MobileNAV
ERP anywhere

MobileNAV Add-On – User Guide

Table of Contents

1 Introduction

MobileNAV is a mobile application based on Dynamics 365 Business Central and Dynamics NAV with fully customizable operation. With MobileNAV it is possible to access and work with Dynamics 365 Business Central and Dynamics NAV via smartphone and tablet either online or offline.

For the successful operation of the product, the following components are needed:

- **MobileNAV application (client)**: installed and used on mobile devices
  MobileNAV is supported on the following mobile platforms: iPhone and iPad (iOS), Android phones and tablets (Android operating system), handheld scanners running on Windows CE and Windows Mobile operating system, and Windows Phone 7, 8 and 10, and Windows 8.1 and 10.

- **Configuration and administration interface**:
  - **MobileNAV Add-on:** provides configuration management and device management capabilities and services related to the MobileNAV application
  - **MobileNAV License:** determines the maximum number of the mobile devices (or pages) for using the application
  - **MobileNAV Components (only for C/AL version):** contribute to the proper operation of the Add-on and the MobileNAV application

- **MobileNAV Configuration**: the configured objects and their settings, specifying the behavior of the MobileNAV application

The MobileNAV application communicates with the server through the standard SOAP web service interface. The MobileNAV Addon provides special interface for authentication, user-based configuration download, and others like GPS information upload. The traditional data upload and modification is done through the published page web services. Configuration modifications are downloaded automatically to the application at each login while modifications made in the application are automatically uploaded to the database immediately in online mode. If the MobileNAV application is in offline mode, modifications are uploaded to the database during synchronization when the application is online again. In the MobileNAV application is in auto online-offline mode, then the app continuously monitoring the connections status with the server:

- In case of active connection, it works like the online mode, so all modifications are uploaded immediately
- In case of no connection, it switches to offline mode, and those parts of the configuration which are configured for offline, those will be accessible in disconnected mode as well. If the device gains back the connection, it will start an automatic synchronization, which uploads the pending modifications, and refreshes offline data

MobileNAV utilizes the standard SOAP web service interface of Dynamics 365 Business Central and Dynamics NAV. That’s why it is able to use the already implemented business logic.

For MobileNAV handling you can use the already published pages of Dynamics 365 Business Central and Dynamics NAV or create new pages with the necessary fields only. The latter option is recommended in order to keep the configuration clear and data traffic low. New pages must be published as web services to be accessible thru the SOAP web service interface, furthermore they need to be configured for MobileNAV to determine the outlook of pages and functioning of the options on the mobile devices.

This User Guide provides support for the administration and configuration of the MobileNAV application presenting the way the modification of certain configuration elements affects the behavior of the MobileNAV application.

2 MobileNAV components

2.1 Server-side components

MobileNAV solution's server-side components are shipped both C/AL and AL version.

2.1.1 C/AL version

The MobileNAV solution consist of the following components in the C/AL version of MobileNAV:

- **Dynamics 365 Business Central and Dynamics NAV license file**: For the proper operation of the solution, the MobileNAV object range must be ordered into the end-user's NAV license through PartnerSource. In case of developer licenses, the license must be updated once access to the granule in VOICE is granted by MultiSoft (after the MobileNAV Partner Contract has been signed).
- **MobileNAV Components**: With the help of this component the Add-on gains license information from the license key (for example, maximum number of active devices, the expiration of the license, etc.). The file is provided by MultiSoft Ltd.
- **MobileNAV Add-On objects**: A file with **.fob** extension containing the objects necessary for the operation of the Add-on. The file is generated by MultiSoft Ltd. (Version list: MNAD,x.x.x.x)
- **MobileNAV License String**: The license key specifies the maximum number of devices (or pages) that can be connected to the database, and in case of demo license, the expiration date. The string is generated by MultiSoft Ltd.
- **MobileNAV Configuration objects**: A file with **.fob** extension containing the objects for the base configuration. These objects are necessary for the proper operation of the business processes in the MobileNAV application. The file is generated by MultiSoft Ltd. (Version list: MNCOx.x.x.x)
- **MobileNAV Configuration Translation files:** Files with **.txt** extension containing the translations for configuration objects or categories separated by languages. These files are generated by MultiSoft Ltd.
- **MobileNAV Configuration XML**: A file with **.xml** extension containing the configured data of the base configuration objects. The file is generated by MultiSoft Ltd.
- **Standard Object Modifications object:** Files with both **.txt** and **.fob** extension containing the standard objects which needs to be modified in order to make the base configuration work properly. You need to merge the customizations made by MultiSoft Ltd. into your database. These files are generated by MultiSoft Ltd. (Version list: MNST,x.x.x.x)

2.1.2 AL version

The MobileNAV solution consist of the following components in the AL version of MobileNAV:

- **Dynamics 365 Business Central and Dynamics NAV license file (only for on-prem installations)**: For the proper operation of the solution, the MobileNAV object range must be ordered into the end-user's NAV license through PartnerSource. In case of developer licenses, the license must be updated once access to the granule in VOICE is granted by MultiSoft (after the MobileNAV Partner Contract has been signed).
- **MobileNAV Extension package**: A file with **.app** extension containing the objects necessary for the operation of the Add-on. The file is generated by MultiSoft Ltd. (Version list: MNAD,x.x.x.x)
- **MobileNAV License String**: The license key specifies the maximum number of devices (or pages) that can be connected to the database, and in case of demo license, the expiration date. The string is generated by MultiSoft Ltd.
- **MobileNAV Configuration XML**: A file with **.xml** extension containing the configured data of the base configuration objects. The file is generated by MultiSoft Ltd.

2.2 Client-side components

- **MobileNAV client app:** The MobileNAV client application can be downloaded from the marketplaces (for example, App Store, Google Play) to your mobile platform.

3 Installing MobileNAV solution

Installing MobileNAV solution consists of these steps:

- Install the MobileNAV objects (as C/AL objects or as AL Extension)
- Finishing the MobileNAV installation with Assisted Setup
- Finishing the MobileNAV installation “manually”
    - Import the MobileNAV License String
    - Import the MobileNAV Configuration XML
    - Setting up the MobileNAV users

3.1 Install the MobileNAV objects

The MobileNAV objects can be installed in C/AL or AL extension version. We have separate installation guide for both.

For more information about the individual steps of installing the MobileNAV solution, see the *MobileNAV Installation Guide*. We provide separate installation guide for AL version and C/AL version.

3.2 Finishing the MobileNAV installation with Assisted Setup

MobileNAV provides Assisted Setup to provide a step-by-step guide for the most important installation steps.

Step 1: MobileNAV License: In the first step the Assisted Setup asks to import the MobileNAV License String. Here you have two options:

- **Start Trial Period:** this option only available with the AL version. With this option you can start a trial period for 30 days with 5 devices.
- **Import License:** here you can import the MobileNAV License String provided by MultiSoft Ltd.

Step 2: MobileNAV Configuration: In this step the Assisted Setup asks to import the Configuration XML. Here you have two options:

- **Import Base Configuration:** the MobileNAV Base Configuration objects includes the proper XML file as well, so if you install the MobileNAV Base Configuration, you don’t need to import the XML file manually.
- **Import Custom Configuration:** this option allows you to import a custom configuration XML file manually. This option is useful, if you already having a custom configuration, and you want to use that instead of the base configuration.

Step 3: MobileNAV Users: In this step the Assisted Setup asks to set up the MobileNAV Users. Important to note that in this step you choose from the already existing Users, so first you need to create Users in Dynamics 365 Business Central or Dynamics NAV. If you have imported the base configuration, then the Assisted Setup offers to choose from simplified profiles like (Sales, Service, etc.), and you need to set up the Login Mode as well. In case of a custom configuration, then the you need to choose from the profiles included in the custom configuration.

3.3 Finishing the MobileNAV installation “manually”

3.3.1 Import the MobileNAV License String

MobileNAV solution requires a valid license string. You can check your license information on the Device Management window of MobileNAV.

Import your MobileNAV License String, as follows:

1. In Dynamics 365 Business Central, in the navigation pane, search for MobileNAV Administration and choose **Devices -> All Devices**
2. On the Device Management window, press **Process -> Change License**
3. Press **Yes** to confirm that you want to change the license
4. Copy and paste the MobileNAV License String you got from your partner or from the MobileNAV Sales Team
5. Press **OK**
6. Check that your MobileNAV License Information changed on the Device Management window

3.3.2 Loading MobileNAV configuration

The MobileNAV solution is shipped with a base configuration, designed to meet the usual user requirements (for example, field sales, field service, etc.). The base configuration can be used as a starting point for a company. In order to work on MobileNAV configuration, it is necessary to load the base configuration with the necessary objects: configuration objects, standard object modifications, extra configuration, and translation objects.

Load the MobileNAV configuration, as follows:

1. Import the MobileNAV configuration .xml file.
   In Dynamics 365 Business Central, search for MobileNAV Administration and choose **Configuration -> Configuration Handling Report**.
   The configuration file contains the extra configuration on top of the configuration objects.
   For more information, see Configuration Handling Report.

2. Import the MobileNAV category translations in Dynamics 365 Business Central.
   MultiSoft provides translations for the base configuration objects in several languages.
   Note: Before importing the translation files, check the language codes and make the necessary preparations. For more information, see Chapter *Import MobileNAV Configuration translations* in the *MobileNAV Installation Guide* and Category Translation Import.

For more information about the individual steps of loading the configuration, see Chapter *Installing MobileNAV solution* in the *MobileNAV Installation Guide*.

3.3.3 Setting up MobileNAV users

The security management of the MobileNAV application is based on the security system of Dynamics 365 Business Central or Dynamics NAV. Therefore, a normal User must be created and configured for the Dynamics 365 Business Central database for each MobileNAV user, because through the application, they are connected to the database and use the content of the database.

Users signing in through the mobile device have the same access rights as if signing in through the Dynamics 365 Business Central or Dynamics NAV client.

Set up MobileNAV users, as follows:

1. Create a user in the Dynamics 365 Business Central or Dynamics NAV: you can create User with Windows Authentication, or Business Central Password Authentication (formerly known as NAVUserPassword), or assign users from Azure Active Directory.
2. Set MobileNAV user properties:
    a. In Dynamics 365 Business Central, search for MobileNAV Administration, and choose **Setup -> Users**.

    b. Press **New** to add the appropriate users from the **User ID** combo box.

    c. Define the appropriate salesperson code in the **Salesperson Code** column in case of a salesperson, or a service resource filter in the **Service Resource Filter** column in case of a service technician, or a location in the **Location Code** column in case of a location assigned to a mobile user.
    This combination provides users of the MobileNAV application with pre-filtered lists so that they can easily access, for example, the list of their customers and invoices.
    If the user is neither a salesperson nor a service technician nor has assigned location, leave these fields empty.

    d. Assign **Profiles** to the mobile user. If your configuration contains Profiles, then you might want to assign one or more Profile to the user. For more information about Profiles, see Profiles.

    e. Select the desired **Login Mode** to meet your requirements:

    - **By Client:** the user can decide whether to use online mode, offline mode, or auto online/offline mode.
    - **Online:** the user is forced to use online mode and cannot switch to offline mode.
    - **Offline:** the user is forced to use offline mode and cannot switch to online mode.
    - **Online/Offline:** the user can switch between online and offline modes but cannot enable auto online/offline mode.
    - **Auto Online/Offline:** the user is forced to use auto online/offline mode and cannot switch to other modes.

- **Strict Auto Online/Offline:** the user is forced to use auto online/offline mode in strict mode and cannot switch to other modes. Strict mode means that online features are not available until all offline changes are successfully uploaded to the server.

f. Turn **Background Synchronization On** or **Off**

g. Select the proper setting for **Connection Status Monitoring Interval**

For more information about user properties, see Users.

After a MobileNAV user is set up and logs in to the MobileNAV application successfully, the Add-on prepares a unique configuration for the user currently logged in. This unique configuration is a subset of the base configuration. There are two filters which will be applied. One filter is the assigned Profiles. Only those pages will be part of the user dependent configuration, where the page’s assigned Profile(s) and user’s assigned Profile(s) match at least in one Profile. If either the user or the page does not have assigned Profile, then the page will be part of the user dependent configuration. The second filter depends on the user’s permissions, which means, that only those pages are included in the configuration, to which the user has at least read-only permission. This applies to the related pages, as well. If the user does not have at least read-only permission to the source table of the related pages, data describing the look-up and drill-down relations will not be included in the user’s configuration. In case of write permission, the insert, update and delete operations are controlled by the configuration settings of the particular page.

For more information about user settings, see the *Online Help of Dynamics 365 Business Central*.

For more information about MobileNAV user settings, see the *MobileNAV Installation Guide*.

4 Procedure of MobileNAV configuration

During the configuration you define the behavior of the MobileNAV application. This includes the following:

- data that should be displayed in the MobileNAV application
- the form in which the data should be displayed
- the relationship between the displayed data
- the kind of operations that can be performed

In the Add-on, you can customize the base configuration according to the customer's needs, or create a totally new configuration.

Note: The configuration is company specific, however, the web services and the page objects are valid for the entire database.

4.1 Configuration effects

For modifying the configuration, a test database is recommended for the following reason: Changes made in the configuration immediately affect the mobile devices that are connected to the database. Therefore, configuration changes should not be carried out in production environment, in order to avoid malfunction of the connected devices in the case of a configuration error, and to avoid increased data traffic and slow login processes (if the configuration is modified, the entire configuration will be downloaded by the MobileNAV application at login).

It is recommended, instead, to use a test database and the associated test Server Service, through which the configuration changes can be tested. All companies must have their own secluded and differentiated configuration in the test database.

When the configuration modifications are done, the following actions must be taken:

1. The configuration objects (pages, reports or codeunits) and other objects related to the customization must be exported from the test database and imported into the production database in the standard way.
2. The configuration XML must be exported from the test database and imported into the production database with the **Configuration Handling Report** dialog under Departments/MobileNAV in Dynamics 365 Business Central. For more information, see Configuration Handling Report.

Configuration for the MobileNAV application includes page editing in Microsoft Dynamics NAV Development Environment, and page publishing and page configuration in Dynamics 365 Business Central (the Add-on configuration page).

4.2 Object range for configuration

The MobileNAV granule contains 300 pages in the 42012822 – 42013181 object range, and it is split into following ranges:

- **42012822 – 42012869:** Page objects for the Add-On.
  Note: Do not insert new Pages into this range.
- **42012870 – 42013049**: Page objects for the Base Configuration.
  Note: Do not insert new Pages into this range.
- **42013050 – 42013099**: **Unused range**, where you can insert the new pages.
- **42013100 – 42013121: Unused range,** kept for Hands-on-Lab exercises. If you do not want to execute the exercises, or you have already executed them, you can use this range for the new pages.

Note: It is recommended to create the new pages into the 42013050 – 42013121 object range, so that upgrade to a newer Add-On version is easier, since this range is kept for your configuration pages.

4.3 Creating and editing pages

During configuration changes, it may be necessary to modify existing pages or create new ones.

Use the Object Designer in Microsoft Dynamics NAV Development Environment to create or edit pages.

1. Open Microsoft Dynamics NAV Development Environment.
2. Select the **Page** category and
   - for a new page, click **New** and create the new page with the help of the New Page wizard.
   - for modifying an existing page, locate the desired page and click **Design**.
     The **Page Designer** window is displayed.
3. Open the **Properties** window from the **View** menu or with the Shift+F4 shortcut from an empty line to edit the page properties.

   The following page properties must be set for MobileNAV configuration:

   - **ID**: The identification number of the page. It must be defined in case of new pages, based on the allocated ranges, in accordance with the customer’s Dynamics 365 Business Central license. See Object range for configuration.
   - **Name**: The name of the page. It is recommended to follow the MobileNAV naming conventions: “MN” prefix + reference to the source of the data to be displayed (e.g.: MN Customer).
   - **CaptionML**: The page name on the desired language to be displayed when the page is opened in the MobileNAV application.
   - **SourceTable**: Data source table displayed on the page.
   - **SourceTableView**: The data of the source table set are displayed by the MobileNAV application based on the defined key, order, and filter. Here you can pre-filter the source table and set the default sorting for the page.

   The other page properties have no influence on the MobileNAV application.

4. Add new fields to the page in the order in which they should be displayed in the MobileNAV application.
   Note: All fields must be added as Field type.

   A page object must definitely include the following fields:

   - all table fields that are part of the primary key of the source table, in the right order
   - the table fields to be displayed in the MobileNAV application
   - in case of lists, the table fields necessary for filtering (see List configuration)
   - in case of cards, the (dummy) fields related to the function appearing in the menu (see Card configuration)
   - in case of reports, the (dummy) fields for entering the input parameters (see Report configuration)

5. Edit the field properties by opening the **Properties** window from the **View** menu or with the Shift+F4 shortcut from the desired field.

   The following field properties must be set for MobileNAV configuration:

   - **Name:** The name of the field. If the field is from the source table, it inherits the field name from the table. The inherited field name can be modified according to your needs.
     Important: For fields with global variable or without SourceExpr, it is mandatory to specify a field name.
   - **SourceExpr**: Typically, a field of the source table, but it can also be a global variable within the page.
     In case of placeholder fields (dummy fields) that do not have any impact on a particular database field, it must be left empty.
   - **Visible**: If it is set to FALSE, the field is not displayed in the MobileNAV application.
     In case of card-type pages, those elements of the key can be hidden that are not relevant for a particular page. In case of list-type pages, those elements can be hidden that will not be used for filtering.
     Note: This property must be set on page level, because it is not inherited from the source table. The default value is TRUE.
     Note: You can also overrule the visible property in the MobileNAV Page Configuration as well, as well as inside a Profile
   - **Editable**: If it is set to FALSE, the field cannot be edited in the MobileNAV application.
     It is interpreted only in case of a Visible field.
     Note: This property must be set on page level, because it is not inherited from the source table. The default value is TRUE.
     Note: You can also overrule the editable property in the MobileNAV Page Configuration as well, as well as inside a Profile
   - **CaptionML**: The field name on the selected language to be displayed when the page is opened in the MobileNAV application (currently English, German, Spanish, French, Dutch, Italian, Greek, Romanian, Croatian, Chinese, Turkish, Portuguese, Swedes, Norvegian,

Japan, Icelandic, Danish, Czech, Slovak, Polish, and Hungarian languages are supported). If the SourceExpr is a field of the source table, it is not necessary to fill out this property, since it is inherited from the source table.

- **OptionCaptionML**: If the SourceExpr is an Option type global variable, the name of the option string ordered to the global variable must be entered in the selected language to be displayed when an option is selected in the MobileNAV application (currently English, German, Spanish, French, Dutch, Italian, Greek, Romanian, Croatian, Chinese, Turkish, Portuguese, Swedes, Norvegian, Japan, Icelandic, Danish, Czech, Slovak, Polish, and Hungarian languages are supported).

4.4 Publishing pages

Newly created pages must be published to the web services to be accessible from the MobileNAV application.

1. In Dynamics 365 Business Central, search for Web Services.
2. Add a new line with **New** from the **Edit** menu or with Ctrl + N.
3. Define the object type, object ID and service name of the new page.
   Note: It is recommended to avoid spaces and special characters for the service name.
4. Enable publishing in the **Publish** column.

4.5 Configuring pages

All created and modified pages must be configured for MobileNAV usage.

1. In Dynamics 365 Business Central, search for MobileNAV Administration.
2. Choose **Configuration** from the MobileNAV menu.
3. Open an existing page configuration from the list or create a new configuration with **New** from the ribbon bar.
   When creating a new configuration, the appropriate object must be selected from the published web services in the Object ID field.
4. Select the page type of new page configurations according to your needs.
   The following page types are available:
   - List and Card: For more information see List and Card configuration.
   - List: For more information, see List configuration.
   - Card: For more information, see Card configuration.
   - Report: For more information, see Report configuration.
   - Offline: For more information, see Offline configuration.
   - Offline Card: For more information, see Offline configuration.

5. Refresh the page configuration to load all changes made in the Page Designer with **Refresh page data** from the **Actions** ribbon bar.
   Note: To refresh all page configurations at once, select **Refresh data for all configured pages** from the MobileNAV menu.

   The refreshed page configurations reflect the changes made in the Page Designer during page editing. If the order of the fields has been modified in the Page Designer, it will be changed the same way in the configuration after the page data refresh, but the already configured connections, conditions, and filters will not be changed. If a field has been deleted in the Page Designer, the field and the associated relations, conditions, and filters will all be deleted from the configuration after the page data refresh.

   Note: The configurable properties of newly added fields will have default values (for example, **MobileType** will be set Normal).

   Note: A particular field of a page can be referenced in the configuration of another page (for example, at the settings of the relations, conditions and filters). In this case, the settings of the corresponding pages must be checked separately.

6. Customize the configuration according to your needs.
   For information on configuring lists, cards and reports, see the respective sections below.

4.5.1 Page configuration elements

In the **Page Configuration** editor you can configure new pages or modify the settings of existing pages.

The following settings are available from the FastTabs:

General FastTab: here you can view and define general settings for the page, such as the identifier, the type and the name of the page, the action that should be taken when the page is opened directly from the Main Menu (create new record or open the first record), synchronization settings in case of offline pages, staging settings, etc.

Field Configuration FastTab: here you can view and modify field settings or modify field order for the page.

For example, you can assign field categories under **Field Category** to frequently used fields to make navigation in the application easier or to handle the visibility settings of fields of the same category at once from the **General Setup**.

You can modify the visibility (**Visible** column) and editability (**Editable** column) of the fields, which come primarily from the underlying page object, even overwriting the original settings. For example, you can decide to hide certain fields from the base configuration, and for this you do not need to modify the page object, but you can change the visibility settings of the chosen fields on the **Field Configuration** FastTab directly.
Note: If a field is not editable on the page object, however, you cannot make it editable in the configuration either.

You can also define whether a field should be displayed on the toolbar of the page in the MobileNAV application (**Display in Menu** column).

For more information on the field configuration on lists, cards and reports, see the respective sections below.

Line Format FastTab: here you can define the data content of the first and second lines of the list item and the header of the card displayed in the MobileNAV application.

For more information on defining the line format, see Specifying line format and Specifying card title – first line format.

Relation FastTab: here you can define connection between pages.
You can associate card pages with lists in order to display more information about the list items. In this case, when selecting an item in a list, the card of the list item opens to show the item's details. For more information, see Defining related card page.
You can associate default drill-down fields with card pages. In this case, when opening the card of a list item, the drill-down relation specified for the field is displayed automatically beside the card. For more information, see Defining default drill-down relation on card.

Behavior FastTab: here you can enable MobileNAV users to edit pages in the MobileNAV application.

For example, you can allow new item creation in a list for MobileNAV users. For more information, see Creating new list item in MobileNAV. Or you can allow content update or content deletion on the card. For more information, see Updating or deleting card content in MobileNAV. In addition, you can set automatic refresh at opening a list page to display all lines of the list, so that the user does not have to refresh the page manually, or display the filter panel automatically instead of the automatic refresh, so that the user can filter list items.

Offline FastTab: here you can define settings related to offline operation and synchronization.
For more information, see Offline configuration.

Visibility FastTab: here you can customize the UI of the MobileNAV application by setting the visibility of toolbar buttons, and you can fully control the title of the list and card pages.

For example, you can turn off the default option "New" for cards which can be opened directly from the Main Menu (**Hide "New" Prefix**), or hide the title text of a page or a parent page on the title bar (**Hide Page Name Prefix on Card**, **Hide Parent Page Title on Drill-Down List**), or show the caption of the source field name in the field caption from where the lookup is opened (**Show Source Field Name**), or you can hide toolbar buttons if not needed by the MobileNAV user.

4.5.2 Assigning menu picture to pages

Pictures make navigation in the MobileNAV application easier. Therefore, it is recommended to assign pictures to all pages.

Note: It is recommended to use transparent 64x64 pixel images with .png extension.

Add picture to a page, as follows:

1. Click **Picture** on the **Actions** ribbon bar and select **Import**.
2. Browse to the appropriate picture.
3. Import the selected picture.

Pictures can also be exported and deleted from the card of a particular page.

The **Menu Picture Version** on the **General** FastTab indicates whether the page has any associated pictures.

4.6 List configuration

In the MobileNAV application, lists present particular data of multiple records at the same time.

For MobileNAV handling, it is recommended to add the least fields possible. The recommended fields are the following:

- all table fields that are part of the primary key of the source table, in the right order (for example, No.)
- the fields used for displaying a list item (for example, Name and City)
- the fields necessary for filtering the list items (for example, Salesperson Code)

This way, you can control the downloaded amount of data (significant especially in case of slower network connection) when updating the list, as only those fields will be downloaded by the application, which are included on the page.

Example: The Item table has several items, which can be accessed by MobileNAV users from the mobile device. If you include five fields (No., Name, Name2, City, and Salesperson Code) on the MN Customer List page, only those five fields will be downloaded by the mobile device for each list item. If you include twenty fields on the page, the mobile device downloads twenty fields, which results in a downloading period four times longer than in the case of the five fields. The card page can contain more fields than the list page, but the fields of the card will only be downloaded, when the user selects a single list item. This optimization is highly recommended for lists and offline pages with many items.

Furthermore it also make sense to filter the page via specifying SourceTableView or programmatic filtering in the OnOpenPage trigger of the page object. In this way you can also limit the number of records being downloaded to the device. This is extremely important for offline pages.

The menu items in the main menu of the MobileNAV application display lists. Through the lists you can navigate to the cards of the particular list items. The look-up and drill-down connections of particular fields of the cards also display other lists.

For initiating a list page configuration, see Configuring pages.

4.6.1 Refreshing the list at opening

All lines of a list can be displayed automatically at opening, avoiding the obligatory use of **Refresh** when no filters are used.

Automatically refresh the content of a list at opening, as follows:

1. In the **Page Configuration** editor of the list page, expand the **Behavior** FastTab.
2. Select the **Auto Refresh On Open** checkbox.

It is recommended to use this option only in case of short lists to avoid slow data traffic. When dynamic filters are defined in the configuration, for example, on a drill-down, this option is not interpreted. In this case, the lines to which the filters apply are displayed automatically at opening of the list.

4.6.2 Specifying line format

The line format of a page displayed in the MobileNAV application can be set on the **Line Format** FastTab.

The **First Line Format** provides the data content of the first line of the list item in the MobileNAV application.

The **Second Line Format** provides the data content of the second line of the list item in the MobileNAV application.

Define the content of the lines manually or with the **Line Format Configuration** window, as follows:

1. Open the **Line Format Configuration** window with the AssistEdit button.
2. In the **Line Format Configuration** window, click on an empty row.
3. Define the line format from the following three elements (**Line Types**):
    - Caption of field ([]): For this line type, select the field ID, the caption of which to be displayed in the first or second line of a list item in the MobileNAV application.
      The identifier of the fields corresponds to the name of the field list of the page.
    - Value of field ({}): For this line type, select the field ID, the value of which to be displayed in the first or second line of a list item in the MobileNAV application.
      The identifier of the fields corresponds to the name of the field list of the page.
    - Separator: For this line type, enter free text, for eg. ":", "-" or space, in the **Separator** column.

    Note: The length of the first and second lines is limited to 100 characters.

Example: On the Customer list page, the first field is No., the second field is Name, and third field is Name 2, and the fourth field is City.

First Line Format: {Name} {Name 2}

where the content of field *Name* is **John** and the content of *Name 2* is **Doe**

Second Line Format: [City]: {City}

where the content of field *City* is **Budapest**

The result of this example in the MobileNAV application is the following:

John Doe

City: Budapest

Note: When using a translated version, the appropriate translated field name is used for the *City* field (for example, ”*Stadt*” in German). The separators are used uniformly in all languages, translated versions are not available.

4.6.3 Defining related card page

The connection between pages can be set on the **Relation** FastTab.

Here you can associate card pages with lists in order to display more information about the list items. In this case, when selecting an item in a list, the card of the list item opens to show the item’s details.

To define a card page relation to a list, define the appropriate card page in the **Related Page Name** combo box. Information found on the card page will be displayed when selecting a list item.

Note: Only those configured cards can be selected as related page whose source table is the same as the source table of the list.

Important: The primary key fields of the source table must be provided both on the list and card pages in the same order and with the same field name. Otherwise, the MobileNAV application cannot open the appropriate card of the list item.

4.6.4 Creating new list item in MobileNAV

You can enable MobileNAV users to create new items in a list from their mobile devices.

To permit new item creation in a list for MobileNAV users, the following requirements must be met:

- Supply the MobileNAV user with write permission to the source table of the page, based on the Dynamics 365 Business Central security model.
- Allow insertion in the configuration of the page by enabling **Page Insert** on the **Behavior** FastTab.

4.6.5 Making lists sortable and/or enable FlowFilters

List pages can be made sortable and contain FlowFilters in the MobileNAV application. Sorting makes it easier to find the proper record in the list. FlowFilters can be used to dynamically control how the FlowFields are calculated.

Make a page sortable or enable FlowFilters in the MobileNAV application, as follows:

1. Open Microsoft Dynamics NAV Development Environment.
2. Open the list page that you want to make sortable with the **Design** button in the **Page** category.
3. Open the **C/AL Editor** in the Page Designer from the **View** menu or with F9.
4. Find the **OnOpenPage** trigger.
5. Open the **C/AL Locals** from the **View** menu.
6. Create a new variable on the Variables tab with the **MobileNAVSortingFunctions** name, **Codeunit** data type, and **MobileNAV Sorting Functions** subtype and close the **C/AL Locals**.

7. For the **OnOpenPage** trigger enter the following code in the **C/AL Editor**:

```
// MobileNAV: apply user specific sorting or flowfilters

SETVIEW(MobileNAVSortingFunctions.GetUserView(CurrPage.OBJECTID(FALSE), GETVIEW(TRUE)));
```

8. Save and compile the page.

With this technique the sorting and the applied FlowFilters of the page will be user specific, the selected key of sorting, the sorting order and the specified FlowFilters will be automatically stored and applied for the user both in online and in offline mode.

Add FlowFilters on a page in the MobileNAV configuration, as follows:

1. Open Dynamics 365 Business Central.
2. Find and edit the page to which you want to add FlowFilters.
3. Press the **Add/Remove FlowFilters.**
4. Select the desired FlowFilter in the Field No. field.

4.6.6 Field configuration on list page

The fields of the list can be configured on the **Field Configuration** FastTab. Only the Mobile Type and the Related Pages field properties are relevant for list pages.

If a list has too many fields, you can create a filter, to display only the necessary fields with the **Filter** button on the toolbar of the **Field Configuration** FastTab. To clear the filter, click the **Clear Filter** button.

4.6.6.1 Defining Mobile Type

With the **Mobile Type** property you can define a special operation that is performed by the MobileNAV application in the current field on the mobile device.

Choose from the following options for list pages:

- Normal: No special operation. This is the default value.
- Barcode: Barcode scanning will be available for the actual field. The scanned barcode will be used as filter for the field, and after scanning the list will be refreshed with the new filter.
  In case of multiple fields with barcode mobile type, the MobileNAV application asks the user for which field to apply the barcode filter.
- SalesPersonCode: If a salesperson code is set to the user, the MobileNAV application provides the „Own items only” quick-filter that can be turned on and off conveniently. The „Own items only” quick-filter shows only the own items depending on the user’s assigned Salesperson Code.
  For more information, see Users.
- ServiceResourceFilter: If a service resource filter is set to the user, the MobileNAV application provides the „Own items only” quick-filter that can be turned on and off conveniently. The „Own items only” quick-filter shows only the own items depending on the user’s assigned ServiceResourceFilter.
  For more information, see Users.

- LocationCode: If a location code is set to the user, the MobileNAV application provides the „Own items only" quick-filter that can be turned on and off conveniently. The „Own items only" quick-filter shows only the own items depending on the user's assigned Location Code.
  For more information, see Users.
- UserID: the MobileNAV application provides the „Own items only" quick-filter that can be turned on and off conveniently. The „Own items only" quick-filter shows only the own items depending on the user's identifier.
  For more information, see Users.
- CacheableImage: The Blob type field is interpreted as an image. Upload, taking a picture and upload, saving and deletion of the image are possible, or in case of function execution, the result of the function is displayed as an image. The CacheableImage-type image will be cached by the MobileNAV client app, and the list will also display it.

4.6.6.2 Defining look-up relation

With the **Related Pages** property you can configure the relationship of other lists available from a particular field of the current page. In case of list-type fields, connection can be configured in the following cases:

- to support filtering conditions
  For example, MobileNAV users can select a customer from the customer list, for filtering on the customer code in the sales orders list, instead of entering the code manually.
- to display the names or descriptions of the codes instead of the code itself
  For example, on the sales order list, the salesperson name is displayed in the MobileNAV application, instead of the salesperson code. This approach can also be used in case of the currency code, the payment method code, and similar fields making the data displayed by the list more informative.
  Note: In this case, it is recommended to set the related list (list of salesperson) to Offline, so that the MobileNAV application downloads the content of the list once, and retrieves the proper description value for each list item without downloading it from the server again. Otherwise, the MobileNAV application retrieves the related record in case of each list item, generating extra data traffic, which results in slow operation.

Define a look-up relation to a field of the current page as follows:

1. Select the appropriate field and open the **Relation Configuration** window with the **Relations Setup** button from the toolbar of the **Field Configuration** FastTab.
   You can also open the **Relation Configuration** window from the **Related Pages** column of the appropriate field.

2. Select **New** from the ribbon bar to create a new row.

3. Define the name of the related page in the **Related Page Name** column.

4. Define the code of the appropriate field of the related page in the **Related Page Code Field Name** column.

   This field of the related page is set for the current field, when selecting a related record from the related page.

5. Define the description of the appropriate field of the related page in the **Related Page Desc. Field Name** column.

   The value of this field is used for displaying the current field in the list items and the card as well.

   Note: If you do not want to configure a different display value for the field, specify the same name as in the **Related Page Code Field Name** column.

6. Define filters for the related page, as follows:

   a. Open the **Filter Configuration** window from the **Filters** column to define filter conditions for the list defined in the relation.

   b. Define the name of the appropriate field of the related page in the **Dest. Field Name** column.

   c. Define the filter type in the **Filter Type** column (FIELD, FILTER, CONST).

   d. Define the type of comparison in the **Filter Comparison Type** column.

   e. Provide the source field name or the filter value:

      i. In case of the FIELD filter type, provide the field name of the currently configured page in the **Source Field Name** column.

      ii. In case of the FILTER or CONST filter types, provide a textual filter condition in the **Filter Value** column.

   With this approach, you can set dynamic filters on the related page depending on the fields of the current record. For example, you can dynamically filter the Fault Codes depending on the Fault Area Code of the actual Service Item card.

   Note: Static filters in the SourceTableView property of the page are also taken into account.

   Extra filtering can be defined in the **OnOpenPage** trigger of the page.

7. Define conditions for the related page, as follows:

   a. Open the **Condition Configuration** window from the **Conditions** column to define conditions for the list defined in the relation.
   b. Define the field name of the currently configured page in the **Source Field Name** column.
   c. Provide a textual filter condition in the **Filter Value** column.

   Note: With this approach, you can define conditions, which must be satisfied in order to apply the relation. For example, if the Type of the sales order line is Item, the relation must be applied to the Item list, and if the Type of the sales order line is Resource, the relation must be applied to the Resource list.

When the conditions are satisfied, the page defined in the **Related Page Name** column is available. When selecting the field with the related page in the MobileNAV application, data corresponding to the defined filters is displayed. This data is the value of the field defined in the **Related Page Code Field Name** and **Related Page Desc. Field Name** columns.

4.7 Card configuration

In the MobileNAV application, cards present multiple data fields of a single record at a time. When a list item is selected, its card page opens with more information about the item. Furthermore, from the fields of the card other lists can be opened if look-up or drill-down connections have been defined for the particular fields.

For MobileNAV handling, it is recommended to add the least fields possible, to avoid slow data download (especially in case of slower network connection) when updating the card. The recommended fields are the following:

- all table fields that are part of the primary key of the source table, in the right order
- the fields to be displayed and/or edited on the mobile device
- the placeholder/dummy fields for toolbar buttons on the mobile device

For initiating a card page configuration, see Configuring pages.

4.7.1 Specifying card title – first line format

The line format of a page displayed in the MobileNAV application can be set on the **Line Format** FastTab.

The **First Line Format** provides the header of the card page in the MobileNAV application. The **Second Line Format** is not available for card pages.

Define the content of the line manually or with the **Line Format Configuration** window, as follows:

1. Open the **Line Format Configuration** window with the AssistEdit button.
2. In the **Line Format Configuration** window, click on an empty row.
3. Define the line format from the following three elements (**Line Types**):
    - Caption of field ([]): For this line type, select the field ID, the caption of which to be displayed in the header of the card in the MobileNAV application.
      The identifier of the fields corresponds to the name of the field list of the page.
    - Value of field ({}): For this line type, select the field ID, the value of which to be displayed in the header of the card in the MobileNAV application.
      The identifier of the fields corresponds to the name of the field list of the page.
    - Separator: For this line type, enter free text, for eg. ":", "-" or space, in the **Separator** column.

    Note: The length of the line is limited to 100 characters.

Example: On a Customer card page, the first field of the current page is No., the second filed is Name, and the third field is Name 2.

First Line Format: {Name} {Name 2}

where the content of field *Name* is **John** and the content of *Name 2* is **Doe**

The result of this example in the MobileNAV application is the following:

John Doe

4.7.2 Defining default drill-down relation on card

The connection between pages can be set on the **Relation** FastTab.

Here you can associate default drill-down fields with card pages. In this case, when opening the card of a list item, the drill-down relation specified for the field is displayed automatically beside the card.

Note: Only those fields can be selected as default drill-down which have drill-down relation configured.

Define a default drill-down page to a card as follows:

1. Open the **Field List** window in the **Default DrillDown** combo box.
2. Select the appropriate drill-down field from the list.
3. Enable or disable Default DrillDown on Tablets only to following way:
    a. Enabled: means that the Default DrillDown will be executed on tablets only.
    b. Disabled: means that the Default DrillDown will be executed on tablets and smartphones as well.

4.7.3 Updating or deleting card content in MobileNAV

You can enable MobileNAV users to update or delete the content of a card from their mobile devices.

To permit content update on the card for MobileNAV users, enable **Page Update** on the **Behavior** FastTab.

To permit content deletion on the card for MobileNAV users, enable **Page Delete** on the **Behavior** FastTab.

To edit the fields of a card, however, the following requirements must also be met:

- Supply the MobileNAV user with write permission to the source table of the page, based on the Dynamics 365 Business Central security model.
  For more information, see Setting up MobileNAV users.
- Allow editing of the individual fields by setting the **Editable** property of the page objects to TRUE in the Page Designer in Microsoft Dynamics NAV Development Environment.

4.7.4 Enable FlowFilters

Cards can contain FlowFilters in the MobileNAV application. FlowFilters can be used to dynamically control how the FlowFields are calculated.

Enable FlowFilters on a page in the MobileNAV application, as follows:

1. Open Microsoft Dynamics NAV Development Environment.
2. Open the card page that you want to make sortable with the **Design** button in the **Page** category.
3. Open the **C/AL Editor** in the Page Designer from the **View** menu or with F9.
4. Find the **OnOpenPage** trigger.

5. Open the **C/AL Locals** from the **View** menu.

6. Create a new variable on the Variables tab with the **MobileNAVSortingFunctions** name, **Codeunit** data type, and **MobileNAV Sorting Functions** subtype and close the **C/AL Locals**.

7. For the **OnOpenPage** trigger enter the following code in the **C/AL Editor**:

```
// MobileNAV: apply user specific sorting or flowfilters

SETVIEW(MobileNAVSortingFunctions.GetUserView(CurrPage.OBJECTID(FALSE), GETVIEW(TRUE)));
```

8. Save and compile the page.

Add FlowFilters on a page in the MobileNAV configuration, as follows:

1. Open Dynamics 365 Business Central.

2. Find and edit the page to which you want to add FlowFilters

3. Press the **Add/Remove FlowFilters**

4. Select the desired FlowFilter in the Field No. field

With this technique the applied FlowFilters of the page will be user specific, the specified FlowFilters will be automatically stored and applied for the user both in online and in offline mode.

4.7.5 Field configuration on card page

The fields of the card can be configured on the **Field Configuration** FastTab.

If a card has too many fields, you can create a filter, to display only the necessary fields with the **Filter** button on the toolbar of the **Field Configuration** FastTab. To clear the filter, click the **Clear Filter** button.

You can modify the visibility and editability of the fields, which come primarily from the underlying page object, but here you can overwrite the original settings. For example, to hide certain fields from the base configuration, you do not need to modify the page object, you can also do this on the **Field Configuration** FastTab directly.

Note: If a field is not editable on the page object, you cannot make it editable in the configuration.

4.7.5.1 Defining Mobile Type

With the **Mobile Type** property you can define a special operation that is performed by the MobileNAV application in the current field on the mobile device.

Choose from the following options for card pages:

- Normal: No special operation. This is the default value.
- Image: The Blob type field is interpreted as an image. Upload, download and delete of the image are possible, or in case of function execution, the result of the function is displayed as an image. For detailed instructions, see Handling BLOB-type table fields.
- PDF: The Blob type field is interpreted as a PDF file. Upload, download and delete of the file are possible, or in case of function execution, the result of the function is displayed as a PDF file (for

example, report or document generation from card, like sales order confirmation).
For detailed instructions, see Handling BLOB-type table fields and Defining function execution.

- Hyperlink: The specified link can be opened via a browser.
- Barcode: The value of the field can be modified with barcode scanner.
- Email: E-mail can be sent to the addressees.
- PhoneNumber: The phone number can be called.
- Address1, Address2, AddressCountry, AddressRegion, AddressCity, AddressPostCode: Map view or navigation can be started for the addresses defined in fields with data types starting with „Address”.
  The MobileTypes are concatenated together for the search address in the following order: {AddressCountry} {AddressRegion} {AddressPostCode} {AddressCity} {Address1} {Address2}
- Refresh: Data of the specific card page will be refreshed after executing the configured function.
- Close: The specific page will be closed after executing the configured function.
- PhoneNumberMobile: The mobile phone number can be called.
- Signature: The Blob field is interpreted as an image. Uploading, downloading and deleting the signature is possible.
  The signature can be corrected by selecting Upload.
  For detailed instructions, see Handling BLOB-type table fields.
- ContactName: Address book export or import function can be started from cards with this field type.
  In case of export, an address book entry will be created. In case of import, the data of the address book is read into the fields with the appropriate mobile data type. Fields beginning with the following mobile data types are used: “ContactName”, “Link”, “Email”, “PhoneNumber”, “PhoneNumberMobile” and “Address”.
- Quantity: if a numeric card field has this value, then the MobileNAV client app will display a + and – button to increase or decrease the value conveniently.
- Excel: The Blob type field is interpreted as an EXCEL file. Upload, download and delete of the file are possible, or in case of function execution, the result of the function is displayed as an Excel file (for example, inventory valuation report generation in Excel format instead of PDF).
  For detailed instructions, see Handling BLOB-type table fields and Defining function execution.
- FieldControl: the MobileNAV client app will process the content of this field as a special operation set. With FieldControl you can control certain field properties or send operations to the MobileNAV client app, like hiding certain field or making a field read-only, or opening a field for editing, or closing the card. The content of the FieldControl can be controlled from C/AL code depending on custom business logic. For more information, see Handling dynamic field properties.
- CacheableImage: The Blob type field is interpreted as an image. Upload, download and delete of the image are possible, or in case of function execution, the result of the function is displayed as an image. The CacheableImage type image will be cached by the MobileNAV client app, and the card will display it. In case of multiple CacheableImage BLOB type on the same card, the MobileNAV client app will display only one Image viewer on the card, and all the images can be browsed in the same viewer.
  For detailed instructions, see Handling BLOB-type table fields.

- File: The Blob type field is interpreted as a custom file. Upload, download and delete of the file are possible.
  For detailed instructions, see Handling BLOB-type table fields.
- MultilineText: in case of TEXT fields the MobileNAV client app will display the field in multiple lines
- CacheablePDF: The Blob type field is interpreted as a PDF file. Upload, saving and deletion of the PDF are possible, or in case of function execution, the result of the function is displayed as a PDF. The CacheablePDF-type PDF will be cached by the MobileNAV client app, and the list will also display it.
- CacheableFile: The Blob type field is interpreted as a file. Upload, saving and deletion of the file are possible, or in case of function execution, the result of the function is displayed as a file. The CacheableFile-type file will be cached by the MobileNAV client app, and the list will also display it.
  For detailed instructions, see Handling BLOB-type table fields.
- CacheableExcel: The Blob type field is interpreted as an Excel file. Upload, saving and deletion of the Excel are possible, or in case of function execution, the result of the function is displayed as an Excel. The CacheableExcel-type Excel will be cached by the MobileNAV client app, and the list will also display it.

4.7.5.2 Defining look-up relation

With the **Related Pages** property you can configure the relationship of other lists available from a particular field of the current page.

The look-up relation is interpreted on normal fields. With the help of a look-up relation, MobileNAV users can select the value of a particular field from the list defined in the relation and do not have to enter the correct value manually.

Example: The value of the salesperson code field can be selected from the salesperson list.

Define a look-up relation to a field of the current page, as follows:

1. Select the appropriate field and open the **Relation Configuration** window with the **Relations Setup** button from the toolbar of the **Field Configuration** FastTab.
   You can also open the **Relation Configuration** window from the **Related Pages** column of the appropriate field.

2. Create a new row with **New** from the ribbon bar.

3. Define the name of the related page in the **Related Page Name** column.

4. Define the code of the appropriate field of the related page in the **Related Page Code Field Name** column.

   This field of the related page is set for the current field, when selecting a related record from the related page.

5. Define the description of the appropriate field of the related page in the **Related Page Desc. Field Name** column.

   The value of this field is used for displaying the current field in the list items and the card as well.

Note: If you do not want to configure a different display value for the field, specify the same name as in the **Related Page Code Field Name** column.

6. Define filters for the related page, as follows:

    a. Open the **Filter Configuration** window from the **Filters** column to define filter conditions for the list defined in the relation.

    b. Define the name of the appropriate field of the related page in the **Dest. Field Name** column.

    c. Define the filter type in the **Filter Type** column (FIELD, FILTER, CONST).

    d. Define the type of comparison in the **Filter Comparison Type** column.

    e. Provide the source field name or the filter value:

        i. In case of the FIELD filter type, provide the field name of the currently configured page in the **Source Field Name** column.

        ii. In case of the FILTER or CONST filter types, provide a textual filter condition in the **Filter Value** column.

    With this approach, you can set dynamic filters on the related page depending on the fields of the current record. For example, you can dynamically filter the Fault Codes depending on the Fault Area Code of the actual Service Item card.

    Note: Static filters in the SourceTableView property of the page are also taken into account.

    Extra filtering can be defined in the **OnOpenPage** trigger of the page.

7. Define conditions for the related page, as follows:

    a. Open the **Condition Configuration** window from the **Conditions** column to define conditions for the list defined in the relation.
    b. Define the field name of the currently configured page in the **Source Field Name** column.
    c. Provide a textual filter condition in the **Filter Value** column.

    With this approach, you can define conditions, which must be satisfied in order to apply the relation. For example, if the Type of the sales order line is Item, the relation must be applied to the Item list, and if the Type of the sales order line is Resource, the relation must be applied to the Resource list.

When the conditions are satisfied, the page defined in the **Related Page Name** column is available. When selecting the field with the related page in the MobileNAV application, data corresponding to the defined filters is displayed. This data is the value of the field defined in the **Related Page Code Field Name** and **Related Page Desc. Field Name** columns.

4.7.5.3 Defining drill-down relation

With the **Related Pages** property you can configure the relationship of other lists available from a particular field of the current page.

The drill-down relation is interpreted on fields of FlowField type or on dummy fields. In the MobileNAV application, the list defined in the relation opens with the defined filter conditions when selecting a field or a menu item.
Example 1: MobileNAV users can “drill-down” into the balance of the customer and open the open ledger entries of the current customer in a list.
Example 2: MobileNAV users can open the sales orders of the current customer in a list when pressing the Orders dummy button on the toolbar of the customer card.

Define a drill-down relation to a field of the current page, as follows:

1. Select the appropriate field and open the **Relation Configuration** window with the **Relations Setup** button from the toolbar of the **Field Configuration** FastTab.
   You can also open the **Relation Configuration** window from the **Related Pages** column of the appropriate field.
2. Define the name of the related page in the **Related Page Name** column.
3. Define filters for the related page, as follows:
   a. Open the **Filter Configuration** window from the **Filters** column to define filter conditions for the list defined in the relation.
   b. Define the name of the appropriate field of the related page in the **Dest. Field Name** column.
   c. Define the filter type in the **Filter Type** column (FIELD, FILTER, CONST).
   d. Define the type of comparison in the **Filter Comparison Type** column.
   e. Provide the source field name or the filter value:
      i. In case of the FIELD filter type, provide the field name of the currently configured page in the **Source Field Name** column.
      ii. In case of the FILTER or CONST filter types, provide a textual filter condition in the **Filter Value** column.

   With this approach, you can set dynamic filters on the related page depending on the fields of the current record. For example, you can dynamically filter the Fault Codes depending on the Fault Area Code of the actual Service Item card.

   Note: Static filters in the SourceTableView property of the page are also taken into account.

   Extra filtering can be defined in the **OnOpenPage** trigger of the page.
4. Define conditions for the related page, as follows:
   a. Open the **Condition Configuration** window from the **Conditions** column to define conditions for the list defined in the relation.
   b. Define the field name of the currently configured page in the **Source Field Name** column.
   c. Provide a textual filter condition in the **Filter Value** column.

With this approach, you can define conditions, which must be satisfied in order to apply the relation. For example, if the Type of the sales order line is Item, the relation must be applied to the Item list, and if the Type of the sales order line is Resource, the relation must be applied to the Resource list.

4.7.5.4 Defining mandatory fields for insert

On card pages you can make certain fields mandatory for insertion. In this case, the MobileNAV application creates the new record on the server only if all the fields required for the insertion are filled out.

Example: You make the Type and Salesperson Code fields mandatory for insert on the To-do card. As a result, the user must fill out the Type and Salesperson Code fields of the To-do card in order to be able to create the card on the NAV server.

To define a field as a mandatory field for insert, select the **Required for Insert** checkbox for the appropriate field on the card.

4.7.5.5 Displaying fields in the toolbar of the card

Fields can be displayed in the toolbar of the card for easier accessibility. It is recommended to configure the dummy buttons (with drill-down relations or function execution) or BLOB type fields (for example, Signature or Picture) as toolbar buttons.
Enable displaying fields in the toolbar, as follows:

1. Select the **Display In Menu** checkbox in the appropriate field.
2. Choose the appropriate category in the **Category** column of the appropriate field.
   This way, the toolbar buttons are grouped under a category toolbar button, and all fields with the same category will be available as sub-menu items.

   Example: You configure the Quotes, Orders and Return orders dummy buttons into the DOCUMENTS category. As a result, the toolbar contains a Documents button, and if the user presses the button, a sub-menu opens, where the proper operation can be selected. The order of the sub-menu items is taken from the order on the page.

   Note: It is recommended to group toolbar buttons, in order to avoid an overcrowded user interface, when a card has too many toolbar items.

   For more information on category management, see Categories.

4.7.5.6 Defining function execution

The MobileNAV solution allows function execution to be invoked from the mobile application. In this case, via pressing a button in the mobile application, a function is executed on the NAV server and the function receives the current record as input parameter. With this approach, not only data display and modification is possible with the MobileNAV solution, but any kind of NAV business logic can be executed, for example, posting orders, generating reports or documents, approving, rejecting or delegating approval entries, etc.

Function execution requires the web service extension mechanism of Dynamics 365 Business Central. The web service extension is ensured by the following conditions:

- A codeunit must be recorded as web service with the same service name as the related card page.
- The Published flag for this codeunit must be unchecked in the Web Services table.
- The first parameter of the functions must have Key name and Record type and must be identical with the source table of the page.
- In case of Mobile Type PDF, Excel, Image and File (and its cacheable version):
  - The second parameter of the functions must be Base64Result name and BigText type. This will contain the Base64 encoded file content
  - The third (optional) parameter of the functions must be FileName name and Text type. This can overwrite the default filename associated by MobileNAV

As a result, the page is extended with the functions of the codeunit, where the first parameter is the same record type as the source table of the page, so when the user presses the function button in the MobileNAV application, it will invoke this function, and the input parameter will be filled out with the proper record.

Create and configure functions, as follows:

1. Record a new function into the MN Page Functions program module.
   This codeunit includes functions associated with different pages. You can use this codeunit to add your custom functions, or create a separate one (even in MobileNAV object range) to make merging of the new base configuration easier in the future.

   a. Open Microsoft Dynamics NAV Development Environment.

   b. Select the **Codeunit** category.

   c. Locate the MN Page Functions codeunit and click **Design**.
      The C/AL Editor window is displayed.

   d. Open the C/AL Globals from the **View** menu.

   e. Add the new function on the **Functions** tab.

   f. Click the **Locals** button.

   g. Add the input parameters on the **Parameters** tab.

      i. Define the first input parameter as a **Record** variable with **Key** name. This parameter specifies the record on which the particular function is executed.

      ii. If the function should return a file result (typically PDF) in the MobileNAV application, add a **BigText** variable with the name **Base64Result**. In this case, a PDF file or an image is returned when executing the function in the MobileNAV application (the application handles the result depending on the **Mobile Type** defined for the function in the **Page Configuration** editor).

      Important: For functions, only these two parameters can be used with Key and Base64Result name. Otherwise the mobile application cannot invoke the function.

   h. Add the necessary variables on the **Variables** tab, if needed.

i. In the C/AL Editor, find the newly created function and insert the necessary code for the business logic.

j. Save and compile the codeunit.

2. Create a dummy field for the function button on the card page.

   a. In Microsoft Dynamics NAV Development Environment, select the appropriate page, where the function must be configured.

   b. Create a new field with Field type.

   c. Fill out the Name and the CaptionML properties of the dummy field, but leave the SourceExpr empty.

   d. Save and compile the page.

3. Record the function into the web services.

   a. In Dynamics 365 Business Central, in the navigation pane, choose Departments/MobileNAV.

   b. Choose **Web Services** from the MobileNAV menu.

   c. Create a new line with the Ctrl + N shortcut key.

   d. Define the following settings for new line:
      Object Type: Codeunit
      Object ID: MN Page Functions- 42012838
      Service Name: the service name of the published page for which the function is created

   e. Leave the **Published** column unchecked, as the codeunit must not be published.

4. Configure the function in the **Page Configuration** editor.

   a. In Dynamics 365 Business Central, in the navigation pane, choose Departments/MobileNAV.

   b. Choose **Configuration** from the MobileNAV menu.

   c. Select the page for which the function is created.

   d. Refresh the page configuration to load the new function with **Refresh page data** from the **Actions** ribbon bar.

   e. Find the dummy field you have created in Step 2.

   f. Enter the exact function name, specified in the MN Page Functions codeunit, in the **Function Name** column of the field.

g. Set the **Mobile Type**.
   To get a result upon function execution in the MobileNAV application, select a Mobile Type for the dummy field. The following Mobile Types are available for functions:

   - PDF,CacheablePDF: The function returns a PDF file in the Base64Result parameter, and the application displays the file after the function is successfully executed.
   - Excel,CacheableExcel: The function returns an Excel file in the Base64Result parameter, and the application displays the file after the function is successfully executed.
   - Image,CacheableImage: The function returns an image file in the Base64Result parameter, and the application displays the file after the function is successfully executed.
   - Refresh: The function modifies the record itself, so the mobile application refreshes the actual card after the function is successfully executed.
   - Close: The function has possibly finished the record itself, so the mobile application closes the actual card after the function is successfully executed.
     Example: The user makes an order from a quote. In this case, the quote is deleted, so the card of the quote must be closed.

h. To assign a picture to the function button, select the appropriate field (created for the function button) and import the picture through the **Picture**/**Import** option on the toolbar of the **Field Configuration** FastTab.

4.7.5.7 Handling BLOB-type table fields

The MobileNAV solution allows handling of BLOB-type table fields through the mobile application. Since BLOB fields are ignored by the built-in page web services, a function must be created to enable viewing or modification of the BLOB field content. With this approach, viewing and editing of BLOB fields, works like for the Picture field of Item, or for the Signature field of Sales Header.

The modification of BLOB fields requires the web service extension mechanism of Dynamics 365 Business Central. The web service extension is ensured by the following conditions:

- A codeunit must be recorded as web service with the same service name as the related card page.
- The Published flag for this codeunit must be unchecked in the Web Services table.
- The first parameter of the functions must have Key name and Record type and must be identical with the source table of the page.
- The second parameter of the functions must be Action name and Text type with length 3. This will specify the BLOB action, like import, export or delete
- The third parameter of the functions must be Base64Result name and BigText type. This will contain the Base64 encoded BLOB content
- The forth (optional) parameter of the functions must be FileName name and Text type. This can overwrite the default filename associated by MobileNAV

As a result, the page is extended with the functions of the codeunit, where the first parameter is the same record type as the source table of the page, so when the user presses the BLOB field in the MobileNAV application, it will invoke this function, and the first input parameter will be filled out with the proper record and the second input parameter will be filled out with the proper action.

Create and configure BLOB functions, as follows:

1. Record a new function into the MN Page Functions program module.
   This codeunit includes functions associated with different pages. You can use this codeunit to add your custom functions, or create a separate one (even in MobileNAV object range) to make merging of new base configuration easier in the future.

   a. Open Microsoft Dynamics NAV Development Environment.

   b. Select the **Codeunit** category.

   c. Locate the MN Page Functions codeunit and click **Design**.
      The C/AL Editor window is displayed.

   d. Open the C/AL Globals from the **View** menu.

   e. Add the new function on the **Functions** tab.

   f. Click the **Locals** button.

   g. Add the input parameters on the **Parameters** tab.

      i. Define the first input parameter as a **Record** variable with **Key** name. This parameter specifies the record on which the particular function is executed.

      ii. Define the second parameter as a **Text** variable with length of 3 and with **Action** name. This variable will contain the selected action: IMP, EXP, DEL.

      iii. Define the third parameter as a **BigText** variable with **Base64Result** name. In this case, a PDF file or an image is returned when executing the function in the MobileNAV application (the application handles the result depending on the **Mobile Type** defined for the function in the **Page Configuration** editor).

      Important: For functions, only these three parameters can be used with Key, Action and Base64Result name. Otherwise the mobile application cannot invoke the BLOB functions.

   h. In the C/AL Editor, find the newly created function and insert the following code for the BLOB field:

```
CASE Action OF
  'IMP' : BEGIN
            //Import - .net version
            SysArray := SysConvert.FromBase64String(Base64Result);
            SysMemoryStr := SysMemoryStr.MemoryStream(SysArray);
            Key.Picture.CREATEOUTSTREAM(outStr);
            SysMemoryStr.WriteTo(outStr);
            Key.MODIFY;
            CLEAR(Base64Result);
          END;
  'EXP' : BEGIN
            // Export - .net Version
            Key.CALCFIELDS(Picture);
            IF Key.Picture.HASVALUE THEN
            BEGIN
              Key.Picture.CREATEINSTREAM(inStr);
              SysMemoryStr := SysMemoryStr.MemoryStream();
              COPYSTREAM(SysMemoryStr,inStr);
              SysArray := SysMemoryStr.GetBuffer();
              Base64Result.ADDTEXT(SysConvert.ToBase64String(SysArray));
            END ELSE
              Base64Result.ADDTEXT('-1');
          END;
  'DEL' : BEGIN
            // Delete
            IF Key.Picture.HASVALUE THEN
            BEGIN
              CLEAR(Key.Picture);
              Key.MODIFY;
            END;
          END;
END;
```

Note: This example is for handling a Picture BLOB field. The field name should be replaced with the desired field name.

Note: In this function, depending on the Action parameter, the function either imports, exports or deletes the content of the BLOB field.

Import:

- decode the Base64Result, since the BLOB will be base64 encoded
- create a MemoryStream for the content
- create a stream for the BLOB field
- write the content from the memory stream to the BLOB stream
- save the modifications of the record
- clear the Base64Result not to send it back to the client

Export:

- calculate the fields of the record
- check whether the BLOB field has value
- create a stream for the BLOB
- create a memory stream
- copy the content of the BLOB stream to the memory stream
- add the base64 encoded content to the Base64Result

Delete:

- check whether the BLOB field has value
- delete the content of the BLOB field
- save the modifications of the record

i. Save and compile the codeunit.

2. Add the BLOB field on the card page.

    a. Open Microsoft Dynamics NAV Development Environment.

    b. Select the appropriate page, where the BLOB field must be configured.

    c. Create a new field with Field type.

    d. Select the BLOB field as SourceExpr.

    e. Save and compile the page.

3. Add the BLOB size field on the card page.

    a. Open Microsoft Dynamics NAV Development Environment.

    b. Select the appropriate page, where the BLOB field must be configured.

    c. Create a global variable with Integer type

    d. Create new field with Field type.

    e. Select the global field as SourceExpr.

    f. The name of the field should be the BLOB field name + "_Size"

    g. The its Visible property to false

    h. Open the C/AL Editor and find the OnAfterGerRecord trigger

    i. Create two Locals

        i. First Locale

            1. Name: InStr

            2. DataType: InStream

ii. Second Locale

1. Name: SysMemoryStr
2. DataType: DotNet
3. SubType: System.IO.MemoryStream.'mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089'

j. Enter the following code for OnAfterGetRecord (SignatureSize is the global, and Signature is the BLOB name)

```
SignatureSize := 0;
CALCFIELDS(Signature);
IF Signature.HASVALUE THEN
BEGIN
  Signature.CREATEINSTREAM(inStr);
  SysMemoryStr := SysMemoryStr.MemoryStream();
  COPYSTREAM(SysMemoryStr,inStr);
  SignatureSize := SysMemoryStr.Length;
END;
```

k. Save and compile the page.

4. Record the BLOB function into the web services.

   a. In Dynamics 365 Business Central, in the navigation pane, choose Departments/MobileNAV.

   b. Choose **Web Services** from the MobileNAV menu.

   c. Create a new line with the F3 shortcut key.

   d. Define the following settings for the new line:
   Object Type: Codeunit
   Object ID: MN Page Functions - 42012838
   Service Name: the service name of the published page for which the function is created

   e. Leave the **Published** column unchecked, as the codeunit must not be published.

5. Configure the function in the **Page Configuration** editor.

   a. In Dynamics 365 Business Central, in the navigation pane, choose Departments/MobileNAV.

   b. Choose **Configuration** from the MobileNAV menu

   c. Select the page for which the function is created.

   d. Refresh the page configuration to load the new function with **Refresh page data** from the **Actions** ribbon bar.

   e. Find the BLOB field you have added in Step 2.

   f. Enter the exact function name, specified in the MN Page Functions codeunit, in the **Function Name** column of the field.

g. Set the **Mobile Type**.
The MobileNAV application identifies the type of the BLOB field with the help of the Mobile Type defined, and chooses the appropriate file viewer application to open the file. The following Mobile Types are available for BLOB fields:

- PDF: The BLOB stores a PDF file, and the application displays it with a PDF viewer.
- Excel: The BLOB stores an Excel file, and the application displays it with Excel viewer.
- Image: The BLOB stores an Image file, and the application displays it with an image viewer.
- Signature: The BLOB stores a Signature file, and the application displays it with an image viewer. During upload the signature can be captured.

4.7.5.8 Handling dynamic field properties (hide, readonly, edit, close, next, add, open, refresh, etc)

The MobileNAV solution can handle certain field properties dynamically. If the MobileNAV user changes the value of a field on a card page, the visibility and editability of other fields of the specific card can be changed accordingly.

The visibility and editability of fields can also be set in the field properties (**Properties** window) in the **Page Designer** window in Microsoft Dynamics NAV Development Environment, but these settings will be static. It is recommended to use dynamic field properties when the visibility and editability of fields depends on data content.

Example 1: You can hide the Spare Part Action and Spare Part No. fields on the Service Line card if the Type field is not Item. These fields will be visible only when the Type field is changed to Item.

Example 2: If the Type field of the To-do record is Meeting, you can hide the Contact No., Contact Name and Contact Company Name fields, and depending on the All Day Event field, you can set the Start Time and Ending Time fields to read-only. If the Type field of the To-do record is not Meeting, the Contact No., Contact Name, and Contact Company Name fields will be visible, but the No. of Attendees, Location, Subject, All Day Event, Start Time, Ending Time fields can be hidden.

Make the following configurations to handle dynamic field properties in the mobile application:

1. Define the FieldControl field on the appropriate page and set the hidden and read-only fields for the appropriate fields, depending on data content.

   a. Open Microsoft Dynamics NAV Development Environment.
   b. Select the appropriate page and click **Design**.
   c. Open the **C/AL Globals** fom the **View** menu.
   d. Add a new global variable on the **Variables** tab with FieldControl name and Text1024 type.
   e. Add a new function on the **Functions** tab with SetFieldProperties name.
   f. Click the **Locals** button.
   g. Add the MobileNAVObjectFunctions local variable with CodeUnit type and MobileNAV Object Functions subtype.
   h. Open the **C/AL Code** from the **View** menu.

i. In the SetFieldProperties function define the values of the FieldControl variable.
   i. To hide a field, call the **SetFieldHidden** function of the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable
      - the name in text format of the field that will be hidden
   ii. To make a field read-only, call the **SetFieldReadOnly** function of the the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable
      - the name in text format of the field that will be hidden
   iii. To open a field for editing, call the **EditField** function of the the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable
      - the name in text format of the field that will be opened for editing
   iv. To close the card, call the **CloseCard** function of the the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable
   v. To close the parent card together with the actual one, call the **CloseParent** function of the the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable
   vi. To go to the next card, call the **NextCard** function of the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable
   vii. To add a new card, call the **AddCard** function of the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable
   viii. To open a card, call the **OpenCard** function of the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable
      - ServiceName: the page to be opened
      - FieldName1 (for OpenCard1, 2, 3), FieldName2 (for OpenCard2, 3), FieldName3 (for OpenCard3), the FieldNames (for OpenCard20 – array with 20 elements): the filtered field
      - FieldValue1 (for OpenCard1, 2, 3), FieldValue2 (for OpenCard2, 3), FieldValue3 (for OpenCard3), FieldValues (for OpenCard20 – array with 20 elements): the value for filtering the field

        Note: OpenCard1 needs one field and one parameter defined, OpenCard2 needs two fields and two parameters defined, OpenCard3 needs three fields and three parameters defined, and OpenCard20 needs a parameter array defined.

      The application will open the card with the defined filtering conditions. If that page is the actually open page, the page is closed and then reopened.
   ix. To open a card after closing all open cards, call the **CloseAllAndOpenCard** function of the MobileNAVObjectFunction code unit with the following parameters:
      - the FieldControl variable

- ServiceName: the page to be opened
- FieldName1 (for CloseAllAndOpenCard1, 2, 3), FieldName2 (for CloseAllAndOpenCard2, 3), the FieldName3 (for CloseAllAndOpenCard3), FieldNames (for CloseAllAndOpenCard20 – array with 20 elements): the filtered field
- FieldValue1 (for CloseAllAndOpenCard1, 2, 3), FieldValue2 (for CloseAllAndOpenCard2, 3), FieldValue3 (for CloseAllAndOpenCard3), FieldValues (for CloseAllAndOpenCard20 – array with 20 elements): the value for filtering the field
  Note: CloseAllAndOpenCard1 needs one field and one parameter defined, CloseAllAndOpenCard2 needs two fields and two parameters defined, CloseAllAndOpenCard3 needs three fields and three parameters defined, and CloseAllAndOpenCard20 needs a parameter array defined.

The application will close all currently open cards and open the card with the defined filtering conditions.

x. To refresh the actual card, call the **RefreshCurrent** function of the MobileNAVObjectFunction code unit with the following parameters:
   - the FieldControl variable

   Use this for function execution. You can also set the MobileType of the field to Refresh on the **Field Configuration** FastTab.

xi. To refresh the parent card, call the **RefreshParent** function of the MobileNAVObjectFunction code unit with the following parameters:
   - the FieldControl variable

xii. To refresh the child card, call the **RefreshChild** function of the MobileNAVObjectFunction code unit with the following parameters:
   - the FieldControl variable

xiii. To refresh the grandparent card, call the **RefreshGrandParent** function of the MobileNAVObjectFunction code unit with the following parameters:
   - the FieldControl variable

xiv. To set the barcode focus to a certain field on a card, call the **SetFocus** function of the MobileNAVObjectFunction code unit with the following parameters:
   - the FieldControl variable
   - FieldName: the name of the field to which the focus should be set
     Not available with SmartScanning.

xv. To remove the filters automatically from the parent list of the actual card, call the **RemoveParentFilters** function of the MobileNAVObjectFunction code unit with the following parameters:
   - the FieldControl variable
   - FieldName1 (for RemoveParentFilters1, 2, 3), FieldName2 (for RemoveParentFilters2, 3), FieldName3 (for RemoveParentFilters3), the FieldNames (for RemoveParentFilters20– array with 20 elements): the name of the field which should be removed from filtering

xvi. To set the color of the First Line of the actual list item, call the **SetFirstLineColor** function of the MobileNAVObjectFunction code unit with the following parameters:

- the FieldControl variable
- ValueColor Red/Green/Blue: the RGB code of the first line

xvii. To set the color of the Second Line of the actual list item, call the **SetSecondLineColor** function of the MobileNAVObjectFunction code unit with the following parameters:

- the FieldControl variable
- ValueColor Red/Green/Blue: the RGB code of the second line

xviii. To set the color of the background of the actual list item, call the **SetBackgroundColor** function of the MobileNAVObjectFunction code unit with the following parameters:

- the FieldControl variable
- ValueColor Red/Green/Blue: the RGB code of the background

xix. To set an icon for the actual list item, call the SetRowIcon function of the MobileNAVObjectFunction code unit with the following parameters:

- the FieldControl variable
- IconName: the name of the icon (in lower case!)

You can choose among the application icons or import new app icons for this purpose with the **Import** option on the toolbar of the **General Setup**.

xx. To set field properties dinamically, call the **SetFieldProperties** function of the MobileNAVObjectFunction code unit with the following parameters:

- the FieldControl variable
- FieldName: the field, whose properties you want to set.
- Hidden: a character (“0” – not hidden, “1” - hidden) or an empty string (factory settings not changed) defining the field to be hidden
- ReadOnly: a character (“0” – not read-only, “1” – read-only) or an empty string (factory settings not changed) defining the field to be read-only
- Caption: defines the caption (if empty, factory settings are not changed)
- CaptionColor Red/Green/Blue: the RGB code of the caption (value “-1” means that factory settings are not changed)
- ValueColor Red/Green/Blue: the RGB code of the value (value “-1” means that factory settings are not changed)

xxi. To display a message, call the **SetDialog** function of the MobileNAVObjectFunction code unit with the following parameters:

- the FieldControl variable
- the identifier of the dialog to be able to react on the answer
- the title of the dialog
- the text of the dialog
- the button set for the dialog. Available options are: Ok,OkCancel,YesNo,YesNoCancel

xxii. To display a toast message, call the **SetToastMessage** function of the MobileNAVObjectFunction code unit with the following parameters:

- the FieldControl variable
- MessageText: the text of the message to be displayed

xxiii. To react on the pressed button, call the **GetSelectedButton** function of the MobileNAVObjectFunction code unit with the following parameters:

- the FieldControl variable
- the identifier of the dialog
- the output parameter will be selected button

Example from the MNToDoCard page:

```
FieldControl := '';


IF Type = Type::Meeting THEN

BEGIN

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'Contact No.');

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'Contact Name');

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'Contact Company Name');


  IF "All Day Event" = TRUE THEN

  BEGIN

    MobileNAVObjectFunctions.SetFieldReadOnly(FieldControl,'Start Time');

    MobileNAVObjectFunctions.SetFieldReadOnly(FieldControl,'Ending Time');

  END

END

ELSE

BEGIN

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'No. of Attendees');

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'Location');

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'Subject');

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'All Day Event');

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'Start Time');

  MobileNAVObjectFunctions.SetFieldHidden(FieldControl,'Ending Time');

END;
```

a. Call the SetFieldProperties function in the **OnAfterGetRecord** trigger of the page.
b. Save and compile the page.

2. Create a new field for the FieldControl variable on the card page.

a. In Microsoft Dynamics NAV Development Environment, select the appropriate page, where the new settings must be configured.

b. Create a new field with Field type and FieldControl name.
Define FieldControl for the Caption and SourceExpr properties and set the visibility to FALSE.

c. Save and compile the page.

3. Configure the new settings in the **Page Configuration** editor.

   a. In Dynamics 365 Business Central, in the navigation pane, choose Departments/MobileNAV Administration.

   b. Choose **Configuration** from the MobileNAV menu.

   c. Select the page for which the new field and the function is created.

   d. Refresh the page configuration to load the new field and function with **Refresh page data** from the **Actions** ribbon bar.
   The FieldControl field is displayed.

   e. Set the **Mobile Type** of the new field to **FieldControl**.

4.7.6 Configure automatic function execution on opening or closing of the card

You can configure automatic function execution on opening and closing of a card. Sometimes you might need to prepare certain temporary records prior to open them as a drill-down, and you want to clean them up on closing. With this technique you can configure functions, and configure them to be automatically executed during opening or closing of the card.

Function execution requires the web service extension mechanism of Dynamics 365 Business Central. The web service extension is ensured by the following conditions:

- A codeunit must be recorded as web service with the same service name as the related card page.
- The Published flag for this codeunit must be unchecked in the Web Services table.
- The first parameter of the functions must have Key name and Record type, and must be identical with the source table of the page.

As a result, the page is extended with the functions of the codeunit, where the first parameter is the same record type as the source table of the page, so when the user presses the function button in the MobileNAV application, it will invoke this function, and the input parameter will be filled out with the proper record.

Create and configure functions, as follows:

1. Record a new function into the MN Page Functions program module.
This codeunit includes functions associated with different pages. You can use this codeunit to add your custom functions, or create a separate one (even in MobileNAV object range) to make merging of the new base configuration easier in the future.

a. Open Microsoft Dynamics NAV Development Environment.

b. Select the **Codeunit** category.

c. Locate the MN Page Functions codeunit and click **Design**.
The C/AL Editor window is displayed.

d. Open the C/AL Globals from the **View** menu.

e. Add the new function on the **Functions** tab.

f. Click the **Locals** button.

g. Add the input parameter on the **Parameters** tab.

   i. Define the first input parameter as a **Record** variable with **Key** name. This parameter specifies the record on which the particular function is executed.

Important: For functions, only this parameter can be used with Key name. Otherwise the mobile application cannot invoke the function.

h. Add the necessary variables on the **Variables** tab, if needed.

i. In the C/AL Editor, find the newly created function and insert the necessary code for the business logic.

j. Save and compile the codeunit.

2. Record the function into the web services.

   a. In Dynamics 365 Business Central, in the navigation pane, choose Departments/MobileNAV Administration.

   b. Choose **Web Services** from the MobileNAV menu.

   c. Create a new line with the Ctrl + N shortcut key.

   d. Define the following settings for new line:
   Object Type: Codeunit
   Object ID: MN Page Functions- 42012838
   Service Name: the service name of the published page for which the function is created

   e. Leave the **Published** column unchecked, as the codeunit must not be published.

3. Configure the function in the **Page Configuration** editor.

   a. In Dynamics 365 Business Central, in the navigation pane, choose Departments/MobileNAV Administration.

   b. Choose **Configuration** from the MobileNAV menu.

   c. Select the page for which the function is created.

d. Enter the exact function name, specified in the MN Page Functions codeunit, in the **OnOpen Function Name** or **OnClose Function Name** column of the page.

4.7.7 Creating workflows – Staging

You can set up workflow-driven user interface for guiding the user step-by-step through filling the card. This way, you can reduce the chances of mistakes. You can define mandatory fields, editable fields, fields that need confirmation manually or through scanning, etc.

In such a workflow, the user goes through the predefined steps and fills in or confirms the mandatory fields. When finished, the user saves the data with the **Register** button.

Example for workflow-driven Pick:

1. Scan the Item Cross Reference to confirm that you picked the proper Item.
2. Scan the Bin Code to confirm that you picked the Item from the right Bin.
3. Enter the quality to handle.
4. Scan the Serial Number of the picked Item.
5. Press the **Register** button.

To create workflow for a page, open the page configuration and enable staging on the **General** FastTab.

In the **Stage Configurator** (opened with the **Assist Edit** button from **Stages** on the **General** FastTab) window you can create new stages (**New Stage**), delete stages (**Remove Stage**) or change the order of the stages (**Move Up Stage**/**Move Down Stage**) with the toolbar buttons under **Stages**. You can also manage translations with the **Show Translations** toolbar button under **View**.

You can decide which fields should be visible and enable the fields that require action from the user. Define the needed action with the **Validation Behavior** options:

- **ScanToValidate:** read-only field with barcode scanning option, where the user must confirm the defined value by scanning.
- **Scan:** editable field with barcode scanning option, where the user can scan any value, but only scanning is possible.
- **ScanWithChangeConfirm:** editable field with barcode scanning option, where the user can scan any value (the value to be scanned can be predefined). If the scanned value does not correspond to the predefined value, MobileNAV displays a confirmation message to the user.
- **Mandatory:** editable field, optionally with barcode scanning (not necessarily), where the user must enter or scan any value to fill or confirm a field. You can set this behavior also for buttons. In this case, MobileNAV will press the button automatically in the given stage.
- **MandatoryToValidate:** read-only field, optionally with barcode scanning option, where the user must confirm the defined value by entering or scanning the same value. If set for a button, the user must press the button in order to proceed to the next stage.
- **MandatoryWithChangeConfirm:** editable field, optionally with barcode scanning option, where the user can enter or scan any value (the value is not predefined). If the entered or

scanned value does not correspond to the predefined value, MobileNAV displays a confirmation message to the user.

You can set automatic proceeding to the next stage if the **Validation Behavior** requirement of the current stage is met with the **Auto Next Stage** option on the **General** FastTab of the page configuration.

You can display **Back** and **Next** buttons for the user to enable navigation between the stages of the wizard with the **Back-Next Visible** option on the **General** FastTab of the page configuration.

For more information on workflows, see the *MobileNAV – Hands-on-Lab* document.

4.8 List and Card configuration

List and Card configuration offers combined configuration possibilities for list pages and card pages on a common configuration page. This type of configuration is advantageous in case of list-card relations with the following conditions:

- if the list contains only a few items, so list and card page type separation is not so important
- if mostly the same fields are needed for displaying the list and the card

In other cases, it is recommended to use the list configuration and card configuration separately.

When using the List and Card configuration, the localized name, picture, first line format, and other common properties will be the same for the list and the card. Properties that belong only to the list or to the card will be handled separately.

Note, however, that in case of offline pages, the **Page Type** on the **General** FastTab must be set to List and Card, and the related card page must be the page itself.

For initiating a card page configuration, see Configuring pages.

For configuration details, see List configuration and Card configuration.

4.9 Offline configuration

Offline configuration is recommended when using the MobileNAV application without Internet connection, or when you are in movement and Internet connection may not be continuous. In the latter case, however Auto online/offline mode is recommended.

To use the MobileNAV application in offline mode, offline pages must be created. These pages are downloaded at offline login and the application will use only the downloaded data. To improve data download procedure, adjust the chunk size to the proper value in the **General Setup** or per page on the **Offline** FastTab of the page configuration, matching the network and server performance.

Data on the offline pages can be modified the same way as in online mode. The modifications are then uploaded to the Dynamics 365 Business Central database when the connection goes online.

The creation, modification, publishing and configuration of offline pages work the same way as for online pages. For more information on these tasks, see Procedure of MobileNAV configuration.

Note, however, that in case of offline pages, the **Page Type** on the **General** FastTab must be set to Offline or Offline Card. This way, the MobileNAV application downloads the content during offline login and uploads the modifications when doing offline data refresh (login or background synch). In offline mode, list and card page types are not differentiated, only offline page type exists, which is basically a List and Card for the offline mode. The Offline Card is to force displaying cards from offline cards. All fields of the cards are necessary in offline mode, so that the user can open any of the required cards from a list. Therefore, the properties of lists and cards should be joined on an offline page type, similarly to the List and Card configuration, so that the MobileNAV application downloads the combined data.

To avoid slow synchronization processes for offline pages with much data, you can use record-level synchronization and periodic check for changes by NAS or by Trigger.

4.9.1 Record-level synchronization

With record-level synchronization, the MobileNAV application downloads only those records that have been modified since the last offline synchronization, and not the entire table, where modifications have been made.

It is recommended to use record-level synchronization in case of offline pages with too much data, in order to avoid slow synchronization process.

Example: The Item table contains 80.000 records, but only 500 records have been modified since the last synchronization. If record-level synchronization is enabled, only the modified 500 records will be downloaded to the application at login instead of all the 80.000 records.

To use record-level synchronization, configuration settings must be made in the table (it can be a standard table of NAV), on the specific page and in the configuration of the page.

Make the following settings in the table:

1. In Microsoft Dynamics NAV Development Environment, select the table in which the configuration settings should be made, and open it for editing with the **Design** button.
2. Add the following new fields to the table:

Enabled	Field Name	Data Type	Filed Class	Description
Yes	RecID	RecordID	Normal	MobileNAV
Yes	User Filter	Code (50)	FlowFilter	MobileNAV
Yes	Page No. Filter	Integer	FlowFilter	MobileNAV
Yes	Last DateTime Modified	DateTime	FlowField	MobileNAV

3. Make the Last DateTime Modified field a FlowField with the following CalcForm:

```
Max("MobileNAV Offline Admin."."Last DateTime Modified" WHERE (User ID=FIELD(User
Filter),Page No.=FIELD(Page No. Filter),Record ID=FIELD(RecID)))
```

This FlowField shows when the specific record was modified last time.

Note: These modifications ensure that the last modification date and time of the record can be queried for synchronization purposes.

4. Create a **SaveRecordID** function, which saves the RECORDID of the record into the newly created RecID field.
    a. Create a local variable for the function with RecRef name and RecordRef Data type.
    b. Insert the following code into the function:

    ```
    // MobileNAV - Offline

    RecRef.GETTABLE(Rec);

    Rec.RecID := RecRef.RECORDID;
    ```

    c. Call the SaveRecordID function in the **OnInsert** and **OnRename** triggers of the table.

Make the following settings on the page:

1. In Microsoft Dynamics NAV Development Environment, create the Last DateTime Modified field on the specific page.
2. Set the source expression according to the name of the field added to the table (Last DateTime Modified).
3. Set the Visible property to FALSE.
4. Create the necessary local variables for the OnOpenPage trigger:
    i. Open the **C/AL Editor** in the Page Designer from the **View** menu or with F9.
    ii. Find the **OnOpenPage** trigger.
    iii. Open the **C/AL Locals** from the **View** menu.
    iv. Create the following variables on the Variables tab:
        - **MobileNAVOfflineFunctions** with **CodeUnit** type and **MobileNAV Offline Functions** subtype
        - **ObjID** with **Integer** type
    v. Close the **C/AL Locals**.
5. Enter the following code to the beginning of the **OnOpenPage** trigger in the **C/AL Editor**:

```
// MobileNAV – Offline

SETRANGE("User Filter",USERID);
EVALUATE(ObjID,COPYSTR(CurrPage.OBJECTID(FALSE),STRPOS(CurrPage.OBJECTID(FALSE),' ')+1));
SETRANGE("Page No. Filter",ObjID);
```

Make the following settings in the page configuration:

1. In Dynamics 365 Business Central, open the specific page configuration.
2. Load the changes into the page configuration with **Refresh page data** from the **Actions** ribbon bar.
3. Select the Last DateTime Modified field in the **Check for Synchronization** field on the **Relation** FastTab.
4. Run the **Initialize Record IDs for Record-level Sync.** option from the MobileNAV menu, in order to automatically fill in the record IDs of the offline pages, for which record-level synchronization has been set.
   For more information, see Initialize Record IDs for Record-level Sync.

With these settings accomplished, the MobileNAV application filters the specific table to those records only, where the value of the Last DateTime Modified field is greater than the date and time of the last synchronization, but smaller than the actual server date and time, that is, the time of the current synchronization. As a result, synchronization between the Dynamics 365 Business Central Server and the MobileNAV application will be faster.

4.9.2 Periodic check for changes

During offline login or offline data refresh, the client application should know, which offline page needs to be synchronized, because there were some changes since the last refresh. The offline pages needs to be “checked for changes”. During periodic check for changes, the modifications made since the last login are analyzed at a pre-defined date and time, in order to ensure a faster login process to the MobileNAV application This way, only modified data is downloaded during login, based on the analysis made previously (at the date and time defined).

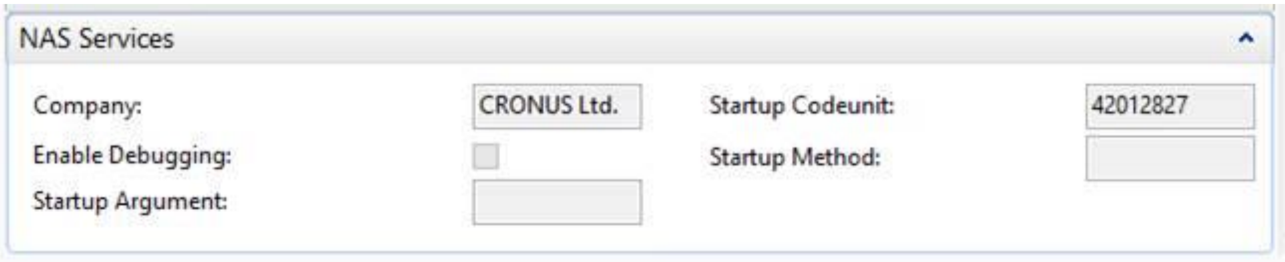
It is recommended to use periodic synchronization in case of offline pages with too much data, in order to avoid slow login process.

The analysis (check for changes) can be made either by the client during login, by NAS (Job Queue) or by Trigger (external logic), based on your setting in the **Check for Changes Period (Hours)** field on the **General** FastTab of the **Page configuration** editor.

To use check for changes by NAS, a NAV Application Server (NAS) can be configured for a scheduled execution of offline data synchronization. For the installation and configuration of NAS, see http://msdn.microsoft.com/en-us/library/dd355183(v=nav.70).aspx

Configure NAS for MobileNAV offline data synchronization as follows:

- If used only for MobileNAV purposes, the following settings must be made:



NAS Services

Company:	CRONUS Ltd.	Startup Codeunit:	42012827
Enable Debugging:	☐	Startup Method:	
Startup Argument:			

Replace CRONUS Ltd. with own company name.

- If used also for other purposes (for example, for running some standard or own codes), complete the NASHandler function of the NASManagement 44 codeunit in the following way:

Original Code:

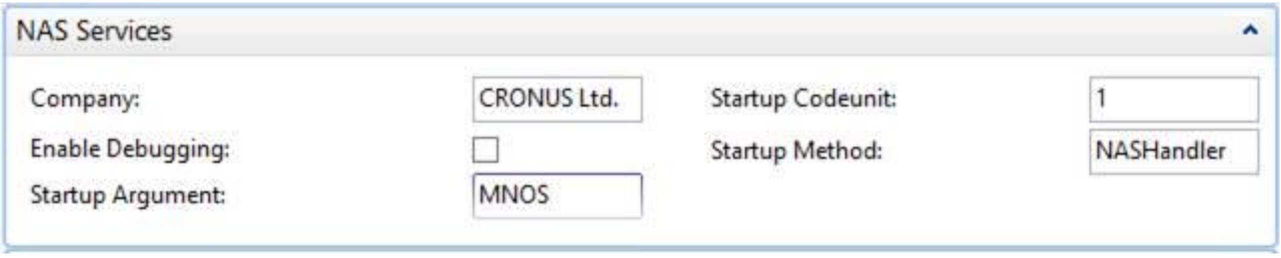
```
CASE Parameter OF
  'JOBQUEUE':
    CODEUNIT.RUN(CODEUNIT::"Job Queue - NAS Start Up");
END;
```

Modified Code:

```
CASE Parameter OF
  'JOBQUEUE':
    CODEUNIT.RUN(CODEUNIT::"Job Queue - NAS Start Up");
  'MNOS':
    CODEUNIT.RUN(CODEUNIT::"MobileNAV Offline Functions");
END;
```

You can also create your own parameter and add your own code, which you have been running on NAS, into the CASE statement.

In this case, the NAS configuration settings are the following:



NAS Services

Company:	CRONUS Ltd.	Startup Codeunit:	1
Enable Debugging:	□	Startup Method:	NASHandler
Startup Argument:	MNOS		

Replace CRONUS Ltd. with own company name.

You must also add the parameters of the standard or your own code (for example, JOBQUEUE) to the startup parameters, separated with comma.

After the NAS configuration, make the following settings in the **Page configuration** editor:

1. In Dynamics 365 Business Central, open the specific page configuration.
2. Select the **General** FastTab.
3. Define the period for offline data synchronization in the **Check for Changes Period (Hours)** field.
   The period can be defined in hours with decimal value (the minimum value is 0.1 hour = 6 minutes)

4. Define how should the check for changes be performed in the **Type of Check for Changes** field.
   By Client: the check is executed as part of the login process.
   By NAS: the check is postponed to NAS.
   By Trigger: you must create a trigger which signals when changes are made by the user and you must refresh data. From version NAV 2016, Event Subscriber methods updates the MobileNAV Add-on automatically for most the offline pages.
5. Define the date and time for the first offline data synchronization in the **Initial Check for Changes (date & time)**field.
   The date and time specified here is the starting point for the synchronization. From this time on, NAS or the client will run the synchronization periodically, as specified in the **Check for Changes Period (Hours)** field. The **Initial Check for Changes (date & time)**field is automatically updated after each synchronization, based on the specified period, always showing the next time of offline synchronization.

   Note: These fields are valid only for offline page types.

When a new user is added to the MobileNAV group, manual synchronization can be made for the specific user in order to enable immediate offline usage of MobileNAV. First, the new user must log in to the MobileNAV application, so that the system knows which data should be updated. After the successful login of the user, you can run a manual synchronization by selecting the user in the **User Setup** window and clicking **Rapid Refresh for Background Sync.** on the ribbon bar. Without manual synchronization, the user can use the application in offline mode only after the next offline synchronization (displayed in the **Initial Check for Changes (date & time)** field).

4.9.3 Background synchronization of the MobileNAV client application

During background synchronization uploading the offline changes to the Dynamics 365 Business Central database is made automatically in the background while the user can continue the offline work. At the same time, offline data on the mobile device is automatically refreshed by downloading changes from the database. Background synchronization is initiated periodically at the specified time interval and is performed in case of successful connection to the database (Internet connection is necessary).

You can define background synchronization settings in the **User Setup** window.

4.9.4 Synchronization recommendations

When neither record-level synchronization nor periodic synchronization is configured for an offline page (this is the case when you set the **Page Type** to Offline), the Dynamics 365 Business Central Server analyzes the modifications made on the page since the last synchronization of the user and if modifications are found, downloads the entire page to the application during the login process. This may result in a long login process or even unsuccessful login, depending on the amount of data on the page to be synchronized.

If record-level synchronization is used, at login to the application, the modifications made since the last synchronization are analyzed, but only the modified records are downloaded. This way, the login process becomes shorter, because less data is downloaded to the application.

This option is recommended for pages with too many records. In this case, use this option together with periodic synchronization to be really effective.

If periodic check for changes by NAS or by Trigger option is used, the modifications made since the last synchronization are analyzed at a pre-defined date and time by NAS and data is downloaded at login. This way, the login process becomes shorter, because analysis is not made during the login process.

This option is recommended for pages with rare modifications both with few records (for example, the CountryCodes or the CurrencyList pages) and with too many records (for example, the Item page used as offline page).

The shortest login process can be achieved if both record-level synchronization and periodic synchronization are used for offline pages. In this case, at login only the modified records are downloaded, and since the analysis is taken over by NAS, it does not affect the login process.

Note: Using both synchronization options is strongly recommended in case of offline pages with too much data (for example, the Item page used as offline page), because not only the download but also the analysis of such pages may take too long, which may result in timeout during the login process.

However, in case of offline pages with a few records and frequent modifications (for example, the MyServiceTasks page) usage of the synchronization options is not sufficient, since the MobileNAV client app might not able to download the latest data immediately.

If you experience timeout during offline login, enable the Log Login Process for the mobile user, then analyze the Login Information generated on the next login attempt.

Analysis of the Login Log:

- Each log line contains the User ID, an Entry DateTime, a Log Message and an Elapsed Time, which is the time difference to the previous log line
- Each Login process will start with a log record: Login process has started
- The Settings Buffer is the generation of the user specific configuration. Normally this should not be longer than 2-3 seconds.
- After this the offline page calculations will follow
    - each offline page calculation contains a started and a finished
    - sometimes the offline page contains that much data, that only its started line is visible (still calculating)
    - the finished line contains how many records has been checked by the Add-on, and you can also check the Elapsed Time to check how long did it take
    - normally a calculation should not be longer than 1-5 seconds, depending on the record count
- Those offline pages where the calculation is longer (like more than 10 seconds), you need to consider to set the **Check for Changes Period (Hours)**, which means that this page won't be calculated during login, so you don't experience timeouti

4.9.5 Offline Administration

During the synchronization process, errors may occur. In this case, the MobileNAV client stops the synchronization process and cannot continue uploading the remaining modifications until the issue is resolved by the user. However, if you enable Offline Administration (in the **General Setup** > **Offline**

Administration FastTab), the MobileNAV user can upload the pending changes to the Offline Cockpit, where you can handle the error and store the changes into the Dynamics 365 Business Central database.

You can handle the offline modifications under Offline Cockpit in MobileNAV Administration. For more information, see Offline Cockpit.

4.9.6 Offline business logic

4.9.6.1 Offline operations

Important: For an improved offline business logic, we recommend using JavaScript execution. For more information, see Offline JavaScript execution.

It is possible to enable operations in offline mode, which facilitate the user's work.

Note: These operations will not be added to the database, even during synchronization. They only ensure a limited offline business logic.

Define an offline operation for the related page, as follows:

1. In the **Page configuration** editor in Dynamics 365 Business Central, select the appropriate field and open the **Relation Configuration** window with the **Relations Setup** button from the toolbar of the **Field Configuration** FastTab.
   You can also open the **Relation Configuration** window from the **Related Pages** column of the appropriate field.

2. Define the necessary operation in the **Operations** column with the help of the **Operation Configuration** window.

The following offline operations can be defined:

Positive-negative adjustment (Pos. Adjmt, Neg. Adjmt)

With this offline operation, the MobileNAV application adjusts the offline inventory positively or negatively if changes occur in the quantity.

Example: Create an operation in the look-up relation on the No. Field of the MN SalesOrderLine Offline, where the **Dest. Field Name** is the Inventory field of the Item, and the **Source Field Name** is the Quantity on the Sales order line.
Result: When the MobileNAV user modifies the Quantity on the Sales order line, the Inventory of the Item is adjusted accordingly.

Field data transfer on creation (Transferfields (Creating))

The content of fields is automatically filled out when creating a new item in a drill-down list, as follows:

- The content of predefined fields are copied from the header to the lines of the new item.
  Example: The currency code is copied from the sales order header to the newly created line.

- Predefined fields of the new item created are automatically filled out with a constant value.
  Example: Line type is set to Item, as default, for the new sales order lines.

Field data transfer on look-up (Transferfields (Lookup))
The content of fields is automatically filled out, when adding the new item from a list, as follows:

- The content of predefined fields of the look-up item are copied to the appropriate fields of the card.
  Example: Copy the Item Description, Unit Price, etc. to the Description, Unit Price, etc. field of the Sales order line when an Item is selected from the opened list for the Sales order line.

- Predefined fields of the new item selected from the look-up list are automatically filled out with a constant value.
  Example: On the Sales order line, Line Discount % is set to 0 when selecting an item from the list.

Line amount calculation

With the UnitPrice, Quantity, Discount%, and LineAmount mobile types, the line amount can be calculated in offline mode, for example, on a sales order line. Set the mobile types in the **Mobile Type** column in the **Page configuration** editor.

Example: The MobileNAV user creates a new line and selects an item. The unit price is filled out automatically (currency and the potential discount for the company is defined in the sales order line). When the user defines the quantity of the item, the line amount is automatically calculated based on the values of the fields.

Multiplication
With this offline operation, you can execute multiplication on a look-up relation. One of the operands must be a field of the selected look-up record, while the other operand must be a field of the specific page.

Example 1: The value of the Unit Price is updated based on the Currency Code and the Exchange Rate (UnitPrice of the selected Item is multiplied with ExchangeRate and the UnitPriceCurrent field is updated with this value).
Example 2: The value of the Quantity (Base) is updated based on the Unit of Measure and the Quantity (Qty. per Unit of Measure of the selected Unit of Measure is multiplied with Quantity and the Quantity (Base) field is updated with this value).

Field data transfer with calculation (Transferfields (Min/Max/Avg/Sum/Count))
With this offline operation, minimum, maximum, average, sum or count values are created automatically in the fields of records from drill-down relations, also taking existing filters into account.

Example 1: The lowest price (Min of Unit Price) is counted based on the customer, the items, the number of the items, and the date of the order from the Sales Price table.
Example 2: The highest line discount (Max. of Line Discount%) is counted based on the customer, the items, the number of the items, and the date of the order from the Sales Line Discount table.

Page Level Operations

Page-level operations are available from the **Offline** FastTab of offline pages and show operations within the page.

The following offline operations can be defined on page level:

Field data transfer (Transferfields (Lookup))

The content of a field is copied to another field within the same page.

Example: On the SalesOrderLineOffline page the value of the Quantity field is copied to the Quantity (Base) field.

Field data transfer with calculation (Transferfields (Min/Max/Avg/Sum/Count))

The minimum, maximum, average, sum, or count value is copied to the specified field within the same page. When copying is made from several source fields to a destination field, the minimum, maximum, average, sum, or count value of the source fields is written into the destination field.

Example: On the SalesOrderLineOffline page, the value of the Unit Price field will be the absolute minimum of the values of the four source fields.

Field Level Operations

Field-level operations are available from the **Field Configuration** FastTab of offline pages and show operations belonging to a field. Field-level operations are designed to be configured for dummy fields with function execution. Via pressing the function button the configured offline operation belonging to the field will be executed.

Field data transfer (Transferfields (Lookup))

The content of a field is copied to another field within the same page.

Example: On the ServiceItemLineOffline page on the Start field the value of the Repair Status Code field is set to IN PROCESS as constant value.

Modification

This type of offline operation is a special operation, because the value set by the offline operation will be uploaded as a function parameter.

Example: On the ServiceItemLineOffline page on the Start field the value of the Starting Date and Starting Time field is set to the current date and time as constant value, and these two parameters will be uploaded as additional function parameter defined for ServiceTaskStartOffline.

4.9.6.2 Offline JavaScript execution

JavaScript execution provides a flexible way to customize your offline business logic according to your needs in an object-oriented development environment. You can create, modify or delete operations or create a dynamic offline UI with the FieldControl feature.

Note: JavaScript execution is not supported by Windows CE.

JavaScript execution contains the following steps:

1. Export the JavaScript template file of the page configuration for an offline page by clicking **Export** under **Actions > Page > Offline Javascript** on the toolbar.

2. Edit the JavaScript template in a JavaScript editor (eg: Microsoft Visual Studio) according to your needs.
   Note: Do not edit the generated codes at the beginning of the file. Please enter your codes below "User defined method starts" for the OnInsert, OnDelete, OnOpen, OnClose events.
   For more information, see the *MobileNAV – Hands-on-Lab* document with exercises on different JavaScript tasks.
3. Import the edited JavaScript template back to the configuration by clicking **Import** under **Actions > Page > Offline Javascript** on the toolbar.
   The system analyzes the imported file and ticks the fields in the page configuration for which codes have been found in the JavaScript file.
   The Template Version number is increased at each import.
4. Log out and then log in again to test the JavaScript execution in offline mode.

At the next login of the MobileNAV user, the MobileNAV client will automatically download the new JavaScript files and use them in offline mode.

You can also delete the JavaScript template of a page configuration by clicking **Delete** under **Actions > Page > Offline Javascript**.

To export, import and delete all JavaScript files at once, go to **General Setup > Actions > Offline JavaSript** section.
Note: AL extension version does not support this action.

You can find more information in the *MobileNAV – Hands-on-Lab* document.

4.9.7 Offline printing

Generating reports or documents is not available in offline mode, since processing the RDLC report layout, and executing the C/AL logic without server connection is almost impossible. However, in MobileNAV, users can generate and print simplified HTML-based documents or other printer-specific files if you configure a layout (HTML or ZPL template) that each mobile platform can process and print.

To configure a HTML or ZPL template for offline printing you need to do the following:

1. Create a dummy field for the function button on the card page.
   a. In Microsoft Dynamics NAV Development Environment, select the appropriate page, where the function must be configured.
   b. Create a new field with Field type.
   c. Fill out the Name and the CaptionML properties of the dummy field, but leave the SourceExpr empty.
   d. Save and compile the page.
2. Configure the function in the **Page Configuration** editor.
   a. In Dynamics 365 Business Central, in the navigation pane, choose Departments/MobileNAV.

b. Choose **Configuration** from the MobileNAV menu.

c. Select the page for which the function is created.

d. Refresh the page configuration to load the new function with **Refresh page data** from the **Actions** ribbon bar.

e. Find the dummy field you have created in Step 2.

f. Set the **Mobile Type** to **HTMLTemplate** or **ZPLTemplate**.

g. Press the Template -> Import… to import the pre-generated HTML or ZPL template

The HTML and ZPL templates can use the following placeholders:

- **field captions on actual page:** indicated by [field_name], like [Sell_to_Customer_Name]
- **field values on actual page:** indicated by {field_name}, like {Sell_to_Customer_Name}
- **field captions on lookup page:** indicated by [field_name_with_lookup.field_name_on_lookup], like [Bill_to_Customer_No.VAT_Registration_No]. This can be used also for lookup field's lookup.
- **field values on lookup page:** indicated by {field_name_with_lookup.field_name_on_lookup}, like {Bill_to_Customer_No.VAT_Registration_No}. This can be used also for lookup field's lookup.
- **repeaters for drill-down records**: indicated by #REPEAT.field_with_drill_down# and #END.field_with_drill_down#. The section between these two will be repeated for each record in the drill-down. This can be used for printing all lines of a Sales Order for example, like #REPEAT.Lines#, where Lines is drill-down button which shows the lines of the Sales Order. Inside this section you can use the fields and the captions of the drill-down records the similar way, like {Lines.No} or {Lines.Quantity}
- **totals for drill-down record fields:** indicated by #TOTAL.field_with_drill_down.field_to_sum#, like #TOTAL.Lines.Line_Amount#, where Lines is drill-down button which shows the lines of the Sales Order, and Line_Amount is the field to use for the total
- **base64 encoded value of BLOB fields:** indicated by {blob_field_name}, like {Signature}. This can be used for include the captured signature on the printed document, or print out the ordered item's cached imageOffline No. Series management

4.9.8 Offline NoSeries management

Once you print out something in offline, the Document Numbering becomes very important. If you just create Sales Orders in offline, but you don't print them out, or you don't want to have a reference number for the Order, then the document number is not important, so you don't need to see and manage it in offline. However, if you would like to print an Order Confirmation, it must contain the Order Number, so you must assign a proper No. for the document. If multiple mobile users use the same No. Series in offline, they would collide most of the times. As a simple (and valid) solution, you must assign unique and dedicated No. Series for each mobile user (and per offline document). These No. Series have to be flagged for Manual editing, since the MobileNAV application must be able to enter the No. “manually” during the offline synchronization.

To enable offline No. Series management you need to do the following:

- in Dynamics 365 Business Central, open the specific page configuration
- find the field on which you need offline No. Series management
- set its **MobileType** to **NoSeries**
- create dedicated No. Series for each mobile user for this offline document, and **enable** the **Manual Nos**
- assign the dedicated No. Series to the mobile users
    - go to **MobileNAV -> Setup -> Users**
    - select a mobile user
    - press **No. Series**
    - create a new entry with the offline page’s identifier field, where you have set the NoSeries MobileType, and set the proper No. Series Code for it
- modify the offline page’s code in Development Environment to handle the No. Series properly
    - In the Object Designer, in Microsoft Dynamics NAV Development Environment, click **Page** and find the offline page, where you need to No. Series management.
    - Open the C/AL Editor
    - Find the OnValidate trigger of the field on which you need the No. Series management
    - Go to View -> C/AL Locals
    - Create a new local
        - Name: **MobileNAVSortingFunctions**
        - DataType: **Codeunit**
        - SubType: **MobileNAV Sorting Functions**
    - Enter the following code for the OnValidate trigger (you need to exchange the {field name} with the proper field name of the identifier field

```
MobileNAVSortingFunctions.SetLastUsedNo(CurrPage.OBJECTID(FALSE),"{field name}");
```

    - Save and compile the page

4.9.9 Temporary Offline Key

When creating a record in offline, the system can create a temporary identifier, to allow using the record on other offline records immediately (without synchronizing the modifications and downloading the identifier created by the server). To use the temporary offline key, fill in the **Temporary Offline Key** and **Temp. Offline Key Separator** fields.

During synchronization, the system checks this temporary identifier on all records, and substitutes it with a permanent identifier on all occurrences.

4.9.10 Force uploading of operation result

When you print out a document, which has certain calculations (for example best price calculation based on Sales Prices), then you might want the system force the offline calculated Unit Price to be uploaded (although the mobile user did not modified it, since offline modification would be automatically uploaded during the offline synchronization. For this matter each offline field has a new parameter: Upload Operation Result. If you set it to TRUE, then it means, that this field value will be uploaded when it is modified via offline operations.

To force uploading of offline operation result on a field you need to do the following:

- in Dynamics 365 Business Central, open the specific page configuration
- find the field which you want to force uploading the offline operation result
- set its **Upload Operation Result** to **TRUE**

4.9.11 Offline page level filter

In offline mode you might need to filter the list so that the list items “disappear” after certain field value has changed. This can be achieved with Offline page level filters.

To configure page level filters, do the following:

- in Dynamics 365 Business Central, open the specific page configuration
- find the field on which you need offline page level filter
- go to **Offline** FastTab, and edit the **Page Level Filters** option
- select the field on which you want to apply the filter
- select CONST as Filter Type
- enter the constant filter value

You can define that when the user filters to a parent page, its child pages will also be filtered based on the parent page filtering. Set this feature by enabling the **Filter By Parent Enabled** option on the **Offline** FastTab of the child page configuration.
Note: In case of new custom pages, you must create this code based on the code in the OnOpenPage trigger of the existing configuration pages.

4.9.12 Limitations in offline mode

The following limitations occur in offline mode:

- The process of look-up and drill-down relation configuration is the same as for online pages, but the related page must be offline, as well.
- Creation of reports is not available
- PDF and Excel report or document generation is not available
- The usual NAV business logic is carried out only when the offline modifications are uploaded to the database. Therefore, a synchronization for uploading the offline modifications to the server is necessary after entering online mode.

- Offline function button’s program logic will be executed when the offline modifications are uploaded to the database. Therefore, a synchronization for uploading the offline modifications to the server is necessary after entering online mode. However, you can write JavaScript for the button OnPressed trigger, which can execute some offline business logic.

4.9.13 Auto online/offline mode

Auto online/offline mode is recommended when Internet connection may not be continuous (for example, when the user is in movement). In this mode, the MobileNAV application is continuously monitoring the connection to the server and switches automatically between online and offline modes, depending on the connection. When the connection is active to the server, MobileNAV works as in online mode; all online functions and buttons are available. When there is no connection to the server, MobileNAV goes offline automatically with the latest data downloaded while online. In offline mode, all online functions, buttons and data are visible but blocked. Offline modifications are automatically synchronized immediately when the connection is restored and MobileNAV goes online again.

You can set the time interval for checking the connection to the server in the MobileNAV User Setup per user via changing the Connection Status Monitoring Interval, or the user can set it under Settings of the MobileNAV application.

In auto online/offline mode, MobileNAV combines the user’s online and offline configuration into one single configuration, where the offline part is always available, however, the online part is available only with active connection to the server. So, while the connection is active, MobileNAV is online, using online pages with all online options available, downloading data from the server and storing it in the offline database, and uploading the modifications to the server immediately. When the connection goes off, MobileNAV goes offline, online options will not be available, data is taken from the offline database and synchronization is done automatically when the connection is restored.

For combined pages use the offline pages and set them to **Use as online** in the **Main Menu Editor** under **Profiles**.

In case of field values that must be recalculated as a result of modifying another value (for example, cost calculation), you can define a “Not Available” value by setting the field value to “null” in offline JavaScript. In this case the value will be recalculated and refreshed when the connection is restored.

4.10 Report configuration

In the MobileNAV application, users can generate or execute reports. The configured reports are available from the Reports list in the main menu of the application.

For detailed information on report configuration, see the *Hands-On-Lab Exercise Add new report (Customer – Summary Aging Simple)*.

1. Create a request page for the report, as follows:
    a. In the Object Designer, in Microsoft Dynamics NAV Development Environment, click **Page/New**.

b. In the New Page wizard define Integer for **Table** in the wizard.
Note: The Integer type source table is needed because the report request page is a dummy page with no data underneath, so the page does not have to be based on a real table.

c. Add new fields to the page with variable source expression for each input parameter and filter of the report.
Important: Only those parameters and filters can be added to the request page, which are available on the specific report. The field names must begin with lower-case letter and must not contain special characters. If an option type variable is related to the field, that variable must start with the „_" field name character and at the field properties the OptionCaptionML, SourceExpr and CaptionML values must be added.
The data type of the variable defines the type and format of data that can be specified in the MobileNAV application. The MobileNAV application reads into these variables the filter conditions and other input parameters specified on the interface. The pages created this way function as request pages.

d. Create a dummy field for the function button on the request page.
Add a new field with Field type and fill out the Name and the CaptionML properties of the dummy field, but leave the SourceExpr empty.
This function button will execute the generation of the report.

2. Publish the new page object to the web services to be accessible from the MobileNAV application. For more information, see Publishing pages.

3. **Only for NAV 2013 R2 or below:** In Microsoft Dynamics NAV Development Environment open the appropriate report with **Design** and create a new function in the report itself with the same input parameters as the fields of the report page.
In this codeunit the input parameters must be stored for later processing. Furthermore, you must guarantee that these stored settings will be taken into account during the generation of the report.

Example: To specify the chart type for the Customer Top 10 report, add an input parameter for this function, where the mobile device can set the required parameter.

4. Create the function in the MN Report Functions program module.
The function parameters must correspond to the fields of the report respectively. Furthermore, a Base64Result output parameter must be added with BigText data type.

This function has the following roles in **NAV 2013 R2 or below**:

- sets the language of the report according to the user preferences
- creates a temporary file for the report generation
- applies the parameters on the requested report via the previously created function (in step 3)
- generates the report in PDF or Excel format
In case of process-only reports, there is no report generation.

- returns the content of the generated PDF file in Base64 format for the MobileNAV application (Base64Result)

This function has the following roles in **NAV 2015 or above**:

- sets the language of the report according to the user preferences
- invokes the MobileNAV Report Helper codeunit

    i. call Initialize which initializes the codeunit

    ii. call AddByPageField, AddParameter or AddDataItemFilter to add parameters and filters to the report generation

    iii. call Generate to generate the report in PDF, Word, Excel or HTML, and the result will be inserted into the Base64Result return parameter

5. For initiating a report page configuration, see Configuring pages.
6. In the **Page Configuration** editor, set the properties of the report page.

    The following report page properties are recommended to be set for MobileNAV configuration in the **General** and **Behavior** FastTabs:

    - **Menu Picture Version**: It is recommended to assign a picture to each report as it facilitates navigation. For more information, see the Assigning menu picture to pages.
    - **Category**: It is recommended to select a category for the reports.
      The Reports list in the Finance category, in the main menu of the MobileNAV application, contains the available reports, where the first line contains the name of the report, coming from the **CaptionML** property of its request page, and the second line contains the category of the report.
    - **Page Update**: Enable operations on the report page in the MobileNAV application.
      Note: Page Update must be set separately in order to work properly.
      Page Insert and **Page Delete** must be False.
    - **Report ID**: only from NAV 2015 or above. You can bind the request page with the report object for MobileNAV Report Helper codeunit

    The following report field properties are recommended to be set for MobileNAV configuration in the **Field configuration** FastTab:

    - **Mobile Type**: Define the operation that is performed when executing the configured function in the actual field on the mobile device.
      The condition specified on the mobile device will be dispatched as textual filter condition. The following mobile types are available for reports:

        - Normal: No special operation.
        - Barcode: The value of the field can be loaded with barcode scanner.
        - Filter: Filtering is made based on the defined values.

- **Related Pages**: Configure look-up relations from a particular field.
  Look-up relations are necessary in reports in order to make it possible to select data from a list when compiling parameters or filters.
  Drill-down relations are not interpreted for reports.

- **Report Field Source and Report Field Name:** only in NAV 2015 or above. You can bind request page fields to report parameters or report DataItem fields as filters

The following dummy field properties for function buttons are recommended to be set for MobileNAV configuration in the **Field configuration** FastTab

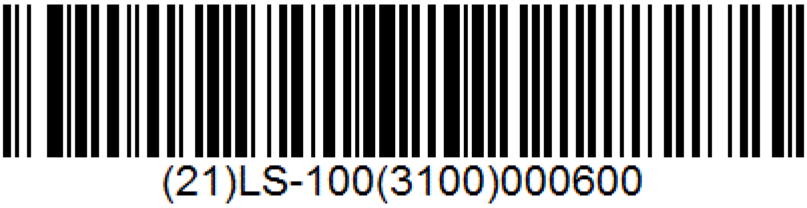
- **Function Name**: Enter the exact function name, specified in the MN Report Functions codeunit in step 4, in the Function Name column of the field.

- **Mobile Type**: To get a result upon function execution in the MobileNAV application, select a Mobile Type for the dummy field. The following Mobile Types are available for functions:
  - PDF: The function returns a PDF file in the Base64Result parameter, and the application displays the file after the function is successfully executed.
  - Excel: The function returns an Excel file in the Base64Result parameter, and the application displays the file after the function is successfully executed.
  - Image: The function returns an image file in the Base64Result parameter, and the application displays the file after the function is successfully executed.
  - Normal or Close: for process-only reports

    Note: To generate the same report both in PDF and Excel format, create separate dummy fields and functions for each type.

4.11 Barcode preprocessing with Regular Expressions

In this exercise you will learn how to handle composite barcodes or QR codes with MobileNAV. Composite barcodes contain more than just the Item identifier, so when you scan it and try to use it as a filter, it will contain all the data what the barcode contains, which results in not finding the proper Item in a list. For example you have GS-128 barcodes with Item No. and Quantity fields, like this:



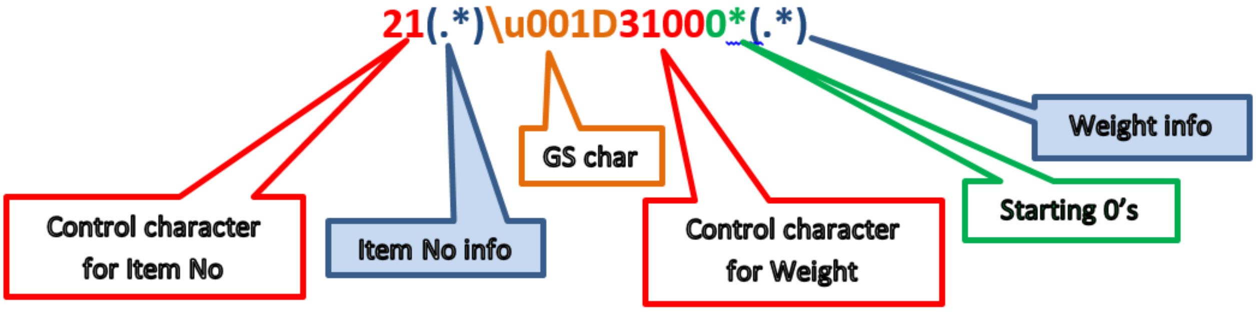
This very simple GS1-128 barcode contains the LS-100 as the Item No. (for simplicity), and 600 as Quantity (310 specifies that it is weight, and 0 specifies the decimal places).

If you try to scan this barcode on the Item list in MobileNAV, then it won't find the proper Item. If you check the Filter of the list after scanning it shows that the full barcode has been read into the Item No. filter, which is not good.

To pre-process the composite barcodes, you need to modify the configuration to let the MobileNAV client know, which part of the barcode should be used for the Item No field, and which part is the quantity.

The approach for this is to use Regular Expressions, which is a standard for pattern matching. You can find information here: http://en.wikipedia.org/wiki/Regular_expression

In MobileNAV Configuration you can define Regular Expression for a field, and you can also specify which potion of the match should be extracted for that field. For a typical GS1-128 barcode you can define the following search pattern:



Each brackets will have a match result, and given the previous barcode sample, the following will be the match results:

1. match: LS-100

2. match: 600 (since the starting 0's is not part of the match)

So you need to extract the second match on the Item list, in order to filter by Item No properly, or you can also extract both on a consumption journal line, so you can enter the Item No and the Quantity with one scanning.

To configure barcode preprocessing with Regular Expressions you need to do the following:

- in Dynamics 365 Business Central, open the specific page configuration
- find the field on which you need barcode preprocessing
- enter the Regular Expression into the **RegEx Text** property of the field
  You can combine multiple Regular Expressions with the | character: ex: 21(.*)\u001D31000*(.*)|31000*(.*)\u001D21(.*)
- enter the Regular Expression match result order into the **RegEx Results** property of the field
  In case of combined Regular Expressions, you must enter the same amount of numbers separated with comma as the number of the Regular Expressions defined in the **RexEx Text** field property (ex: Item no: 1,3 es a Quantity 2,4). If the defined field is not in the specified expression, use an empty character.
- group all the fields into the same group by setting the same **Category**
  - empty Category is also one group
  - the **MobileType Barcode** setting is mandatory for only one group member

5 MobileNAV Department

Here you can find the MobileNAV menu items, through which MobileNAV-related activities can be carried out.

To open the MobileNAV menu, choose Departments/MobileNAV in the navigation pane of Dynamics 365 Business Central.

5.1 Lists

Here you can perform device management settings and configuration settings. The **Device Management** window and the **Page Configuration** editor are accessible from the MobileNAV menu.

5.1.1 Device Management

The MobileNAV application provides device-based access to the database of Dynamics 365 Business Central. This means that during login to the MobileNAV application, the Add-on checks whether the number of the active devices has reached the maximum number defined in the license, based on the unique identifier of the device using the application.

During login with a new device, the Add-on registers the username, the device identifier, and the date and time of the login attempt. The Add-on also checks the number of the active devices. If the maximum limit, defined in the license, has not been reached yet, the device will have “Active” status and access to the MobileNAV application is granted. If the maximum number of active devices has been reached, the device will have “Standby” status and cannot log in to the MobileNAV application.

During login with a registered device, the Add-on checks the status of the device. If the status of the device is “Active”, access to the MobileNAV application is granted. The Add-on supervises the username and the date and time of the login attempt. If the status of the device is “Standby” or “Rejected”, access to the MobileNAV application is denied.

Device management activities

In the **Device Management** window, you can perform device management activities. The window shows the registered devices from all companies, the device information saved during login, and the license information.

The content of the **Device Management** window is refreshed automatically by the seconds, so the list of the registered devices and the license information displays the latest information at any moment. To manually refresh information in this window, click the **Refresh** button.

The registration of a device can be ended by deleting the row that contains the device information with the **Delete** button. In this case, the Add-on will handle the device as a new device at the next login.

If you requested a new license for more devices or your MobileNAV trial license expired, you can change your MobileNAV license with the **Change License** button. In the **Change MobileNAV License** window simply paste the MobileNAV License String from the received email and click **OK**.

To view the registered GPS coordinates of a specific device, select the device from the list and click the **GPS Information** button. In the GPS Coordinates window you can analyze the route and location of the selected device and the distance made. You can also check the location and the route of the device displayed on the Google map with the **Show Route on Google Map** button. For more information, see GPS Coordinates.

The status of the device can be changed manually in the **Status** column, as follows:

- **Active**: Access to the MobileNAV application will be granted at login. The maximum number of active devices is defined in the purchased license.
- **Standby**: Access to the MobileNAV application will be denied at login, because the maximum limit of active devices has been reached. The status of the device will automatically change to “Active” at login if an active device has been deleted from the database before the login.
- **Rejected**: Access to the MobileNAV application will be denied at login, because the device is not used anymore for MobileNAV handling.
- **Support**: You can define one device for supporting purposes. The support device cannot be used commercially, so license is not needed; it is reserved for analysis and bug fixing from the partner.

In addition to the status of the registered devices, the following information helps the identification:

- **User ID:** the identifier of the user who logged in the last time with the device
- **Device ID:** the identifier of the device
- **Description**: this is an editable field where you can enter custom description which helps you identify the devices easier
- **Creation Date:** the date of the first login from the device
- **Last Login Date:** the date of the last login from the device

Under **License Information**, the following data is displayed:

- **Available**: the maximum number of devices defined in the license
- **Assigned**: the number of the active registered devices
- **Remaining**: the number of the devices that can still be registered
- **Schema Version**: the major version of the Add-on
- **License Exp. Date:** in case of Demo licenses, the expiration date of the license
- **Actual Voice ID:** the VOICE ID of the currently imported license owner
- **License Owner Voice ID:** the VOICE ID of the purchased MobileNAV license owner
- **Warning Message:** displayed when the current license differs from the owned license

5.1.2 Configuration

Here you can configure different page types for MobileNAV handling.

For configuration details, see Configuring pages.

5.1.3 GPS Coordinates

Here you can analyze the route and location of the MobileNAV users and the distance made, if GPS tracking has been enabled in the **User Setup** window.
Data is automatically uploaded from the device into the database with the appropriate user ID and device ID.

If the user is logged in to the MobileNAV application and uses it, the application collects data every 5 minutes. In offline mode, the application saves data on the device and uploads the coordinates to the server when the user is online again.

To check the route of a user displayed on the Google map, select the specific coordinates of a user, click **Show Route on Google Map** on the ribbon bar of the **GPS Coordinates** window.

To check the location of a user on the Google map based on a specific coordinate, select only one coordinate, click **Show Route on Google Map** on the ribbon bar of the **GPS Coordinates** window.

Beside the options mentioned above, the GPS coordinates can also be used for other purposes. For example, the distance and location reported by the serviceman or salesperson can be verified.

5.1.4 Offline Cockpit

Here you can manage offline modifications uploaded by the MobileNAV users.

Note: Offline Administration must be enabled in the **General Setup** > **Offline Administration** FastTab.

You can handle the emerging errors and store the offline changes into the Dynamics 365 Business Central database, facilitating the synchronization processes of the MobileNAV client app. For more information, see Offline Administration.

The Offline Cockpit contains the uploaded offline modifications per user. You can view the offline modification log with the necessary information, including the following:

- **Page Captions:** the modified pages
- **Created DateTime:** the date of modification
- **Operation:** the offline operation made on the file
  Operation types: New (creation), Modify, Delete, Function Call (function execution), Upload Blob, Delete Blob
- **Status:** the synchronization status of the file
  Status types: Not synchronized, Synchronized, Dismissed (reverted), Error (see Error Message), Warning (the record has been modified on the server while used offline)
  In case of warning, you can choose whether to upload the offline modification to the database or not. Uploading the offline modification does not revert the server modification.
- **Error Message:** the error message in case of error
- **Client First Line** and **Client Second Line:** the details of the action taken offline

To synchronize the offline modifications, choose from the following options:

- **Synchronize First:** synchronizes the oldest offline modification only
  With this option, you can synchronize the offline modifications one by one, always starting with the oldest modification.
- **Synchronize All:** synchronizes all offline modifications of the user
  The modifications will be synchronized in chronological order. In case of error, the synchronization process stops and will not continue until you solve the issue.

To cancel the offline modifications, choose from the following options:

- **Revert:** revert the selected offline modification
- **Revert all:** revert all offline modifications of the user

Note: When an action is reverted, all related actions will be reverted, too.

With **Show History**, you can view the result of your synchronization or reverting.

5.2 Tasks – Periodic activities

Here you can perform general activities related to the configuration. Periodic activities are accessible from the MobileNAV menu.

5.2.1 Configuration Handling Report

Here you can import or export a particular configuration.

Select Import or Export, browse for the configuration file (.xml file), and click **OK**.

During import, the current configuration is erased and the new configuration takes into effect. The objects (pages, codeunits, reports) that are part of the configuration must be imported and translated manually. It is recommended to import the objects of the configuration before the configuration itself, in order to avoid error in the configuration. You can choose to import the base configuration or a custom configuration. The base configuration matches with the original base configuration objects released by MultiSoft.

Note: Importing the configuration objects must be executed with precaution, in order to avoid overwriting objects that you do not want to.

During export, the configuration data is exported into an .xml file with all the necessary configuration settings. The objects of the configuration (pages, codeunits, reports) must be exported separately using the Designer.

Note: The configuration is always created in English version.

Before saving the configuration, you are asked whether to refresh the configured data of the pages based on the meta-data of the page.

5.2.2 Validate Configuration

You can validate your custom configuration with this predefined report. It checks many common errors in the configuration. You can apply filter for the configuration page, but it is recommended to run this report periodically after making changes in the configuration, and before publishing a newer configuration version. The report indicates errors and warnings. It is recommended to fix all the errors. Fixing the warning is not critical (like assigning menu picture to a page), but it helps in creating a valid/complete configuration.

5.2.3 Draw all page relations from main menu

You can “draw” the page relations starting from the main menu with this predefined report. This could be helpful if you want to discover whether a certain configured page is orphan or not, or from where a certain page is referenced. There are two parameters for the report:

- Show Source Field: if TRUE, then not only the page will be listed where it is referenced, but the field within the page, where the relation is defined.
- Summary Only: if TRUE, then only a summary is generated with the used pages

5.2.4 Change table related filter values

You can change table related filter, condition or operation constant values in one place. If you have a field with a lookup, and you define filters, conditions or operations on this field via referencing a constant value from the table, then you might need to change the configured value in one go, because of the localized version is having different values. For example, in the base configuration the Start or Finish button sets the Repair Status Code to IN PROCESS or FINISHED. In localized databased the values of the Repair Status Code is very likely different, so after importing the base configuration you need to change these values to the proper one quickly. With this feature you can change them on one screen, so you don’t need to browse all the page and change them one by one.

5.2.5 Refresh data for all configured pages

Here you can refresh all page configurations to load all changes made in the Page Designer at once.

To refresh page configurations individually, open the appropriate page configuration and click **Refresh page data** on the **Actions** ribbon bar.

5.2.6 Initialize Record IDs for Record-level Sync.

With this option, the server automatically fills in the record IDs of all the offline records, for which record-level synchronization has been set, but no record ID has been saved previously.

It is recommended to use this option immediately after modifying page configuration settings, if record-level synchronization has been set, so that the record IDs are filled in before user login.

For information on record-level synchronization settings, see Record-level synchronization.

5.2.7 Category Translation Import/Export

Here you can import and export category translations.

Browse for the category translation file (.txt file), select Import or Export and click **OK**.

Category translations are included in the MobileNAV Category Translation .txt file in the installation package.

For more information on translations, see Chapter *Import MobileNAV Configuration translations* in the *MobileNAV Installation Guide*.

5.3 Administration – Setup

Here you can perform general activities related to the setup. Setup activities are accessible from the MobileNAV menu.

5.3.1 Users

Users are added to this list at login with information on the language used at the last login, needed for report generation.

It is possible to provide users of the MobileNAV application with pre-filtered lists so that they can easily access, for example, the list of their customers and invoices. For this, a salesperson code must be assigned to the login identifier of the user in case of a salesperson, or a service resource filter in case of a service technician:

- To associate a user with a salesperson, define the Windows login identifier in the **User ID** column and the salesperson code in the **Salesperson Code** column in the **User Setup** window.
- To associate a user with a resource ID, define the Windows login identifier in the **User ID** column and the resource ID in the **Service Resource Filter** column in the **User Setup** window.

  For more information on user settings, see Setting up MobileNAV users.

You can call web services on behalf of the user without knowing the user's password, if you set a special password in the **Web Services Access Key** column. Defining the web services access key is essential when using Office 365 or NAS. For example, when the MobileNAV application uses O365, it receives one token for one hour, so when the NAS queries pages on behalf of the user, it may run out of time. To avoid timeout errors, it is recommended to use the web services access key. O365 authentication will use this special password and the user can use MobileNAV fluently.

In the **User Setup** window, you can also run manual synchronization of offline data for specific users. Manual synchronization of offline data is recommended when new users are added to the MobileNAV group. For this, select a user and click **Rapid Refresh for Background Sync.** in the ribbon bar. For more information on manual synchronization, see Periodic synchronization.

If you configured No. Series management for offline, then pressing the **No. Series** button you can check and modify the assigned No. Series for a certain user. For more information on offline No. Series management, see Offline No. Series management.

If you created profiles, you can assign them to users in the **Profile Setup** window by clicking on the **Profiles** option on the ribbon bar of the **User Setup** window. By assigning one or more profiles to a user, you can control page access by user from the main menu of the MobileNAV application. For more information on profiles, see Profiles.

If you have enabled **Log Login Information** for the user, then pressing **Login Information** you can access the generated log. For more information on login timeout detection, see Synchronization recommendations.

It is also possible, to view the route and location of MobileNAV users and to check the distance made. For this, GPS tracking must be enabled for the specific users in the **GPS tracking enabled** column in the **User Setup** window. The distance can be queried in kilometers or miles (as set in the **Units** column). For more information on GPS tracking, see GPS Coordinates.

You can also handle background synchronization settings for the user in the **Background Synchronization** column, turning the feature ON or OFF, or empowering the user to set it as needed (By Client).

Note: If turned on, the time interval of the synchronization comes from the **Background Synchronization Interval (min.)** setting on the **General** FastTab in the **General Setup**.

For more information on background synchronization, see Background synchronization.

The login mode of the user can be set in the **Login Mode** column. Login mode determines what the user can do after logging in to the MobileNAV application.

You can define the following values:

- **By Client:** the user can decide whether to use online mode, offline mode, or auto online/offline mode.
- **Online:** the user is forced to use online mode and cannot switch to offline mode.
- **Offline:** the user is forced to use offline mode and cannot switch to online mode.
- **Online/Offline:** the user can switch between online and offline modes but cannot enable auto online/offline mode.
- **Auto Online/Offline:** the user is forced to use auto online/offline mode and cannot switch to other modes.
- **Strict Auto Online/Offline:** the user is forced to use auto online/offline mode in strict mode and cannot switch to other modes. Strict mode means that online features are not available until all offline changes are successfully uploaded to the server.

If you have installed the MobileNAV Diagnostics Tool on the computer where you are running the Dynamics 365 Business Central client, then you can utilize the MobileNAV Diagnostics Tool from the MobileNAV Add-on directly. The MobileNAV Diagnostics Tool helps you to analyze the connection problems or performance. If the mobile user did not logged in with MobileNAV before, then you need to fill out the login parameters like you would do it in MobileNAV client login screen. Additional parameters as follows:

- Language: you can set which language you would like to simulate with the tool
- First Login (downloading configuration): if TRUE, then you will simulate the first login, so the configuration will be downloaded
- Trace Debug Messages: if TRUE, then a detailed trace will be generated
- Read All Pages after Login: if TRUE, then after simulating a login, all pages will be tested with reading all data from them

Pressing the Start Diagnostics you can select the following:

- Online: the tool will simulate an online login
- Offline: the tool will simulate an offline login
- OnlineOffline: the tool will simulate an online login first, then an offline login

Pressing the Open Log Folder will open the folder, where the logs are generated.

5.3.2 Categories

Here you can manage the category names and their translations, used in the main menu of the MobileNAV application.

Categories can be customized according to your needs. Pages and fields of a particular page can be associated to the appropriate category to make navigation in the application easier or to handle the visibility settings of fields of the same category at once from the **General Setup**.

To create new categories, follow the instructions below:

1. Select **New** from the ribbon bar in the **Categories** window.
2. Define an identifier code in the **Code** column.
3. In the **Description** column, enter the name to be displayed by the MobileNAV application.

   Note: Use English for the original code and description.

To assign translated descriptions to the individual categories, follow the instructions below:

1. Select **Translation** on the **Actions** ribbon bar.
   The **Category Translations** window is displayed.
2. Select the appropriate language code in the **Language Code** column.
3. In the **Description** column, enter the translated name to be displayed by the MobileNAV application, if not existing yet.

5.3.3 Profiles

Here you can create profiles for users to determine which pages a user should access from the main menu in the MobileNAV application.

To create new profiles, follow the instructions below:

1. Select **New** from the ribbon bar in the **Profile Setup** window.
2. Define an identifier code in the **Code** column.
3. In the **Description** column, enter the name of the profile.

The created profile must be assigned both to the user and to the pages that the user should access from the main menu:

- Assign profiles to users with the **Profiles** option on the ribbon bar of the **User Setup** window (see Users).
- Configure the profiles in the **Main Menu Editor**.
  For more information, see Main Menu Editor.

Note: If a user has more profiles and those profiles are combined, problems may occur, as different settings of the same page may be defined in the different profiles.

5.3.3.1 Main Menu Editor

You can edit the Main Menu of a MobileNAV app user by opening the user's profile from the list of profiles. The profile is opened in the **Main Menu Editor** displaying the pages assigned to the profile.

Here you can add new pages to the profile, edit or remove pages, and set the order of the pages. You can also filter pages with the **All Pages**, **As Auto Online/Offline**, **As Online**, **As Offline** filters, to view the main menu of the MobileNAV user after an online, offline or auto online-offline login.

Under **Hierarchy**, the pages are displayed in hierarchical order with all related pages, just as in the Main Menu of the user.

Here you can define the following settings for the main pages and the related pages as well:

- **Display in Menu:** you can define which pages should be available in the Main Menu of a specific MobileNAV user.
- **Use as Online:** you can use an offline page as online page in a specific profile (in case of Auto online/offline mode).
  As a result, the user will see and use that page in MobileNAV as online page when connected to the server. For more information, see Auto online/offline mode.
- **Disable Staging:** you can disable staging in a specific profile for a page, where Staging is configured. As a result, staging will not be used for that user. For more information on staging, see Creating workflows – Staging.
- **Exclude from Profile:** you can remove a related page from a specific profile.
  As a result, the user will not see the excluded page in MobileNAV at all.
- **Lookup Only:** you can set a normal page to a lookup-only page for a specific profile.
  As a result, the user will be able to use it for lookup selection but will not be able to access the card page by pressing the icon for the selected lookup value.
- **Exclude by Parent:** automatic setting, which shows whether the page is excluded from the profile because the parent page has been excluded.

The hierarchical pages are color-coded in order to facilitate orientation. The following color codes are used:

- Blue font: offline page
- Black font: online page
- Grey font: excluded page
- **Bold font**: list or card page (online or offline by color) – not look-up only
- Normal font: lookup only page (online or offline by color)

It is also possible to edit the fields of the pages based on profiles. You can define which fields of the page should be visible for a specific user, which fields should be read-only, or what actions the user may take on the page. To edit the fields of a page for the MobileNAV user, select a page from the list and press **Edit Page** under **Actions > Pages** on the toolbar. The **Profiled Page Editor** window opens, where you can hide fields or make fields of the selected page read-only for a specific profile.

5.3.4 Web Services

Here you can publish page objects to the web services in order to access them from the MobileNAV application.

The web services available by the MobileNAV application are classified into two groups:

- **Web services related to Add-On** (Service name: MobileNAV*): The MobileNAV application reaches the configuration, which determines the operation of the application itself, through these web services. These entries must be included accurately among the web services.
- **Web services related to the configuration** (Service name: MN*): The MobileNAV application reaches the master and transaction data through these web services. Pages of the configuration must be included among the web services together with the object ID and services name, exactly the same way, as in the configuration.

These web services are automatically created during the configuration import. However, when a new page or a new function is created, it must be published to the web services manually. For information on the publishing process, see Publishing pages.

5.3.5 General Setup

The following functions are available from the toolbar:

Process:

- **Set Configuration Changed:** with this function you can force to MobileNAV Add-on to recalculate the user dependent configuration on the next login of the user, either the configuration did not change. This can be helpful, if you made changes, but for some reason you don't see the changes in the MobileNAV client app.
- **Delete Login Cache:** deletes all cached login data, like user specific login URL + credentials, which are used when the Add-on invokes himself thru web services to mimic the client app behavior. This feature is useful if the server environment has changed (like the instance name or server address).
- **Clear Settings:** with this function you can reset all general and/or color + logo settings.
- **Rebuild Profiled Hierarchy:** with this function you can rebuild the hierarchical relationship of pages manually.
  The server performs this action automatically when the user makes modifications on a page which affect other pages (for example, deleting a page, creating a relation to another page, etc).

Report:

- **NAS Login Information:** here you can access the Login Information generated by the NAS during background refresh.
- **Hide All:** here you can hide all fields of a specific category at once (Variant Code, Unit of Measure Code, Bin Code, Zone Code, Currency Code, Serial No., Lot No. fields).

You can assign field categories to frequently used fields on the **Field Configuration** FastTab of the page configuration.

Actions: here you can find the most often used functions mentioned above and the following additional function:

- **Picture:** here you can import, export or delete small and large icons to customize the icons of the application and the page configurations according to your needs. In C/AL version you can also import, export or delete all icons at once.
  Note: Please take care about the naming and the resolution of the icons, because the MobileNAV client downloads the icon resolution matching the local screen resolution of the device. For more information, see the *MobileNAV – Hands-on Lab* document.

The following settings are available from the FastTabs:

General FastTab: here you can specify general settings for the MobileNAV installation.

- **Auto. Device Activation:** if TRUE new devices will be automatically activated, if there are free licenses. If FALSE, then the device won't be able to log in until the Device Admin sets its line to Active from Stand-by. The default is TRUE.
- **Timeout Limit (Sec.):** this setting contains the login timeout. With previous version there were no server side timeout (client app timeout was 60 sec), which could cause high server load if the login was too long (in case of much offline data, which is not configured for Background Refresh). The default is 55 sec, which means that it is shorter than the client app timeout, so the user will receive a proper error message if the timeout was because of a long login operation.
- **Log Background Refresh:** with this setting you can control whether you would like to generate log during the Background Refresh by NAS. The default is FALSE.
- **Server Name:** with this setting you can change how the MobileNAV Add-on generates the URL which will be used for invoking himself thru web services. In certain installations it turned out, that using Full Qualified Domain Name for the server PC does not work, and either "localhost" or just the Computer Name has to be used. This option allows you to change this without C/AL code modification in the MobileNAV Add-on code.
  Besides FQDN, MachineName and Localhost, you can also choose Custom for Server URL. In this case, you can enter an arbitrary Web Service URL.
- **Testing Mode:** turn on Testing Mode to see the result of your customization settings.
- **Users Waiting For Refresh:** here you can view the number of users waiting to refresh their offline data.
- **Web Service URL:** here you can define the Web Service URL. If you selected Custom for Server URL at the Server name field, enter an arbitrary server URL.
- **Leave Session at Logout:** turn on this setting to always leave the session at logout from the server.
- **Background Synchronization Interval (min.)**: here you can set the time interval in minutes when background synchronization should be initiated on the MobileNAV client. It is performed in case of successful connection to the database (Internet connection is necessary).
- **Chunk Size for Offline Data:** here you can set the number of records downloaded from the web service in one chunk.
  By adjusting the chunk size, you can improve the user experience during the download. Choose a proper value for chunk size, which matches the network and server performance.
  Note: You can also set the chunk size for pages separately on the **Offline** FastTab of the page configuration. In this case, the setting on the page configuration overwrites the setting in the **General Setup** when downloading that specific page to the device.

- **Deleted Offline Data Purge Interval (Days):** here you can set the time interval in days when deleted offline data should be purged.

Offline Administration FastTab: here you can define Offline Administration settings.

- **Feature Enabled:** here you can enable the Offline Cockpit functionality.
- **Checkout Time Interval (minutes):** here you can set the time interval in minutes when the lock should be released on the files uploaded to the Offline Cockpit. This is important in case of synchronization errors.
- **Send Email to Administrators:** turn on this setting to enable receiving email notifications whenever a new item is uploaded to the Offline Cockpit for synchronization.

For more information, see Offline Administration and Offline Cockpit.

GPS Tracking FastTab: here you can configure the GPS Tracking parameters.

- **Minimum Distance (in meters):** this setting defines the minimum motion in meters after which the GPS coordinates are recorded by the MobileNAV client app.
  Note: If the value is '0', the GPS coordinates are recorded even without any movement.
- **Time Interval (seconds) (approximately):** this setting defines the time interval in seconds of recording the GPS coordinates.
- **Upload Interval (seconds):** this setting defines the time interval in seconds of uploading the recorded GPS coordinates to the server in online mode. In offline mode, the GPS coordinates are uploaded during synchronization.

Color FastTab: here you can change the colors used by the MobileNAV client app to better match your company's branding. Each color has an Area, which specifies where that color will be applied. The following Areas are available:

- **Normal:** This color type defines the base color of the application.
- **Dark:** This color type defines the dark version of the base color to be able to draw a gradient color.
- **Border:** This color defines the border color defines the border of the different windows in the application.
- **Background:** This color defines the background color of the different windows.
- **FirstLine:** This color defines the color of the fonts for the first line on a list, or the description of a card field.
- **Title:** This color defines the base color of the fonts in the application.

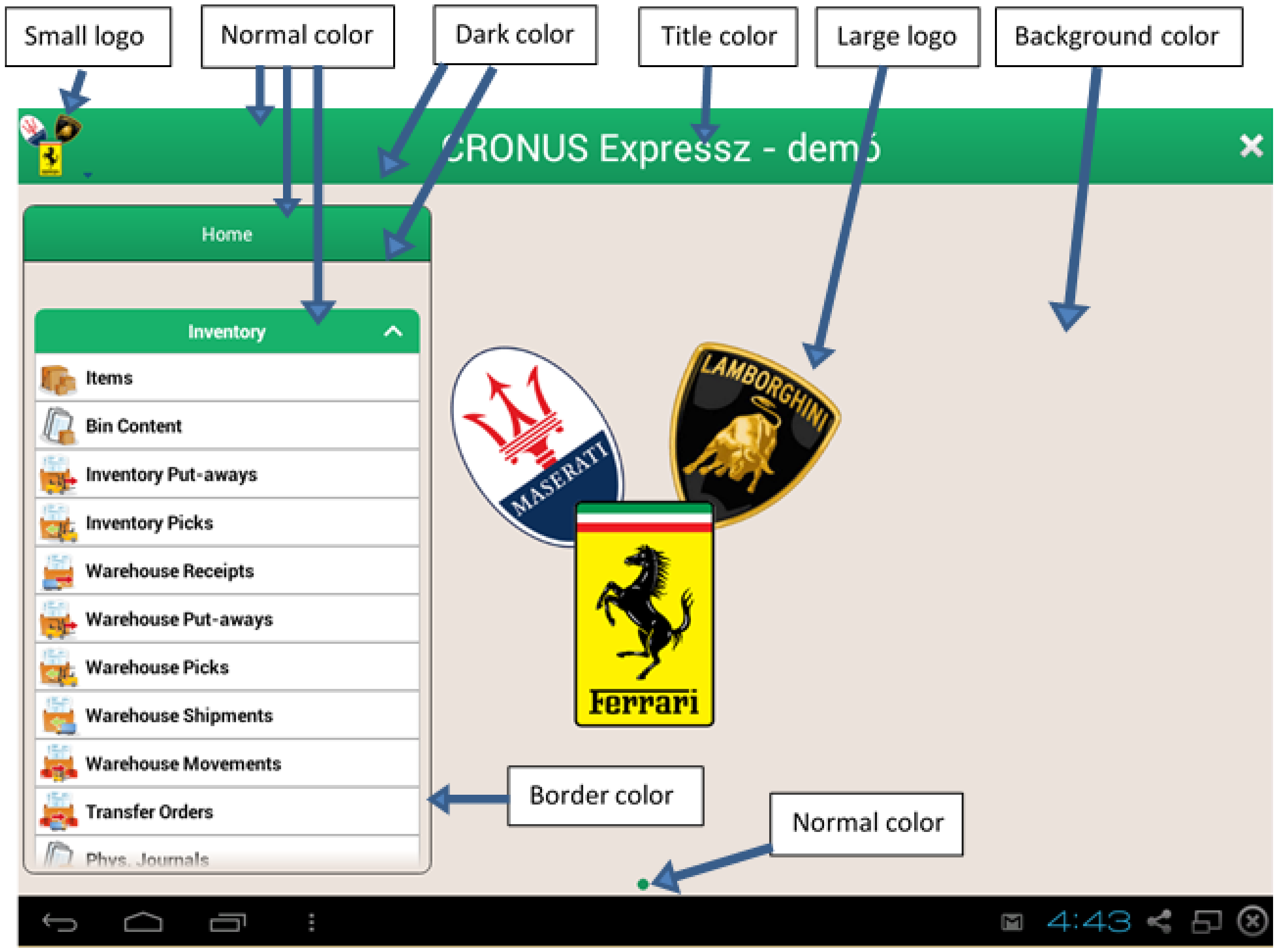
If you does not have any color setting, then the default will be the original "MobileNAV blue style".

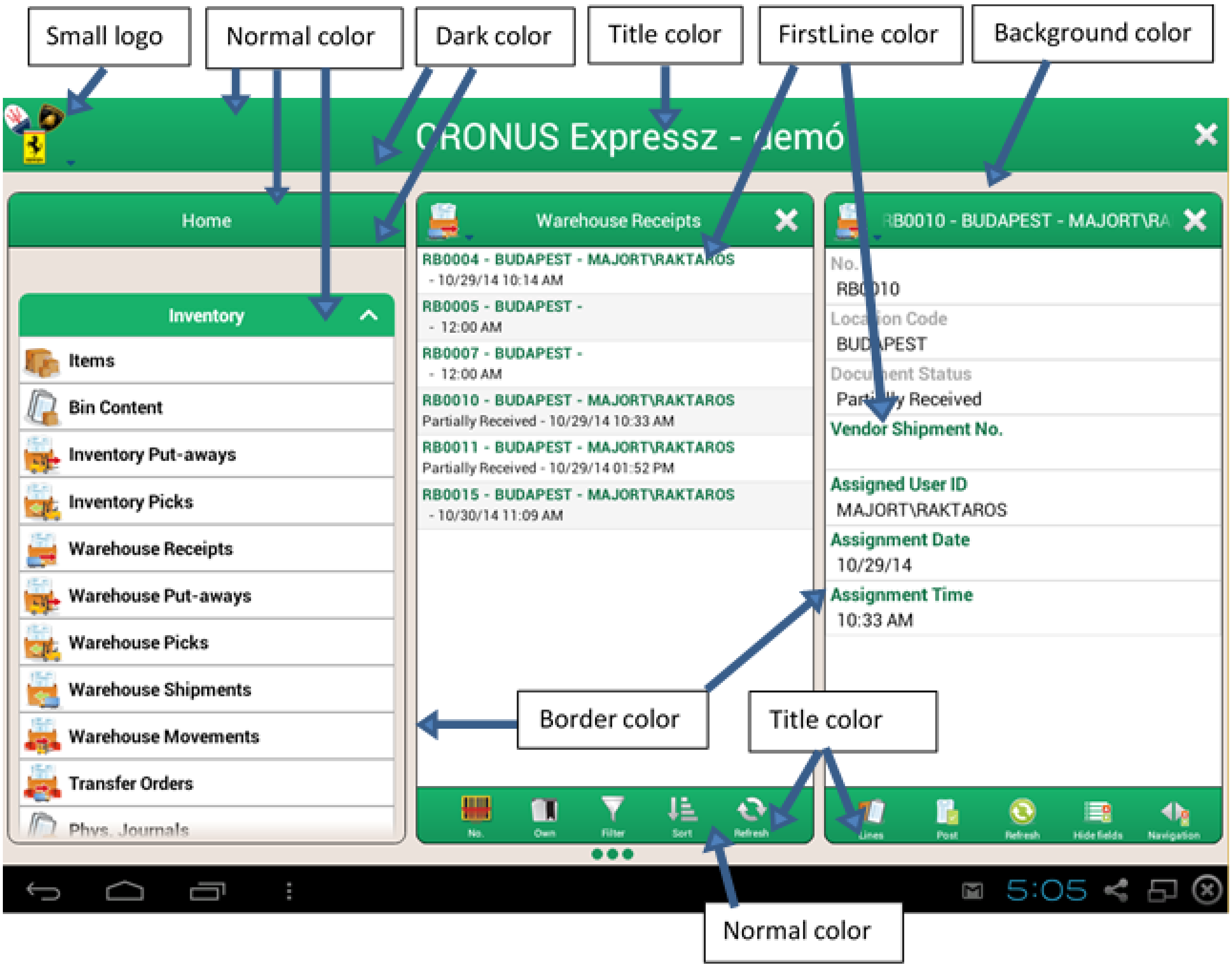
Picture FastTab: here you can change the default MobileNAV logo used by the client app in the upper left corner and in the middle of the tablet screen.

- **Small Picture:** The resolution of the small image should be 64 x 64 picture in PNG format.
- **Large Picture:** The resolution of the large image should be 512 x 512 in PNG format.

Version information FastTab: here you can view information about version numbers.

The following pictures show you an overview of the colors and logo in the MobileNAV application:


Small logo
Normal color
Dark color
Title color
Large logo
Background color
CRONUS Expressz - demó
Home
Inventory
Items
Bin Content
Inventory Put-aways
Inventory Picks
Warehouse Receipts
Warehouse Put-aways
Warehouse Picks
Warehouse Shipments
Warehouse Movements
Transfer Orders
Phys. Journals
Border color
Normal color
4:43



RB0004 - BUDAPEST - MAJORT\RAKTAROS
- 10/29/14 10:14 AM

RB0005 - BUDAPEST -
- 12:00 AM

RB0007 - BUDAPEST -
- 12:00 AM

RB0010 - BUDAPEST - MAJORT\RAKTAROS
Partially Received - 10/29/14 10:33 AM

RB0011 - BUDAPEST - MAJORT\RAKTAROS
Partially Received - 10/29/14 01:52 PM

RB0015 - BUDAPEST - MAJORT\RAKTAROS
- 10/30/14 11:09 AM

No.
RB0010

Location Code
BUDAPEST

Document Status
Partially Received

Vendor Shipment No.

Assigned User ID
MAJORT\RAKTAROS

Assignment Date
10/29/14

Assignment Time
10:33 AM