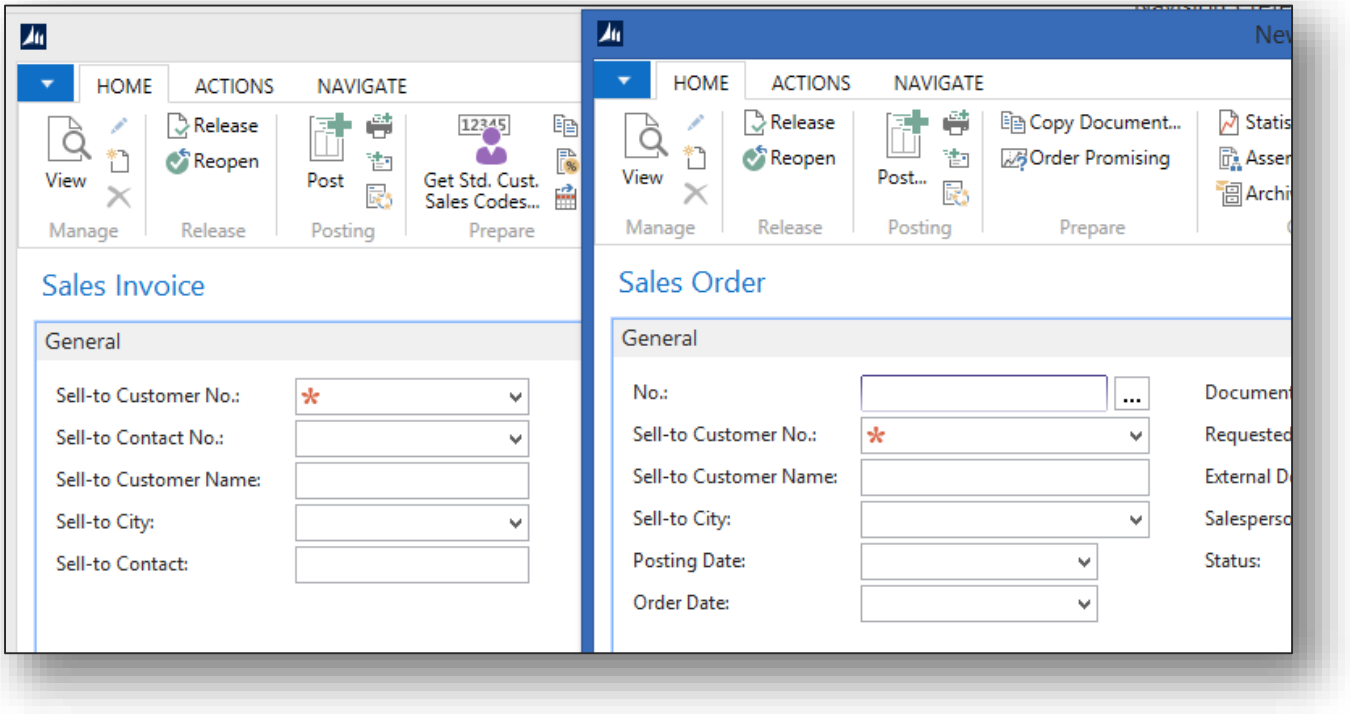
Microsoft Dynamics NAV

Rules to Show/Hide the No. Field

The No. field is automatically filled and hidden in the following cases:

- The No. field is shown on all posted documents.
- The No. field is hidden for already created documents.
- When creating a document, the No. field is only hidden if all the following are true:
  - One number series is assigned.
  - No related number series exists.
  - The assigned number series is set as the default, and manual entry is disallowed.
  - The next number in the series is available (it is not the last one).

In all other cases, the No. field is shown. The following example from demo data shows that sales invoices only have one number series, so the No. field is hidden, while sales orders have multiple related number series, so the No. field is visible.

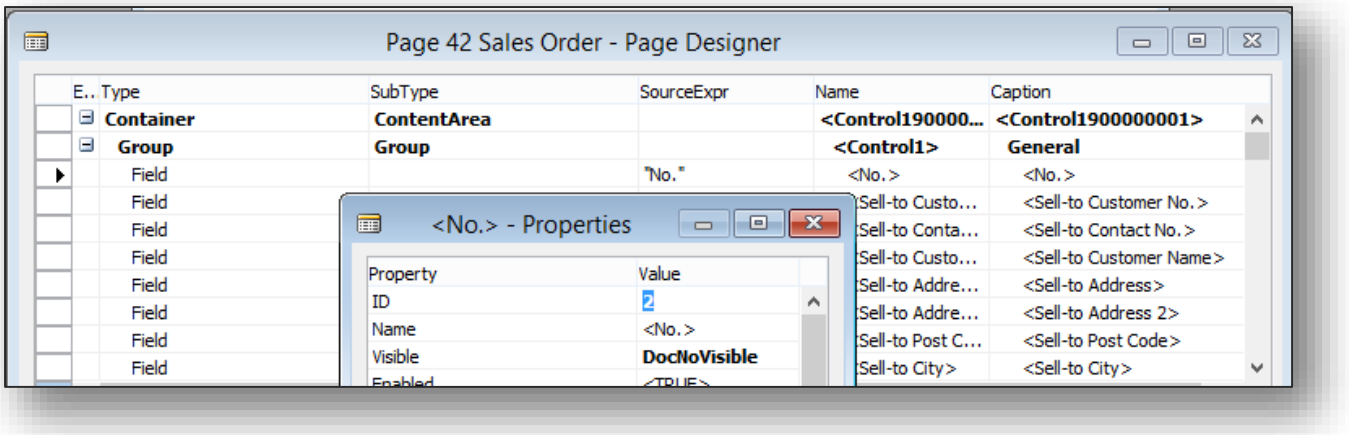


The code that control the visibility is places in codeunit 1400, DocumentNoVisibility.

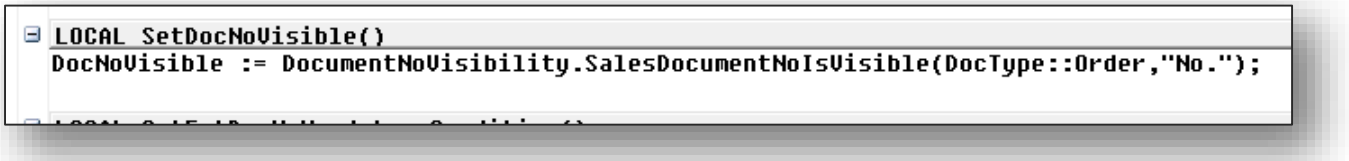
Microsoft Dynamics NAV

Implementation

Since de-promoting the No. field is not supported on the web client, the implementation uses the Visible property of the page control. The implementation follows the same pattern on all pages.
The Visible property of the No. field is controlled by the DocNoVisible variable:

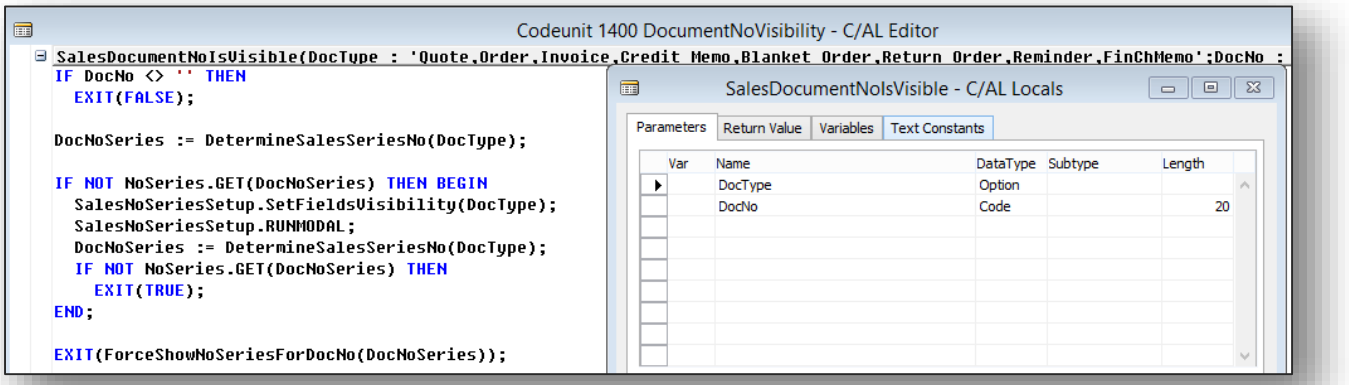


We update this variable only in the SetDocNoVisible function. We call this function with the OnOpenPage trigger. SetDocNoVisible is a wrapper function that calls codeunit 1400, DocumentNoVisibility:



```
LOCAL SetDocNoVisible()
DocNoVisible := DocumentNoVisibility.SalesDocumentNoIsVisible(DocType::Order,"No.");
```

All logic is concentrated in this codeunit, so if you want to change the logic, this is the place:

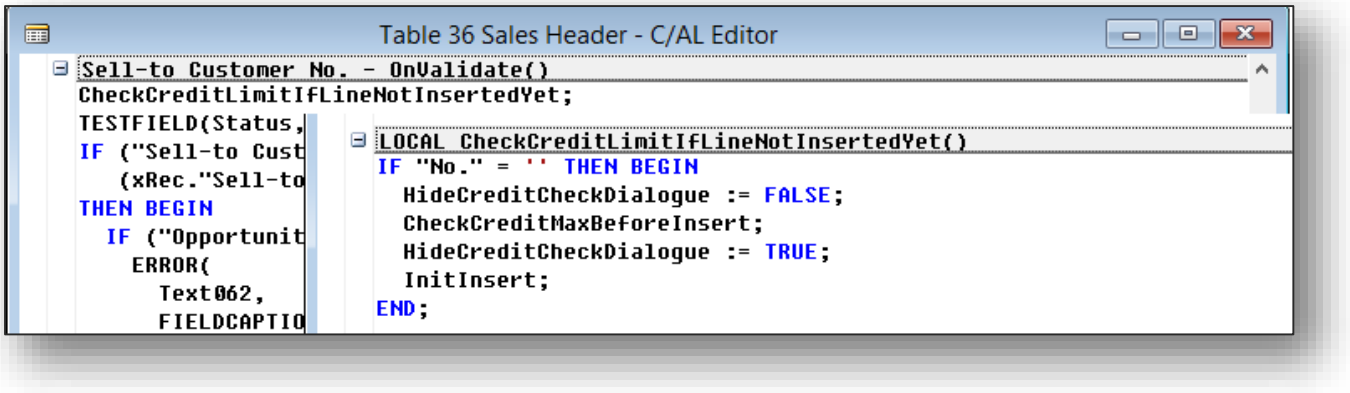


Microsoft Dynamics NAV

For now, the codeunit only has two public functions, to deal with sales and purchases respectively. Notice that the DocType variables of these functions are extended with other document types than Sales Header and Purchase Header. This is done to accommodate pages based on other tables, such as Reminder and Fin.Ch.Memo. This means that you can easily extend the solution.

The Sales Header and Purchase Header tables are modified because the sequence of triggers is a bit different when the user creates any new document than when the user creates a new document for a selected customer or vendor.

Therefore, we had to make sure the record is inserted while running validation on the Sell-to Customer No. and Buy-from Vendor No. fields:



Where Implemented

The Auto-filled and Hidden No. Field feature is implemented on the following core unposted sales and purchase documents:

ID	Name
41	Sales Quote
42	Sales Order
43	Sales Invoice
44	Sales Credit Memo
49	Purchase Quote
50	Purchase Order
51	Purchase Invoice
52	Purchase Credit Memo
434	Reminder
446	Finance Charge Memo
507	Blanket Sales Order
509	Blanket Purchase Order
6630	Sales Return Order
6640	Purchase Return Order

Ideas for Improvement

The implementation could be improved by inserting a record before opening the document page, especially for tablet users when they create, for example, a sales invoice for a selected customer. Then all auto-filled fields, such as the Sell-to Customer Name and No. field, would be prefilled immediately. The only issue is that if a user presses New by mistake, then a document will be created. In the current implementation, the user can press Escape and the number series will not be affected.

Mandatory Fields

Reasons for Change

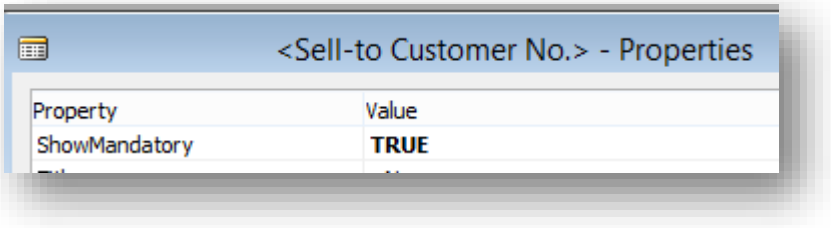
Less experienced users often do not know which fields to fill in as a minimum, for example on the customer card, for the customer to be used in processes, such as invoice posting. All the help they get is an error message when they try to complete the process.

Solution

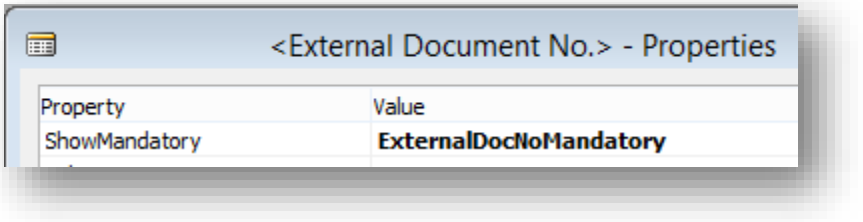
To help users with this issue, the ShowMandatory property has been introduced. In this context, "mandatory" means that a field is marked with red asterisk to indicate to users that the they are expected to fill the field. The user can still leave page without getting error messages if the field is not otherwise validated by business logic.

Implementation

The ShowMandatory property is implemented on page controls as follows:



You can assign TRUE or FALSE, as well as a variable:



Microsoft Dynamics NAV

By using a variable, you can manipulate the Mandatory property dynamically:

```
LOCAL SetExtDocNoMandatoryCondition()
SalesReceivablesSetup.GET;
ExternalDocNoMandatory := SalesReceivablesSetup."Ext. Doc. No. Mandatory"
```

The possibility to manipulate the Mandatory property dynamically is a big difference from the NotBlank property on tables.

Limitations

If you want to use this functionality on numeric fields, you have to set the BlankZero property to Yes. It does not work on booleans because booleans in Microsoft Dynamics NAV only have two values.

The same is true for option fields: If the option string does not contain an empty value, the first option will be selected by default and therefore the asterisk will not be shown.

Create Customer and Vendor Cards on the Fly

General Principle

In small companies where few users cover multiple roles, the product must support the business users in creating master data records as they perform their business tasks. When no dedicated role exists to register new customers, the sales order processor must be able to register new customers on the fly as they prepare the first invoice for the customer.

In the standard product, the user has to go and open a new customer card, enter all the required information, and then go back to the newly created invoice to finish registering the remaining information. With Simplfied UX, the new customer card can be filled from templates with a few steps while creating the invoice.

Mandatory fields on the templates can be auto-filled with predefined values so that the user only has in a few customer-specific details.

Implementation

This functionality is provided by using two simple patterns: The Creating an Entity on the Fly pattern combined with the Template pattern. For more ionformation about the Template pattern, see the “Create Master Data Records from Templates” section.

Implementation considerations

- The Customer/Vendor No. field is not shown to the user. Therefore, the Customer/Vendor Name field is the first field that the user has to fill, and this must trigger the on-the-fly creation.
- Since mandatory fields exist on customer/vendor cards, templates must be used to fill these fields automatically.
- The presentation layer must have simple, one-line calls to the logic.

Microsoft Dynamics NAV

Code

Customer Creation

All code is defined in a single place, COD1302, Mini Customer Mgt., which also contains the on-the-fly functionality for creating a customer based on customer name and the customer-management code specific to Simplified UX functionality. The call to the used templates is also integrated in this codeunit.

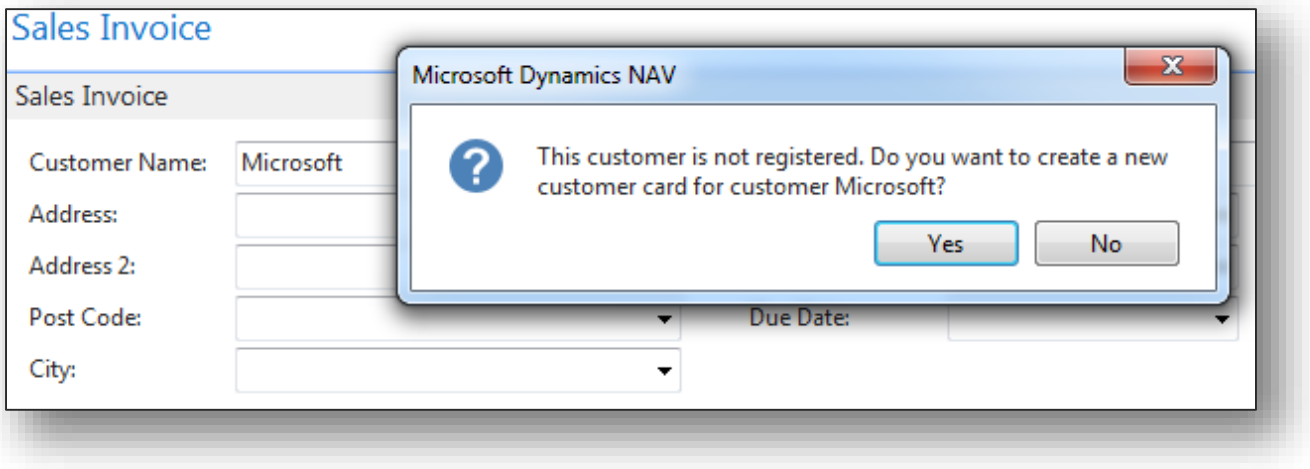
UI – Controls

- *OnValidate – on Sell-To/Pay-to/Bill-to* – One function call is responsible for creating/selecting the specific entity.
    - IF MiniCustomerMgt.ValidateSellToCustomer(Rec,xRec) THEN CurrPage.UPDATE;
- *OnLookup – on Sell-To/Pay-to/Bill-to* – One function call is responsible for creating/selecting the specific entity.
    - IF MiniCustomerMgt.LookupSellToCustomer(Rec,xRec) THEN CurrPage.UPDATE;

Note: Since the primary key of the created entity is not added to the document, special attention should be paid to the order that the triggers are executed in, because there is a slight difference compared to standard functionality and unwanted behavior may occur.

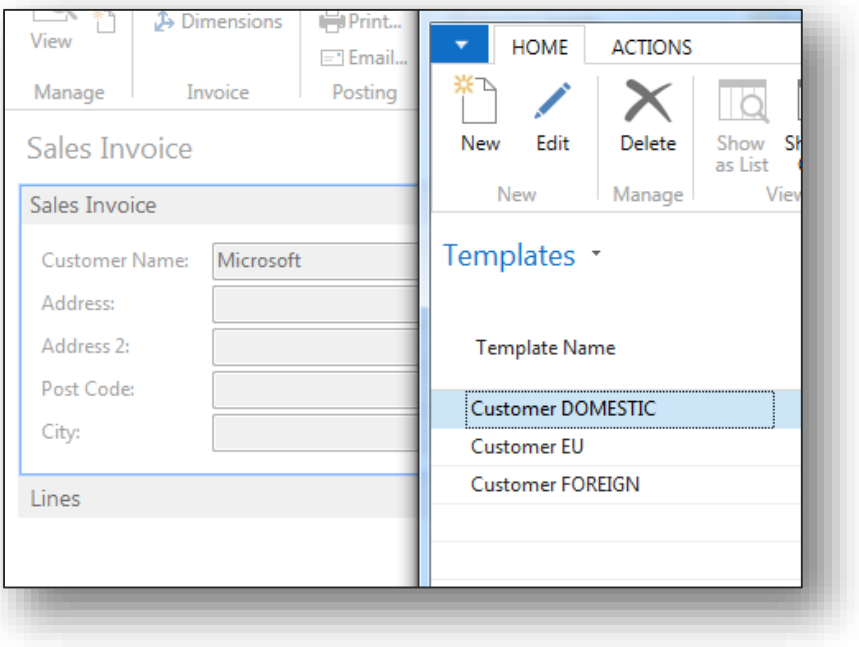
UI – Flow

1. Enter a non-existing customer name on a sales invoice.

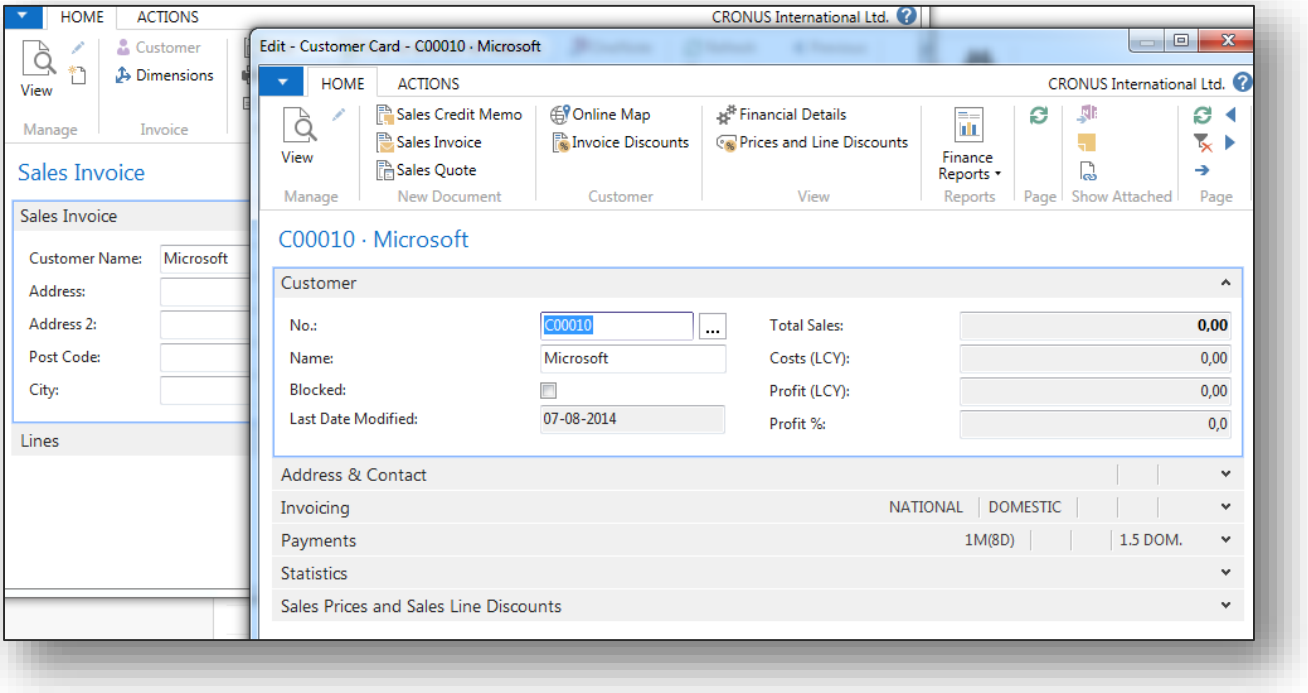


Microsoft Dynamics NAV

2. Select a customer template.



3. The customer card is auto-filled with the template values.



Microsoft Dynamics NAV

Where Implemented

The Create Customer and Vendor Cards on the Fly funtionality is only implemented for customer and vendor cards on selected sales and purchase documents in the 13xx page object range.

Microsoft Dynamics NAV

Create Master Data Records from Templates

General Principle

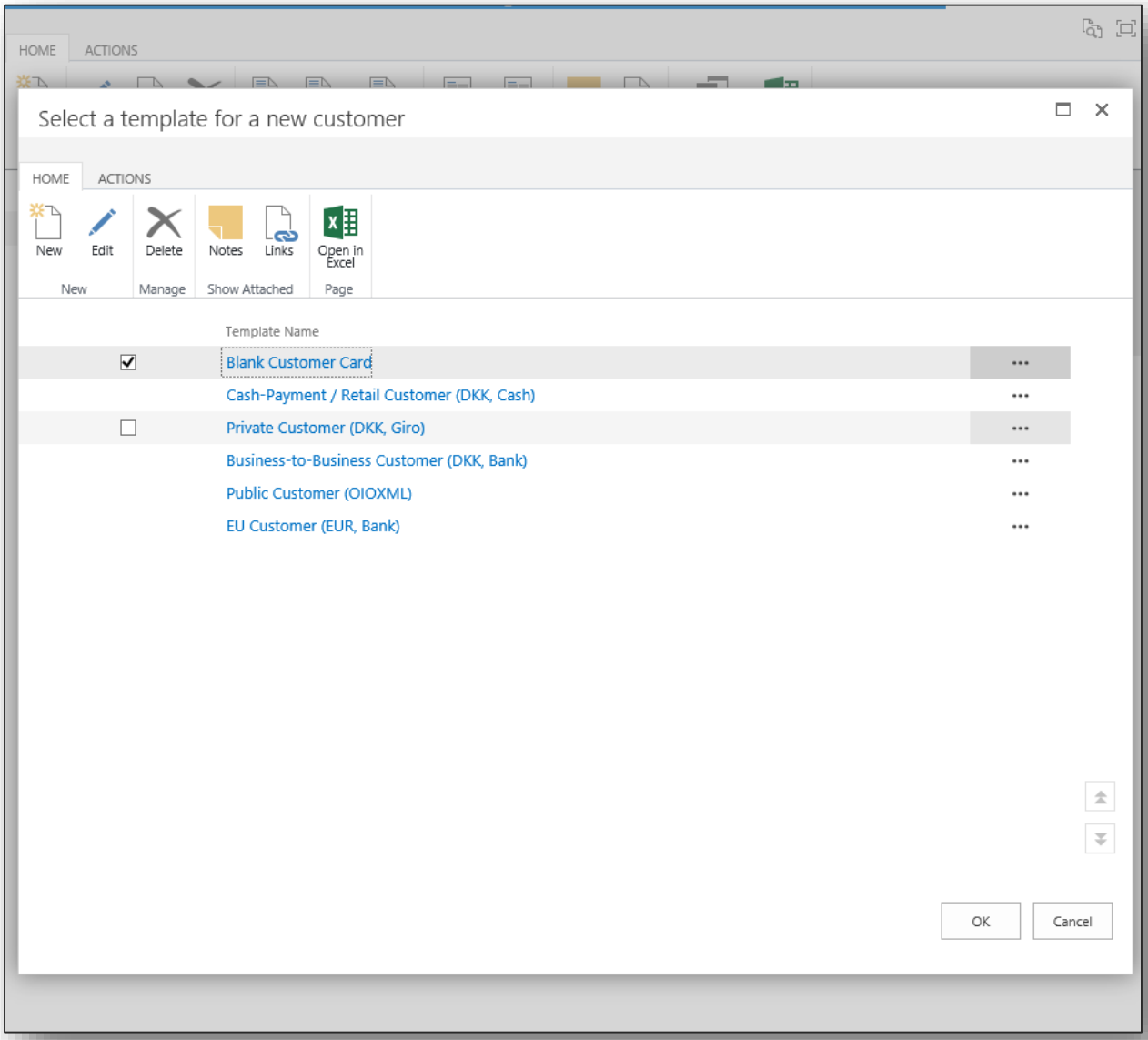
Many new users struggle with the basic task of creating new customers or items. They have problems knowing which fields they must fill and what are the correct data combinations, for example for VAT groups and payment terms. In addition, it is tedious and error prone that users must repeatedly fill the same sets of data.

To enable a simple experience for small-business users, they can create certain master data records (customer, vendor, and item cards) from templates. This simplifies the creation process and makes users more productive.

Using Templates

When users select New in the Customers window, they are prompted to select from the following list of templates:

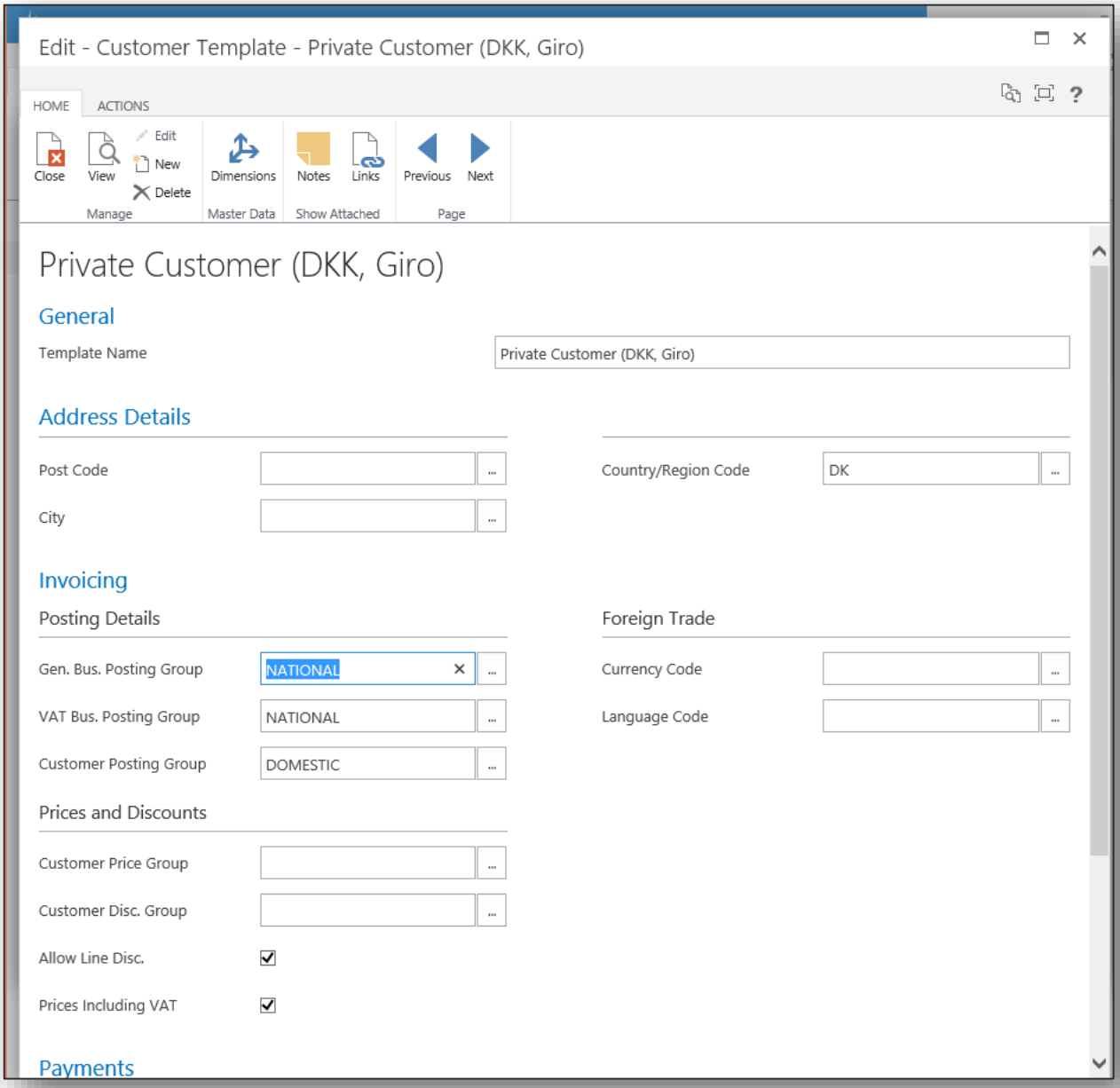
Microsoft Dynamics NAV



The Blank Customer Card template opens an empty customer card where the user must can populate all the mandatory fields. All other templates open customer cards where all mandatory and other relevant fields are pre-filled, so that the user only has to provide the customer's name and contact information. This saves a lot of time, because it is normally necessary to fill 9-12 fields on a customer card before it can be used to post a sales document.

Creating and Updating Templates

It is possible to view and edit templates at any time, so that users can quickly make updates while performing business tasks.

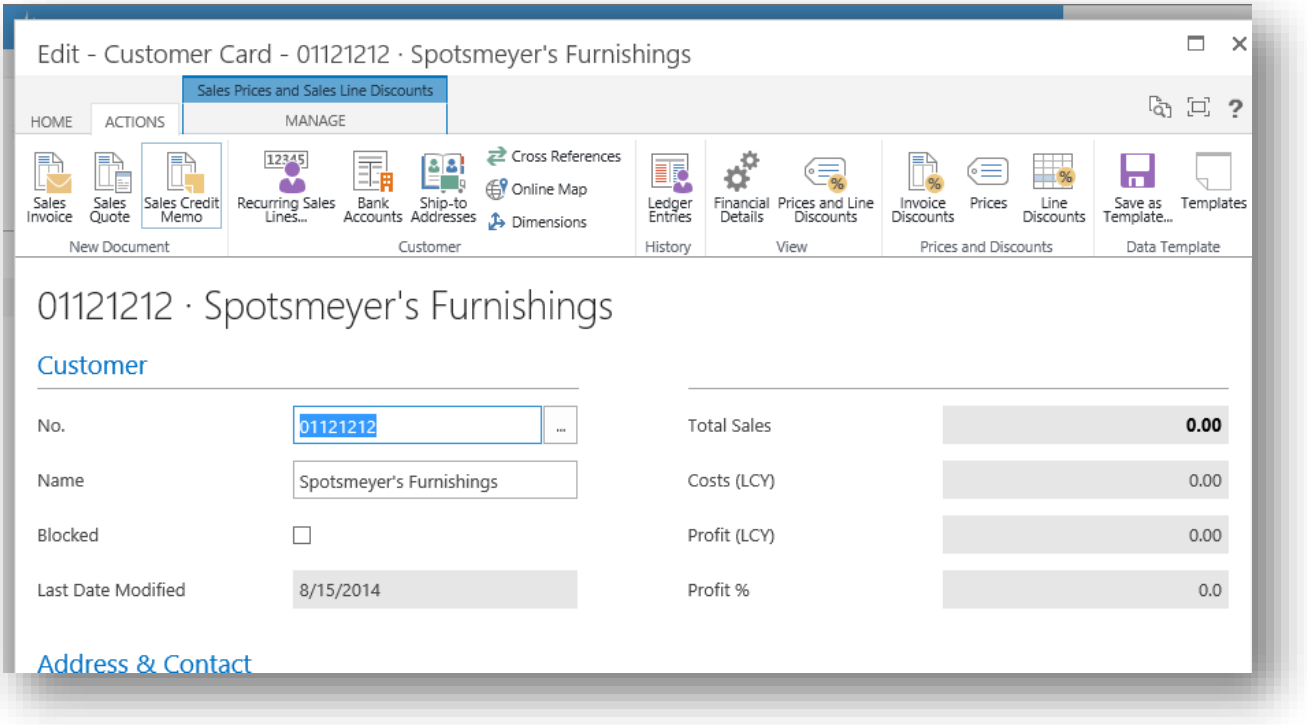


The template window is designed to resemble the master data card that it represents. It is implemented with simple validation and support for lookup to setup options.

Microsoft Dynamics NAV

Save Existing Records as Templates

User can save an existing master data record as a template. The customer, vendor, and item cards contain a Save as Template action that will create a template based on the master data record.



Additional Information

Templates for master data records are easy to implement in your solution. You can find detailed implementation information here.

Microsoft Dynamics NAV

Item as a Service

General Principle

In the standard version, the purchase and sale of goods is done using Item entity while the purchase and sale of services is done using the Resource entity. Because the Resource and Item entities are maintained in separate data structures, the following issues prevent a seamless experience for companies trading in both goods and services:

- Poor usability. Users have to switch between two lists by choosing the appropriate type on document lines. From the user's point of view, goods and services are both products that should exist in the same list.
- Differint look and feel and functionality gaps. Functionality for the Item and Resource entities has evolved unevenly over time, so that the Item entity is more advanced today. For example, the Item entity has richer price and discount functionality compared to the Resource entity.

To solve these issues, the Item entity has been extended in Simplified UX to allow defining it as either a service or as an inventory item. This is achieved by introducing the Type field on the item card to allows the user to categorize the item as an inventory item or as a service. As a consequently, the Type field on document lines is removed, so that users no longer have to select a type before selecting the product.

Implementation

A new option field, Type=Inventory,Service, is introduced on the Item table. When Type=Inventory, the item will have same behavior as before this change. However, when Type=Service, the item can only participate in functionality and capabilities that are not related to physical goods.

Microsoft Dynamics NAV

The following table outlines the functionality and capabilities that support items of type Service.

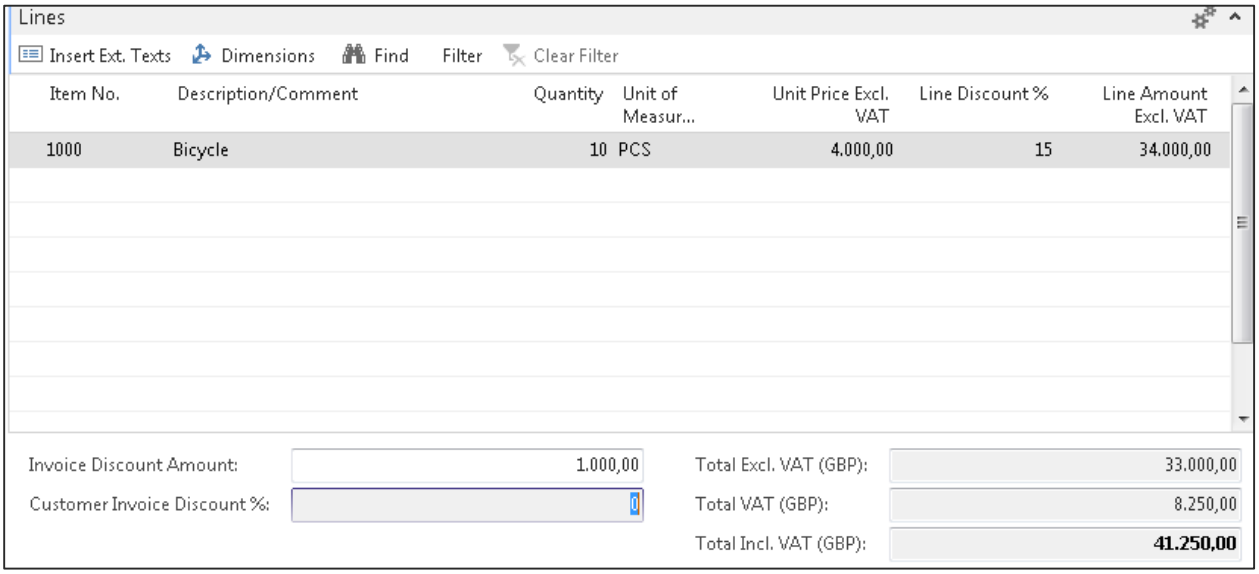
	Supported	Not Supported
Usage	• Purchasing • Sales • Service • Jobs	• Production • Item Journals • Transfer • Assembly • Warehouse
Reports	• Purchase Analysis • Sales Analysis	• Inventory • Availability • Item Tracing
Functionality	• Dimensions • Pricing / Discounts • Extended Texts • Variants	• Multiple Locations • SKU • Revaluation • Planning • Item Tracking • Reservations • Order Tracking • Intrastat
Posting	• Item Ledger Entry *Open=FALSE, Remaining Quantity=0. The entries are closed immediately since they have no inbound/outbound relationship.* • Value Entry *Inventoriable=FALSE to prevent it from affecting the Inventory account.*	• Item Application Entry *No entries are created because they have no inbound/outbound relationship.* • Post Inventory To G/L *No entries are created because inventory accounts will be affected.* • Average Cost Adjustment Point Buffer *No entries are created because the cost will be tracked differently due to the lack of costing method.* • Item.CostIsAdjusted *Adjustment will be turned off.* • Item.CostingMethod *Only necessary for inventory* • Item.UnitCost *Is not affected by posting*

Totals on Sales and Purchase Documents

General Principle

A small-business user wants to see all important information about a document directly on the document. The information must be updated when the user changes values on the document, such as quanities, prices, currencies, and discounts.

The most important information that users need to see is totals and discount amounts. Therefore, all sales and purchase document pages are enhanced with information about totals, VAT, and discounts at the bottom of the page. In addition, functionality for updating this information on the fly is implemented so that any change that affects these attributes are automatically calculated and presented to the user.



The layout of the information fields is made to resemble how such information is typically placed on a printed document. The currency code is included in the caption of the three Total fields to clearly indicate which currency the invoice uses.

Microsoft Dynamics NAV

Implementation

Totals are implemented on both active documents and their related posted documents.

Implementation Considerations

- All code must exist in one place.
- Existing amount calculations, including for VAT and discount, must be reused.
- The presentation layer must be separated from the logic.
- The presentation layer must have simple, one-line calls to the logic.
- Totals and discounts must be updated on the fly with any change that affects amounts.

The following sections outline the artifacts that were implemented to achieve the desired functionality.

Code

Totals Calculation and Update Logic

All code is defined in a single place: COD1301, Mini Document Totals.

Discounts Calculation

Two codeunits, one for the sales side and one for the purchase side, calculate/apply the designated invoice discount: COD1300, Mini Sales-Calc Disc. By Type and COD1310, Mini Purch-Calc Disc. By Type.

UI – Update Controls

Because posted documents can no longer change, the code is added to the subforms in slightly different ways for active and posted documents.

- Subform on active documents:
  - *OnAfterGetCurrRecord* – One function call updates controls related to totals and discounts: *MiniDocumentTotals.CalculateSalesHeaderTotals(TotalSalesHeader,VATAmount,Rec); UpdatePage;*
- Manual invoice discount – One function call when applying a manual invoice discount:
  - *MiniSalesCalcDiscByType.ApplyInvDiscBasedOnAmt(TotalSalesHeader."Invoice Discount Amount",SalesHeader);*
  - "Decorators" to recalculate the amounts on the fly:
    - Totals must always be updated when the user changes a cell.
    - *UpdatePage* is added to all triggers that change amounts to refresh the total controls:

CurrPage.UPDATE;
SalesHeader.GET("Document Type","Document No.");
MiniSalesCalcDiscByType.ApplyDefaultInvoiceDiscount(TotalSalesHeader."Invoice Discount Amount",SalesHeader);

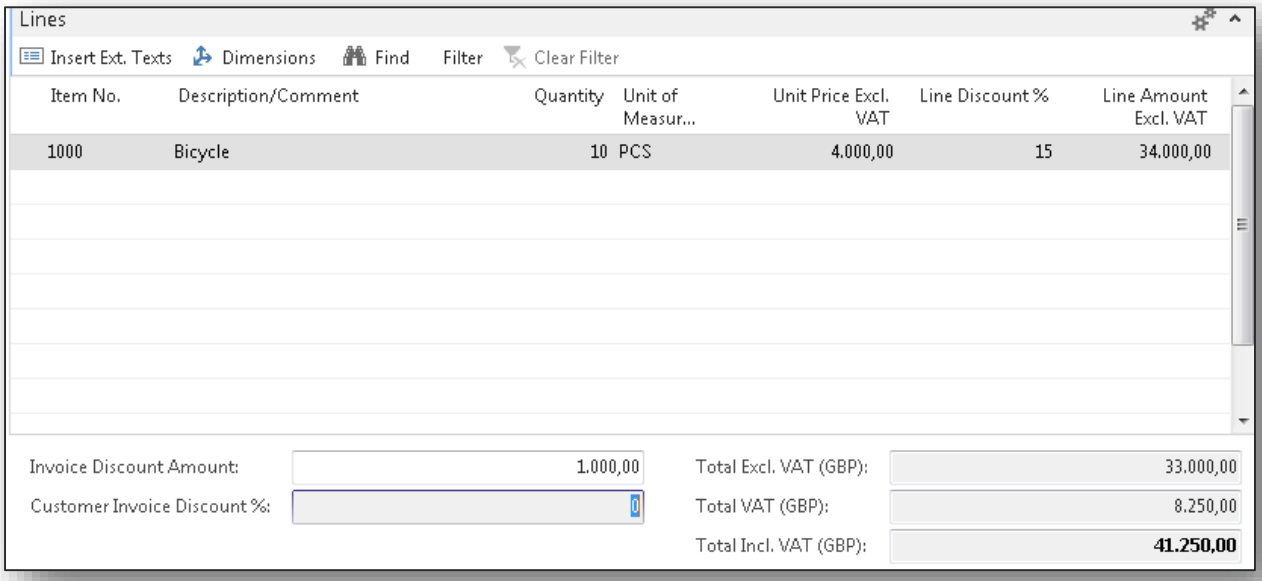
- Subform on posted documents
  - Only the *OnAfterGetCurrRecord* has code because posted documents cannot be changed so totals do not have to be updated:
    MiniDocumentTotals.CalculatePostedSalesInvoiceTotals(TotalSalesInvoiceHeader,VATAmount,Rec);

UI Definition

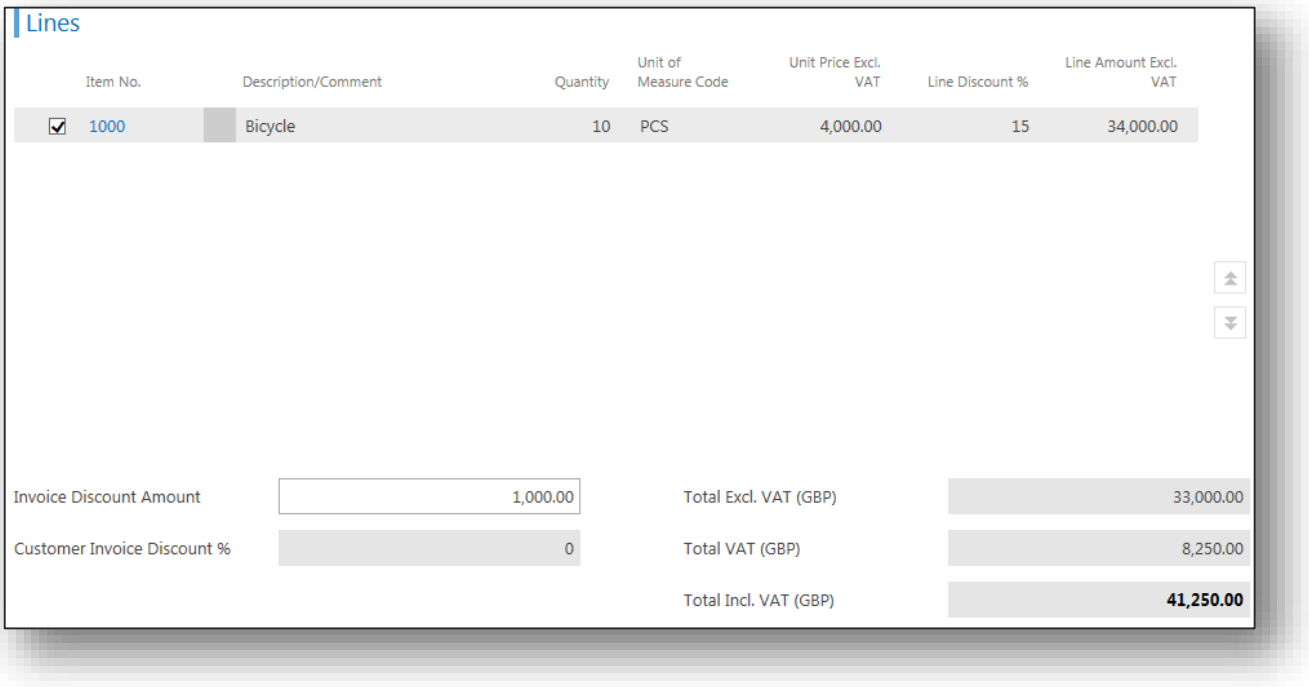
If you design one of the Simplified UX subform pages (1300 range), you will find the following structure:

Group	**Group**		**<Control103>**	**<Control103>**
Group	**Group**		**<Control106>**	**<Control106>**
Field		TotalSalesHeader."Invoice Discount Amount"	Invoice Disco...	Invoice Discount Amount
Field		MiniSalesCalcDiscByType.GetCustInvoiceDiscountPct(Rec)	Cust. Invoice...	Customer Invoice Discou...
Group	**Group**		**<Control131>**	**<Control131>**
Field		TotalSalesHeader.Amount	Total Amount...	Total Amount Excl. VAT
Field		VATAmount	Total VAT Am...	Total VAT
Field		TotalSalesHeader."Amount Including VAT"	Total Amount...	Total Amount Incl. VAT

The structure produces the following layout, for the windows client:



The structure produces the following layout, for the web client:



Where Implemented

The Totals on Sales and Purchase Documents feature is implemented on all core sales and purchase documents: Invoices, quotes, and credit memos, including their posted versions.

The implementation on pages that are not in the 1300 object range is slightly different. For more information, look for the upcoming pattern on the Design Patterns Wiki.

Ideas for Improvement

Implementing of this feature was challenging because some fields must be updated on the header when changing values on the lines (subform). Triggering a page update from the subform only refreshed the subform, not the main page, and that could cause the "Another user has changed..." error when trying to change the values on the document header. The implementation uses decorators and the artifacts explained above to avoid this issue.

The platform has a new built-in support, which is not used it the current implementation: The *UpdatePropagation* property can be added to a part. By setting this property to *Both*, a call to *CurrPage.UPDATE* in the subform will trigger full page update, both the main page and the subpage. For more information about the property, see the Developer and IT Pro Help.

Toggle Financial Setup on Master Data Cards

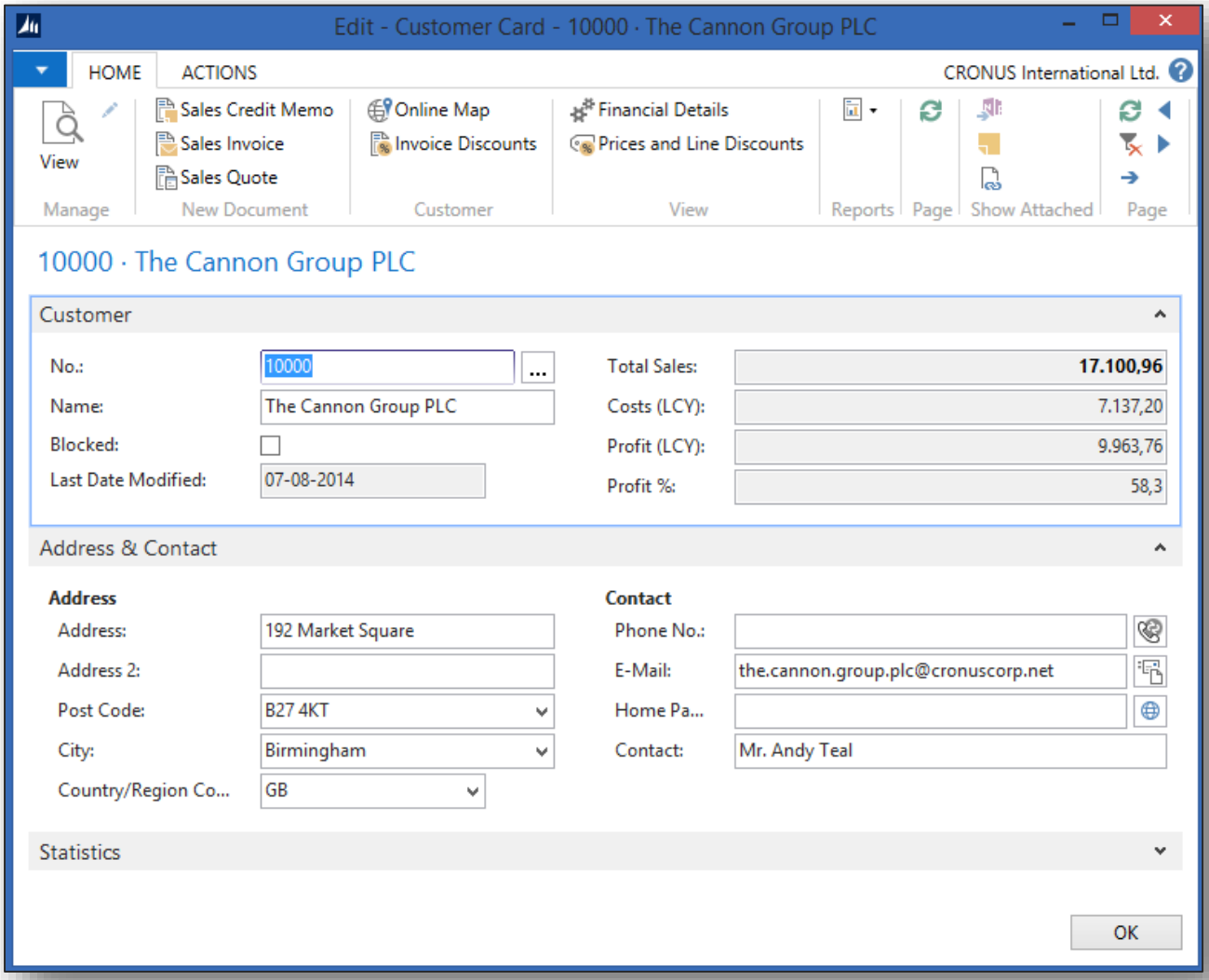
Reasons for Change

Users typically only need to set up financial data for a master data record once, and some of these fields must be filled to be able to post. However, a customer's VAT registration number, payment method, and posting groups practically never change, so once the fields are filled, the user would like to hide them from the daily view of the card.

General Principles

The FastTabs with financial setup fields is shown until the main fields on the FastTabs are filled. When the fields are filled, the user can choose the Financial Details button to show/hide the FastTabs.

Microsoft Dynamics NAV



Implementation

Visibility is controlled by the *ShowMainView* variable. It is assigned to the *Visibility* property of two groups: Invoicing and Payments.

The ShowMainView variable is controlled by the Financial Details button:

```
ShowMainView - OnAction()
ShowMainView := NOT ShowMainView;
```

The ShowMainView variable is also controlled by the *ShouldBeLoadedInSetupView* function, which is called

on *OnOpenPage*. If one of these fields is empty, then the Invoicing and Payments FastTabs are shown):

```
LOCAL ShouldBeLoadedInSetupView() : Boolean
EXIT('' IN [Address,
            "Post Code",
            City,
            "Customer Posting Group",
            "Gen. Bus. Posting Group",
            "VAT Bus. Posting Group"]);
```

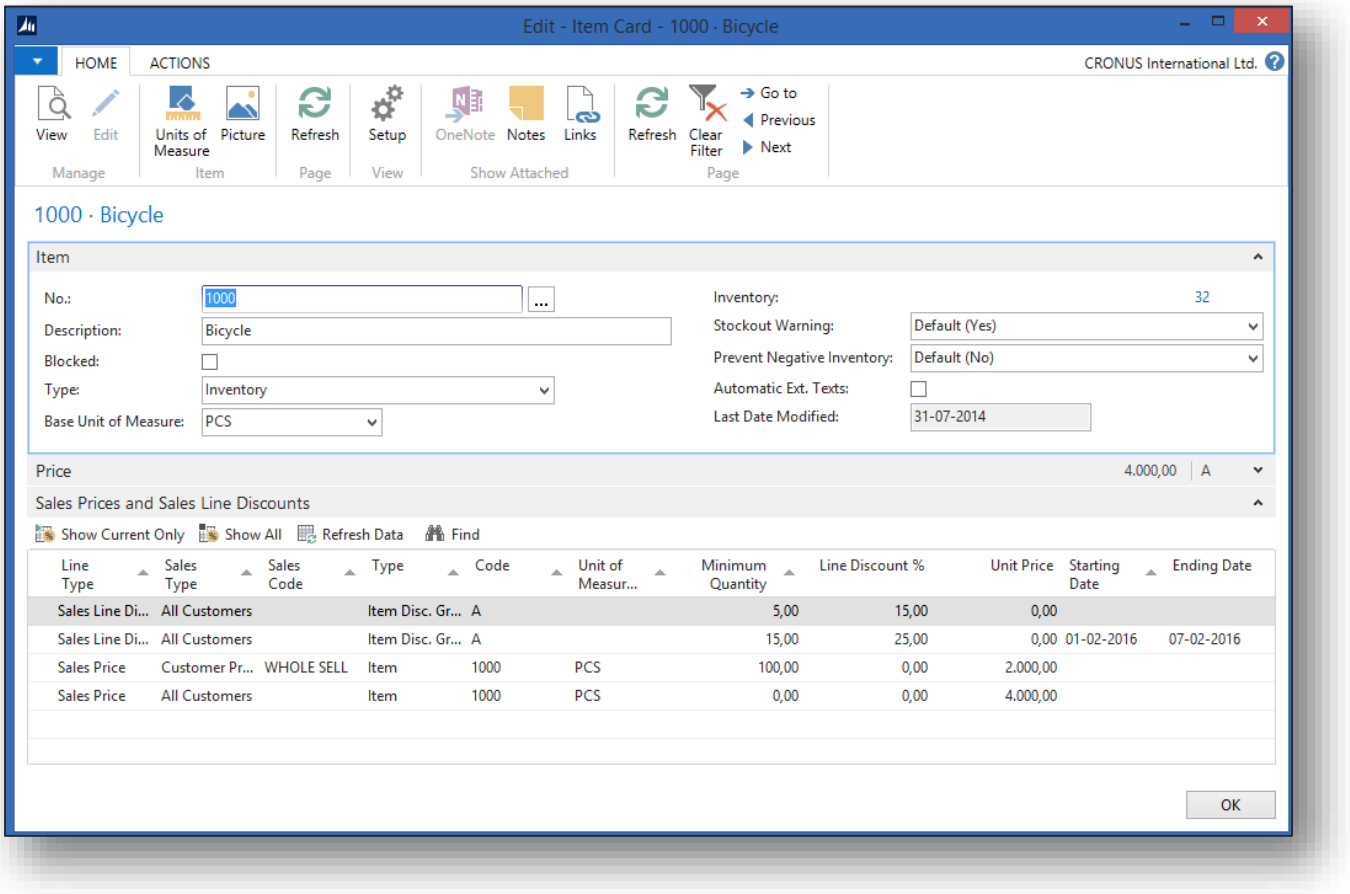
The same pattern is implemented on the item card.

Prices and Line Discounts on Master Data Cards

General Principle

While entering a price or a discount for an item (or a customer), the user should be able to view and edit the item's sales prices and sales line discounts directly on the card.

In Simplified UX, sales prices and sales line discounts are shown together on a FastTab on item and customer cards.



Microsoft Dynamics NAV

Implementation

The Prices and Line Discounts on Master Data Records feature is based on the Presentation Model Pattern. First, the two tables are merged into one:

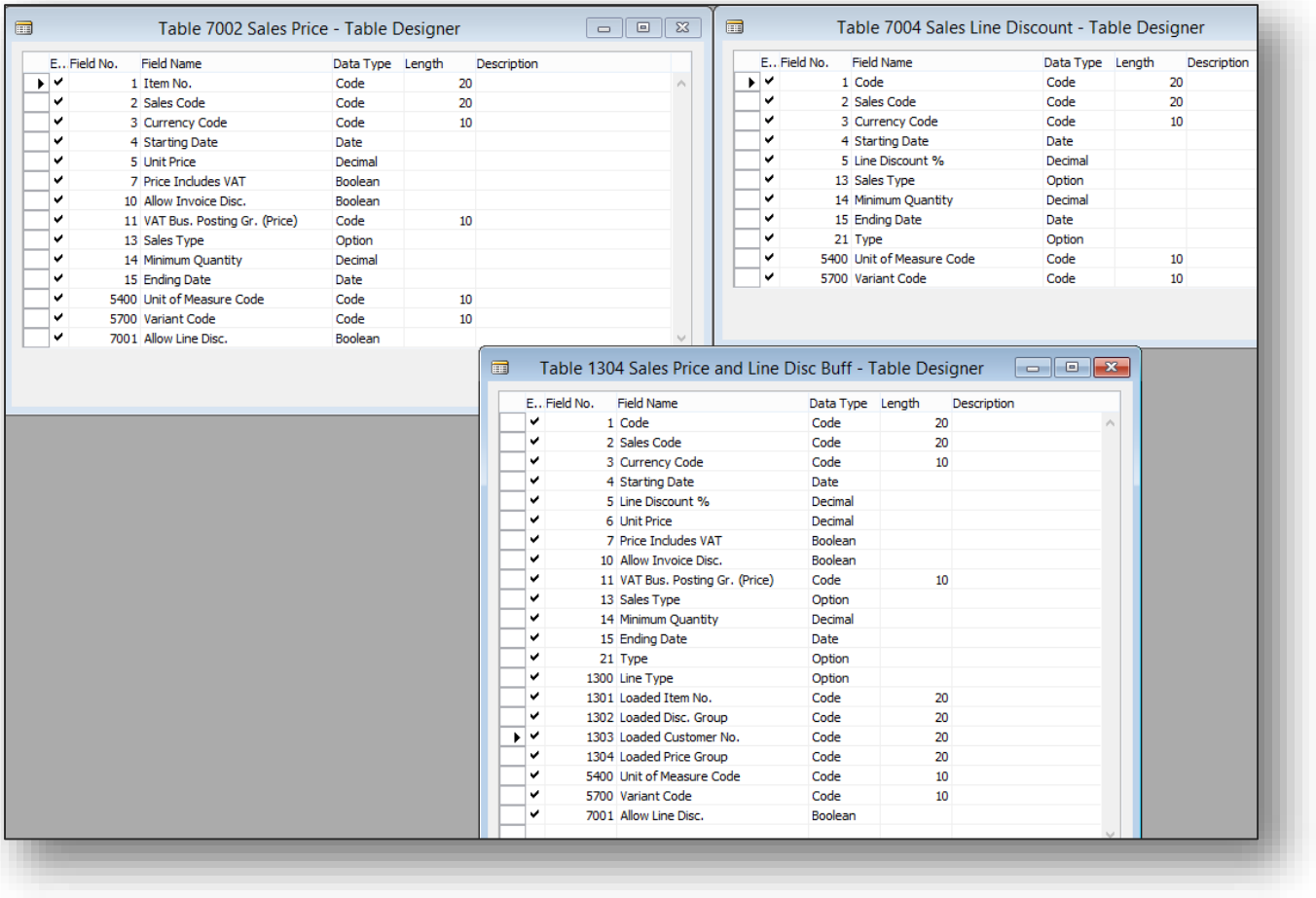


Table 7002 Sales Price - Table Designer

E..	Field No.	Field Name	Data Type	Length	Description
✔	1	Item No.	Code	20	
✔	2	Sales Code	Code	20	
✔	3	Currency Code	Code	10	
✔	4	Starting Date	Date		
✔	5	Unit Price	Decimal		
✔	7	Price Includes VAT	Boolean		
✔	10	Allow Invoice Disc.	Boolean		
✔	11	VAT Bus. Posting Gr. (Price)	Code	10	
✔	13	Sales Type	Option		
✔	14	Minimum Quantity	Decimal		
✔	15	Ending Date	Date		
✔	5400	Unit of Measure Code	Code	10	
✔	5700	Variant Code	Code	10	
✔	7001	Allow Line Disc.	Boolean		

Table 7004 Sales Line Discount - Table Designer

E..	Field No.	Field Name	Data Type	Length	Description
✔	1	Code	Code	20	
✔	2	Sales Code	Code	20	
✔	3	Currency Code	Code	10	
✔	4	Starting Date	Date		
✔	5	Line Discount %	Decimal		
✔	13	Sales Type	Option		
✔	14	Minimum Quantity	Decimal		
✔	15	Ending Date	Date		
✔	21	Type	Option		
✔	5400	Unit of Measure Code	Code	10	
✔	5700	Variant Code	Code	10	

Table 1304 Sales Price and Line Disc Buff - Table Designer

E..	Field No.	Field Name	Data Type	Length	Description
✔	1	Code	Code	20	
✔	2	Sales Code	Code	20	
✔	3	Currency Code	Code	10	
✔	4	Starting Date	Date		
✔	5	Line Discount %	Decimal		
✔	6	Unit Price	Decimal		
✔	7	Price Includes VAT	Boolean		
✔	10	Allow Invoice Disc.	Boolean		
✔	11	VAT Bus. Posting Gr. (Price)	Code	10	
✔	13	Sales Type	Option		
✔	14	Minimum Quantity	Decimal		
✔	15	Ending Date	Date		
✔	21	Type	Option		
✔	1300	Line Type	Option		
✔	1301	Loaded Item No.	Code	20	
✔	1302	Loaded Disc. Group	Code	20	
✔	1303	Loaded Customer No.	Code	20	
✔	1304	Loaded Price Group	Code	20	
✔	5400	Unit of Measure Code	Code	10	
✔	5700	Variant Code	Code	10	
✔	7001	Allow Line Disc.	Boolean		

Fields 13xx are used to identify how the data is loaded.

To load the data, the public *LoadDataForItem* function is called from the *OnAfterGetCurrRecord* trigger on the Mini Item Card page. This provides a set of prices and discounts for the selected item when the user opens the item card or navigates to the card with the Previous/Next actions. In addition, the Refresh Data action is added to the subpage, so the user can always get fresh data.

The *OnModify* trigger is used to update permanent tables for two reasons: 1) Incomplete data must not be published to permanent tables. 2) In small companies with one to three users, it is unlikely that two or more users update an item's or a customer's prices and discounts simultaneously. For these reasons, the solution is simpler and more testable, because the permanent tables are only updated in three places.

To make data update more consistent and easy (the primary key is quite big), data is first deleted and then a new updated record is inserted. Two functions are used to manage this:

- DeleteOldRecordVersion
- InsertNewRecordVersion

These functions are wrappers that identify if prices or discounts were updated. If a line must be deleted or a one inserted, then one of the following functions is called for the Sales Price or Sales Line Discount table:

```
LOCAL DeleteOldRecordVersionFromPrices()
  SalesPrice.GET(
    xRec.Code,
    …
    xRec."Minimum Quantity");

  SalesPrice.DELETE(TRUE);
```

and:

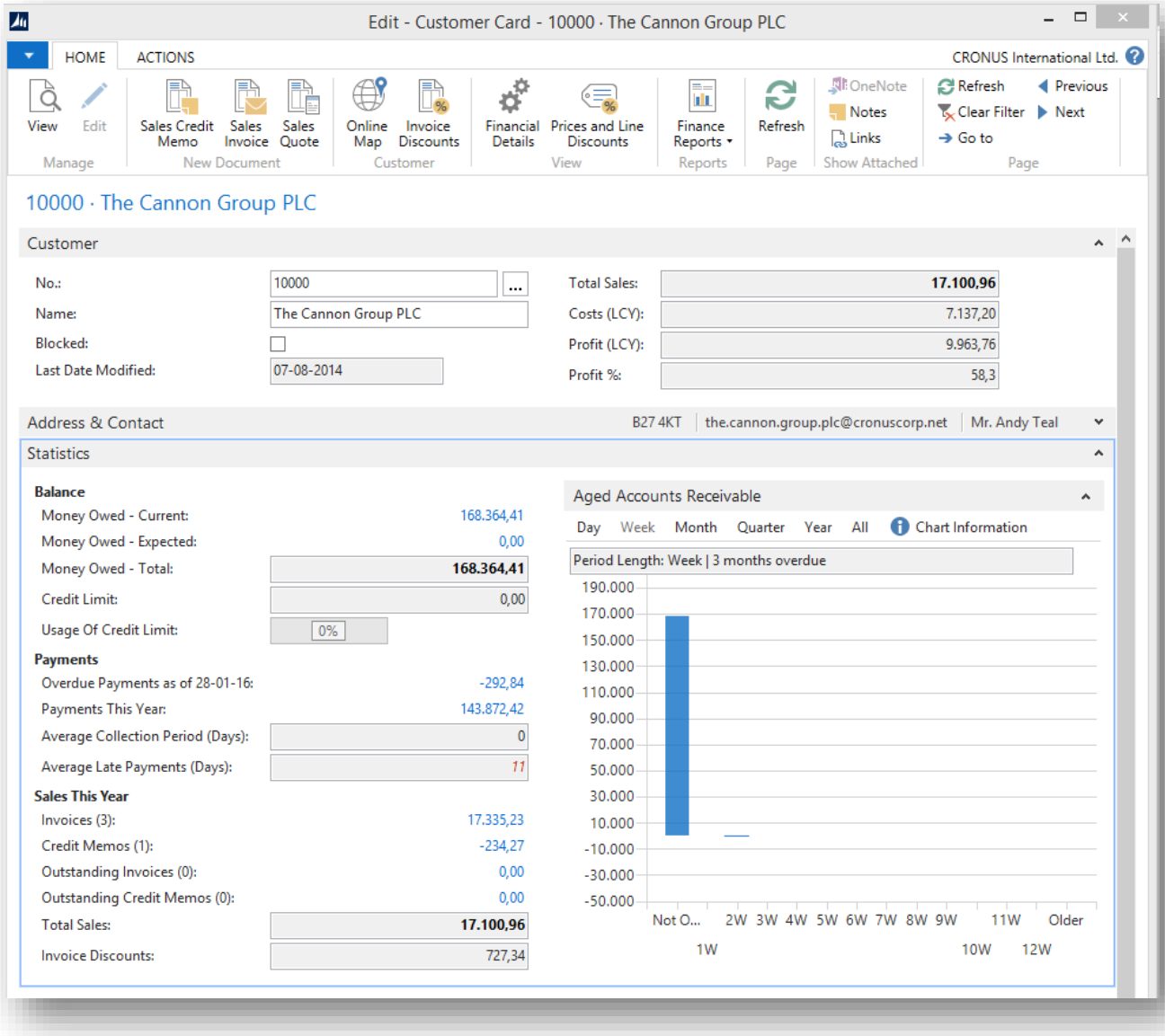
```
LOCAL PROCEDURE InsertNewPriceLine();
  SalesPrice.INIT;
  SalesPrice."Item No." := Code;
  …
  SalesPrice."Price Includes VAT" := "Price Includes VAT";

  SalesPrice.INSERT(TRUE);
```

KPIs on Master Data Cards

Reasons for Change

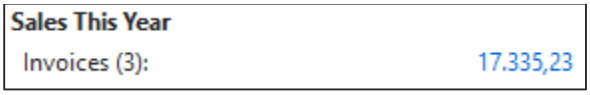
In the standard product, all statistics information for master data records exists on separate page or in FactBoxes. For Simplified UX, selected statistics information is moved to a Statistics FastTab on the customer, vendor, and item cards, so the user has immediate access to it.



Microsoft Dynamics NAV

General Principles

When users wants to see a customer's KPIs, they mainly look for two numbers: Amounts and quantities. For example, they want to know how many invoices were posted and what was the amounts in each invoice. The standard product usually just shows one of these at a time, such as the number of invoices only and the total invoice amount only. In Simplified UX, the two numbers are combined as follows:



To make the information even easier to consume, a chart is included to show the customer's pending payment amount summed for a period that the user can select.

Implementation

Calculation of the chart data is done in COD763, Aged Acc. Receivable. The same codeunit is used in one of the KPI charts on the Small Business Role Center. Calculation of the statistics is done in COD1302, Mini Customer Mgt.. The following functions calculate the values:

```
LOCAL GetTotalSales() : Decimal
NoPostedInvoices := 0;
NoPostedCrMemos := 0;
NoOutstandingInvoices := 0;
NoOutstandingCrMemos := 0;
Totals := 0;

AmountOnPostedInvoices := MiniCustomerMgt.CalcAmountsOnPostedInvoices("No.",NoPostedInvoices);
AmountOnPostedCrMemos := MiniCustomerMgt.CalcAmountsOnPostedCrMemos("No.",NoPostedCrMemos);

AmountOnOutstandingInvoices := MiniCustomerMgt.CalculateAmountsOnUnpostedInvoices("No.",NoOutstandingInvoices);
AmountOnOutstandingCrMemos := MiniCustomerMgt.CalculateAmountsOnUnpostedCrMemos("No.",NoOutstandingCrMemos);

Totals := AmountOnPostedInvoices + AmountOnPostedCrMemos + AmountOnOutstandingInvoices + AmountOnOutstandingCrMemos;

MiniCustomerMgt.CalculateStatistic(
  Rec,
  AdjmtCostLCY,AdjCustProfit,AdjProfitPct,
  CustInvDiscAmountLCY,CustPaymentsLCY,CustSalesLCY,
  CustProfit);
EXIT(Totals)
```

Ideas for Improvement

The statistics FastTab could be hidden with a toggle solution like the one for Financial Details.

Design Patterns

Simplified UX is partly designed using the following NAV design patterns:

- Easy Update of Setup or Supplementary Information
- Create Data from Templates Other
- Creating Custom Charts
- Instructions in the UI

Microsoft Dynamics NAV

Simplified UX – a Foundation to Build Upon

This Design Insights whitepaper described the various parts that make up Simplified UX.
Solutions targeting smaller businesses or verticals can benefit from the same design ideas and principles.
Some of the presented ideas can be applied directly while others need modifications to fit the target.

+ Share

Microsoft Dynamics NAV