

# MODULE 8: DIMENSIONS

## Module Overview

In Microsoft Dynamics NAV® 2013, dimensions are configurable attributes that accompany all master records, documents, journals, posted documents, and ledger entries. Users can define their own dimension structures. Many reporting features use dimensions to enable simple multidimensional analysis of data as it is needed.

Dimensions are a core feature of Microsoft Dynamics NAV 2013. You must integrate dimensions into any new module that you create that involves any of the processes where dimensions are present in the standard application.

When you add new master records, documents, journals, or ledgers, you must make sure that those new record types provide the same dimension functionality as other application areas, such as Sales & Receivables, or General Ledger. This not only safeguards a consistent user experience across the application, but enables users to analyze and understand the data that is generated by a custom module, as they do with the standard application.

Seminar Management application uses many standard application concepts where dimensions are usually present. Therefore, you must extend the Seminar Management functionality to include dimensions functionality in the following areas: master records, documents, and ledger entries.

### Objectives

- Describe Global, Shortcut, and Budget dimension types and their functions.
- List the basic rules of Dimension Setup.
- Present the dimension management data and process models.

Implement dimensions on the master record level.

### Prerequisite Knowledge

A *dimension* is additional information that is attached to records, primarily for analysis. Table fields are static, unchangeable attributes of records and are predetermined by the developer. Whereas, dimensions are dynamic, configurable attributes that the user selects and sets up.


A company's accounts consist of many entries from many sources. Accounts are associated with many activities within the company. It is frequently necessary to create statements, statistics, and analyses that are extracts of the complete financial statements. These extracts can be created by using individual dimensions or combinations of dimensions.

If you set up a dimension called Department, and then use this dimension and a dimension value when you post an entry, you can retrieve information later, such as items that were sold and the departments that sold them. If more than one dimension was used on posted entries, the user can create a richer analysis of a company's activities. For example, a single sales entry can include multiple dimension information about the account to which the item sale was posted, where the item was sold, who sold it, and the type of customer who made the purchase.

By using dimensions, you can analyze trends and compare various characteristics across a range of entries. The analysis view functionality is especially effective for this purpose. However, you can also use filters, account schedules, and reports to create informative dimensions analyses.

You can also use dimensions to support business rules by influencing how a user can combine dimensions and how dimensions can be posted. This may be useful if certain departments cannot use particular accounts or sell to particular customers.

---

 **Additional Reading:** For more information about supporting business rules through dimensions, see *Dimension Combination Table and Default Dimension Table* topics in Microsoft Dynamics NAV help.

---

Many different types of records in Microsoft Dynamics NAV 2013 can have dimensions. The following types of records in Microsoft Dynamics NAV 2013 always have dimensions:

- Master records
- Document headers and lines
- Posted document headers and lines
- Journal lines
- Ledger entries

**Note:** There are few exceptions to this rule. Two important exceptions are the Reminders and Finance Charge Memos, where dimensions are only used at the document header level, not in the lines. This is because neither reminders nor finance charge memos are a core process. They both require analysis only at the header level. Any detailed analysis is available through other tables that are related to the process.

Nevertheless, when providing new functionality, such as supporting a core process of a company, such as Seminar Management, you should provide full dimension management functionality.

Each dimension can have an unlimited number of dimension values. For example, a dimension called Department can have dimension values of Sales, Administration, and so on. Dimensions can also be hierarchical with multiple levels of values organized in parent/child relationships. Dimensions can also include totaling groups that have headers and footers.

For example, the AREA dimension in the demo database for Microsoft Dynamics NAV 2013 has the following value structure. See the "Dimension Values for the AREA Dimension figure."

Code	Name	Dimension ...	Totaling	B.
10	Europe	Begin-Total		<input type="checkbox"/>
20	Europe North	Begin-Total		<input type="checkbox"/>
30	Europe North (EU)	Standard		<input type="checkbox"/>
40	Europe North (Non EU)	Standard		<input type="checkbox"/>
45	Europe North, Total	End-Total	20.45	<input type="checkbox"/>
50	Europe South	Standard		<input type="checkbox"/>
55	Europe, Total	End-Total	10.55	<input type="checkbox"/>
60	America	Begin-Total		<input type="checkbox"/>
70	America North	Standard		<input type="checkbox"/>
80	America South	Standard		<input type="checkbox"/>
85	America, Total	End-Total	60.85	<input type="checkbox"/>

**FIGURE 8.1: DIMENSION VALUES FOR THE AREA DIMENSION**

Users define and customize dimensions and their values to meet their company's needs and business process requirements.

### **Dimension Types**

Users can define as many dimensions as they need in their company. There can be an unlimited number of dimension values for each dimension. You define the values to use on entries in journals and documents and in dimensions-related reports and batch jobs.

Following are the three most important dimension types:

- Global dimensions
- Shortcut dimensions
- Budget dimensions

### **Global Dimensions**

You can configure two dimensions as global dimensions. Users configure the global dimensions in the General Ledger Setup card. The global dimensions are available throughout Microsoft Dynamics NAV 2013, and are stored as physical attributes (fields) on the tables that they describe. This lets users easily filter the entries, or use global dimension filters on reports, batch jobs, account schedules, and so on.



**Note:** The tables that use global dimensions always have the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields. Users never see these captions. Instead, Microsoft Dynamics NAV automatically translates the captions for global dimensions into the captions that the user has configured in the dimension setup, by calling the **CaptionClassTranslate** function of the codeunit 1, *ApplicationManagement*.

**CaptionClassTranslate** function then calls the codeunit 42, *CaptionManagement*. This reads the dimension caption from the **Dimension** table, and shows it to the user. Dimension captions are cached, and you must restart the application if you change the dimension caption in the setup.

---

Global dimensions are typically a company's most used and important dimensions because of their availability throughout the program. Because these dimensions are present as actual fields in tables, users cannot change these dimensions directly. There is a batch job that changes the global dimensions and then updates all records. This process can take some time.

### Shortcut Dimensions

Two global and up to six additional dimensions (a total of eight dimensions) can be shortcut dimensions. Unlike global dimensions, these dimensions are not present as actual fields in tables, but are displayed as fields in document and journal pages. This enables users to enter dimension values directly on the lines in journals or documents. This saves time and increases productivity.

When users enter dimension values for records, they must do the following: open additional pages, enter the dimensions in multiple lines, and then close the page to return to the original record. For example, to enter dimensions on a sales order line, you must click **Line > Dimensions**, or press SHIFT+CTRL+D. Then in the **Edit Dimension Set Entries** page, you must configure multiple dimension values as multiple different lines by first selecting the dimension, and then the dimension value. It is time consuming to enter multiple dimensions for many records. Shortcut dimensions make users more productive by displaying additional dimension columns in journals and document lines pages. Users can enter dimension values for up to eight configured shortcut dimensions.



**Note:** *The first two shortcut dimensions, that are also global dimensions, exist as actual fields in tables. These dimensions are called **Shortcut Dimension 1 Code** and **Shortcut Dimension 2 Code**. The other six shortcut dimensions do not exist as fields in tables. However, they are present as field controls in pages that use shortcut dimensions. These field controls are bound to elements of an array that is usually called `ShortcutDimCode`. Trigger code on all shortcut dimension field controls makes sure that when users enter values into the controls, that an actual dimension set is stored in the **Dimension Set Entry** table.*

*Captions for shortcut dimension field controls are translated exactly like global dimensions.*

---

Because shortcut dimensions 3 through 8 are not present as actual fields in tables, users can only filter on the first two shortcut dimensions.

### Budget Dimensions

When you enter budgets, you typically bind budget entries to specific dimensions, such as projects, or departments. Each budget can use up to four dimensions. Budget dimensions do not have relationships with global dimensions, and users can use any four dimensions as budget dimensions, regardless of which dimensions are configured as global dimensions.

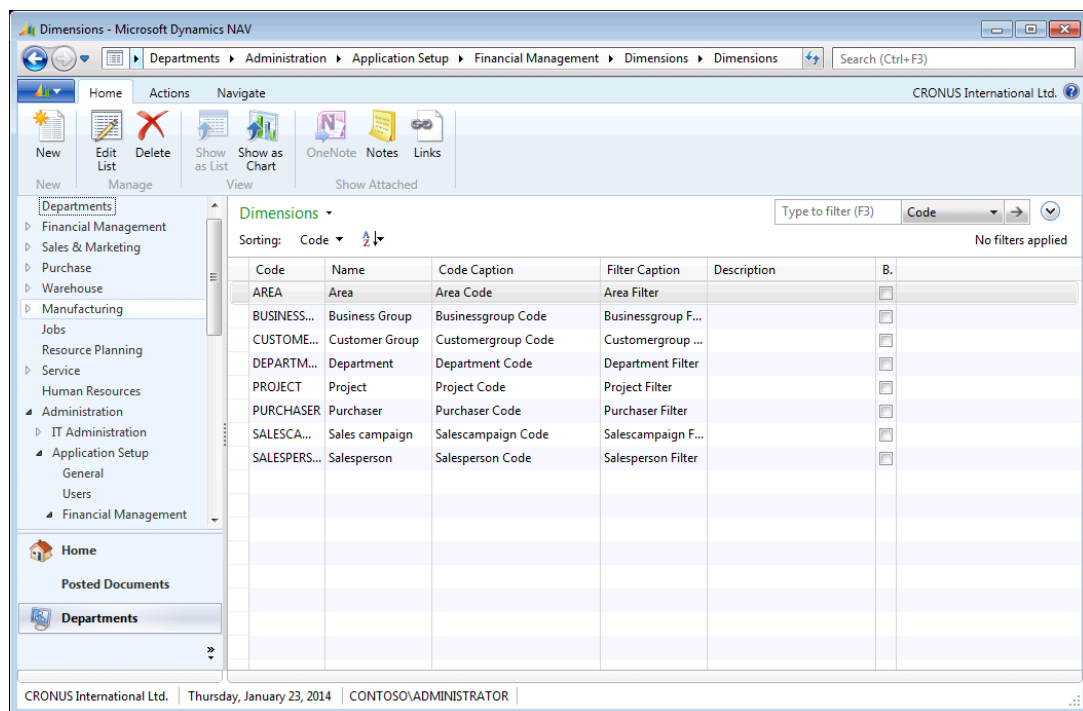
## Dimension Setup

Users must first configure dimensions in Microsoft Dynamics NAV 2013. Configuring dimensions involves two steps: configuring the dimensions and dimension values, and configuring the global and shortcut dimensions for the application.

## Configuring Dimensions

Users configure dimensions in the Administration area in Departments, under **Administration > Application Setup > Financial Management > Dimensions > Dimensions**. For each dimension, users must provide the **Code**, **Name**, **Code Caption**, and **Filter Caption** fields.

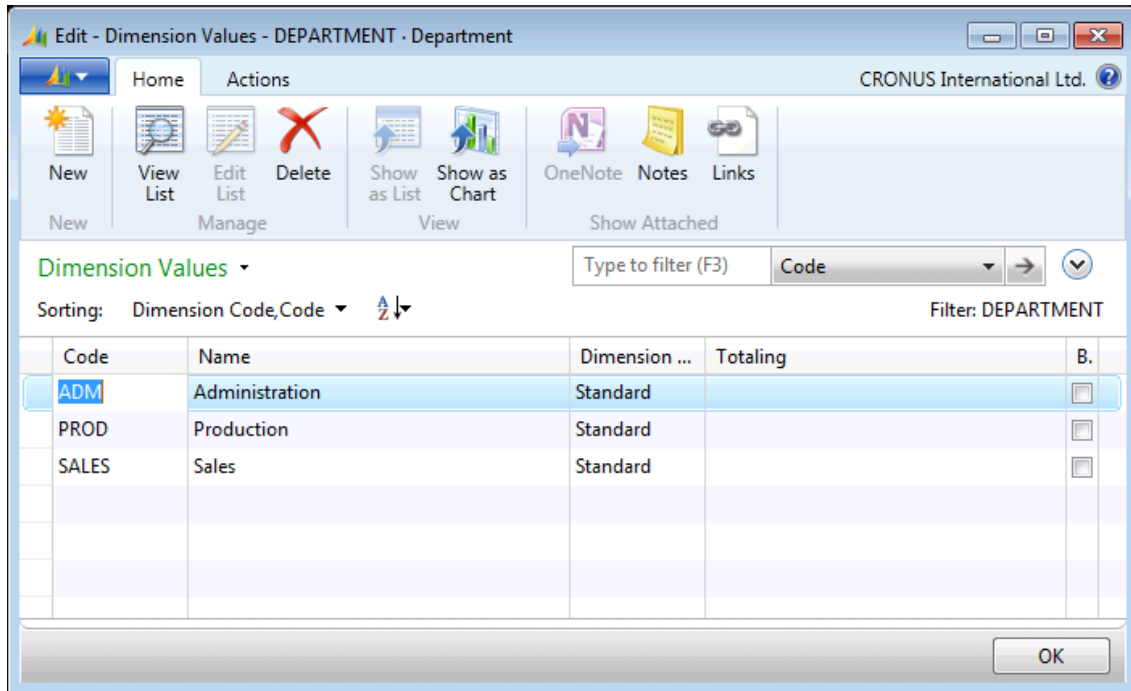
The “Dimensions List Page” figure shows the Dimensions list in the Microsoft Dynamics NAV 2013 client for Windows.




**FIGURE 8.2: THE DIMENSIONS LIST PAGE**

Each dimension contains a series of dimension values. These can be organized as lists, or as hierarchies. To configure the dimension values, in the **Dimension List** page, click **Navigate > Dimension Values** to open the **Dimension Values** page. For each dimension value, users must provide the **Code**, **Name** and **Dimension Value Type** fields.

The “Dimension Values Page” figure shows the **Dimension Values** page for the DEPARTMENT dimension.



**FIGURE 8.3: THE DIMENSION VALUES PAGE**

 **Additional Reading:** For more information about how to configure dimensions, dimension values, and dimension value hierarchies, refer to the *How to: Set Up Dimensions and Dimension Values* topic in the Microsoft Dynamics NAV help, or to the *Setup Dimensions* chapter of the *80435 Application Setup in Microsoft Dynamics NAV 2013* course.

### Configuring Global and Shortcut Dimensions

Users configure global dimensions and shortcut dimensions on the **Dimensions** FastTab of the **General Ledger Setup** page, under **Departments > Administration > Application Setup > Financial Management > Finance > General Ledger Setup**.

For the Microsoft Dynamics NAV 2013 demonstration database, the global dimensions are set to the DEPARTMENT and PROJECT dimensions, as shown in the “Dimensions FastTab on the General Ledger Setup Page” figure.

If Global Dimension 1 and Global Dimension 2 are set up in the system, they now link to these two dimensions. For example, the **Global Dimension 1 Code** field in the **Customer** table will now have a caption of Department Code, because this dimension is set up as the Global Dimension 1 Code in General Ledger Setup.

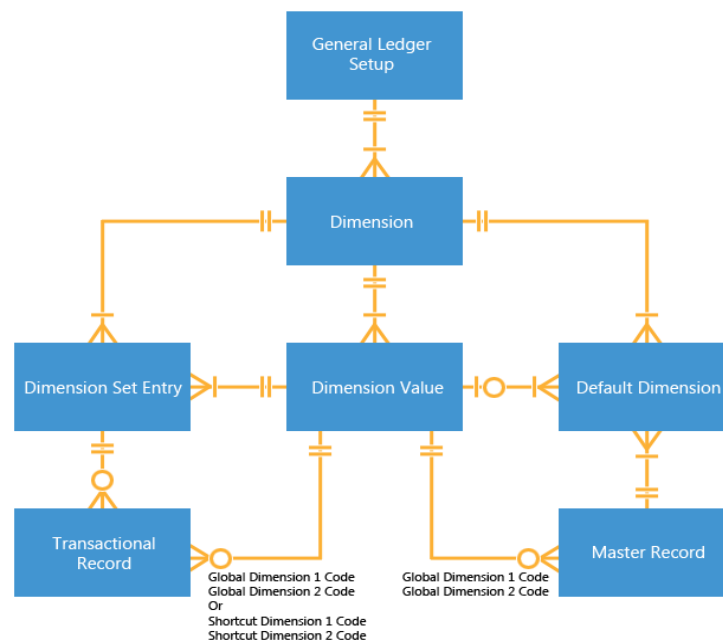
To change the two global dimensions, in the **General Ledger Setup** page, click **Actions > Change Global Dimensions**. This opens the Change Global Dimensions batch job that changes the values in the **General Ledger Setup** page. It also updates the values of the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields in all tables where they are present.

**Note:** Because the first two shortcut dimensions are always equal to the global dimensions, you cannot change the values of the **Shortcut Dimension 1 Code** and **Shortcut Dimension 2 Code** fields manually, and running the Change Global Dimensions batch job also changes the values of these two fields.

### Dimensions Data Model

Microsoft Dynamics NAV 2013 uses several tables to store dimensions, their values, and the relationships between dimensions and various types of records. Regardless of its functionality and the business processes it supports, every application module that requires dimensions uses the same underlying data model to support dimension management.

The "Dimension Management Data Model" figure shows the data model of the dimension management functionality in Microsoft Dynamics NAV 2013.



**FIGURE 8.4: DIMENSION MANAGEMENT DATA MODEL**



### Dimension and Dimension Value Tables

The **Dimension** table stores the dimensions that you can use, and the **Dimension Value** table stores the values for those dimensions. If users want to enable analysis by an attribute that is not present in the data model, they can define a new dimension, and its values.

Suppose that a company wants to track sales by sales channel, such as retail, wholesale, or Internet. The company may define a new dimension called CHANNEL, with values of RETAIL, WHOLESALE, and INTERNET. This information is stored in the **Dimension** and **Dimension Value** tables.

### Default Dimension Table

The **Default Dimension** table stores the default dimension values that are attached to a specific master record, such as a Customer, a G/L Account, or an Item. When you specify default dimensions for master records, the program suggests these dimensions and dimension values when the account is used in a transaction, for example on a journal line or a document. This makes entry posting easier for the user, as the program completes the dimension fields automatically. These dimension values are the default settings. Users can always change the default dimension values that are suggested by the program.



**Note:** The **Default Dimension** table also specifies how a dimension value is used during posting of transactions.

---

The **Default Dimension** table uses the **Table ID** and **No.** fields to specify to which table and record in that table that a default dimension applies. If the **No.** field is empty, then the configured dimension and dimension value are the default for all records in the table that is specified in the **Table ID** field. This feature is called Account Type Default Dim. Users can access it from the Account Type Default Dim. action in the **Navigate** tab of the ribbon on the **Dimensions** list page.



**Note:** When you configure the Account Type Default Dim., the default dimensions are not assigned to all new records of the specified account type (master record type) when you create new master records. Instead, when you use an account type inside journals or documents, the default dimensions are copied from both the specified master record, such as Customer, and from the default dimensions for the master record type. If the master record type is customer, then those default dimensions have **Table ID** field of 18, and empty **No.** field

---

### Dimension Set Entry Table

The **Dimension Set Entry** table stores combinations of dimension values. Each combination of dimension values is assigned a **Dimension Set ID** field. Any transactional record, such as a document header, document line, journal line, posted document header, posted document line, or a ledger entry, that uses the same combination of dimension values, has the same value in the **Dimension Set ID** field (typically, this field has Field No. of 480).

For example, a journal line may use the following combination of dimension values.


Dimension Code	Dimension Value Code
AREA	30
CUSTOMERGROUP	MEDIUM
DEPARTMENT	SALES

This combination of dimensions may be stored in the Dimension Set Entry table as follows.

Dimension Set ID	Dimension Code	Dimension Value Code	Dimension Value ID	Dimension Name	Dimension Value Name
3	AREA	30	17	Area	Europe North (EU)
3	CUSTOMERGROUP	MEDIUM	9	Customer Group	Medium Business
3	DEPARTMENT	SALES	2	Department	Sales

The journal line then stores the value of 3 in the **Dimension Set ID** field. If any other journal line, ledger entry, or any other transactional record requires the same combination, it will reuse the existing one, and set its **Dimension Set ID** field to the same value of 3.

---

 **Note:** By assigning IDs to dimension sets, and reusing those IDs when specific combinations of dimension values are used, Microsoft Dynamics NAV 2013 improves the application performance by reducing storage requirements and posting time.

---

### Master Tables

All master tables have the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields, which have table relation to the **Dimension Value** table.

### Transactional Record Tables

Transactional record tables for open (not-yet-posted) transactions, including document header, document line, journal line tables, and posted document header and lines tables, have the **Shortcut Dimension 1 Code** and **Shortcut Dimension 2 Code** fields. These fields have the same relation to the **Dimension Value** table as global dimension fields have to master tables. For all tables except posted document tables, the OnValidate trigger on these fields calls the appropriate functionality in the DimensionManagement codeunit. This makes sure that values that are entered by users are valid, and it calculates appropriate dimension sets from dimension value combinations that are entered for the record.

All transactional record tables, including all not-yet-posted tables and ledger entry tables, have the additional **Dimension Set ID** field. This field contains the value of the dimension set in the **Dimension Set Entry** table that corresponds to the combination of dimension values that the user has entered for the record.

### Dimension Process Model

In Microsoft Dynamics NAV 2013, the dimension process model comprises a broad set of functionality for maintaining dimension data. The model includes page actions, table and page triggers and functions, and the codeunit 408, DimensionManagement.

When you create new master, journal, document, or ledger tables and pages, you must make sure that you integrate your processes with dimension management. Do this by calling the DimensionManagement codeunit in all locations where a standard application would call it.

### Master Tables

The OnInsert trigger in master tables calls the **UpdateDefaultDim** function of DimensionManagement to make sure that any default dimensions that are assigned to a master record type are inserted for a new master record. Global dimension fields are updated accordingly.

In a similar fashion, the OnDelete trigger calls the **DeleteDefaultDim** function of the DimensionManagement codeunit to delete all default dimensions for a master record when the master record is deleted.

All master tables contain the `ValidateShortcutDimCode` function. This function validates the user's entry in the field, and stores default dimension values for the master record into the **Default Dimension** table. It does this by calling the **ValidateDimValueCode** and **SaveDefaultDim** functions of the `DimensionManagement` codeunit.

This function is called from the `OnValidate` trigger of the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields.

### Document Header Tables

The document header tables contain comprehensive business logic for managing dimensions. Document header tables define the following functions to manage dimensions.

Function	Remarks
<b>CreateDim</b>	<p>Specifies up to ten master record types and numbers for which dimensions should be copied into the document. The <b>CreateDim</b> function then calls the <b>GetDefaultDimID</b> function of the <code>DimensionManagement</code> codeunit, and updates the value of the <b>Dimension Set ID</b> field.</p> <p>For example, in the <b>Sales Header</b> table, it takes the default dimensions from the Customer, Salesperson, Campaign, Responsibility Center, and Customer Template that are contained in the sales document to calculate the <b>Dimension Set ID</b> field for the sales document.</p> <p>This function is typically called from the <code>OnValidate</code> trigger of the field that is related to the most important master record of the document, such as Bill-to Customer No. in the <b>Sales Header</b> table.</p>
<b>ValidateShortcutDimCode</b>	<p>Calls the <b>ValidateShortcutDimValues</b> function of <code>DimensionManagement</code> codeunit to calculate the new <b>Dimension Set ID</b> field, and then changes the record if it is necessary. It also calls the <b>UpdateAllLineDim</b> function.</p> <p>It is called from the <code>OnValidate</code> trigger of the <b>Shortcut Dimension 1 Code</b> and <b>Shortcut Dimension 2 Code</b> fields.</p>

Function	Remarks
<b>ShowDocDim</b>	<p>Calls the <b>EditDimensionSet2</b> function of the DimensionManagement codeunit, then updates the <b>Dimension Set ID</b> field if it is necessary, and calls the <b>UpdateAllLineDim</b> function.</p> <p>This function is called from the OnLookup trigger of the <b>Dimension Set ID</b> field, and from the OnAction trigger for the Dimensions action on the document page.</p>
<b>UpdateAllLineDim</b>	<ul style="list-style-type: none"> <li>• Asks the user for the confirmation, and then does the following for each line:</li> <li>• Calls the <b>GetDeltaDimSetID</b> function of the DimensionManagement codeunit.</li> <li>• Updates the <b>Dimension Set ID</b> field.</li> <li>• Calls the <b>UpdateGlobalDimFromDimSetID</b> function of the DimensionManagement codeunit.</li> </ul>

### Document Line Tables

Similar to document header tables, the document line tables have a similar set of functions for managing dimensions that call the DimensionManagement codeunit when users change or access dimension information that is related to the line.

The following table shows the dimension management functions that are typically defined in a document line table.

Function	Remarks
<b>CreateDim</b>	Performs the same logic as the <b>CreateDim</b> function on document header tables, except it is only for the document line.
<b>ShowDimensions</b>	Achieves the same goal as the <b>ShowDocDim</b> function on document header tables.
<b>ValidateShortcutDimCode</b>	Calls the <b>ValidateShortcutDimValues</b> function of the DimensionManagement codeunit to calculate the new <b>Dimension Set ID</b> field.

Function	Remarks
<b>LookupShortcutDimCode</b>	<p>Calls the <b>LookupDimValueCode</b> function of the DimensionManagement codeunit, and then calls the <b>ValidateShortcutDimCode</b> function.</p> <p>This function is called from the OnLookup triggers of shortcut dimension field controls on document subpages, such as Sales Order Subform.</p>
<b>ShowShortcutDimCode</b>	<p>Calls the <b>GetShortcutDimensions</b> function of the <b>DimensionManagement</b> codeunit.</p> <p>It is called from the OnAfterGetRecord trigger on the page that displays the document line.</p>

### Journal Line Tables

Journal line tables resemble document line tables. There are very few differences between the two table types. Documents have headers, whereas journals do not.

### Posted Document and Ledger Entry Tables

Users cannot usually change tables that contain posted information, which includes posted documents and ledger entries. Therefore, the tables do not contain much business logic. ShowDimensions is the only function that calls the ShowDimensionSet function of the DimensionManagement codeunit. This function is called when users look up the value in the Dimension Set ID field, or click the Dimensions action in a relevant posted document, posted document lines, or ledger entries page.

### Master Record Pages

All card pages for master records contain an action named Dimensions. This action is located in the **Navigate** tab in the ribbon and shows the dimensions that are attached to the master record.



**Note:** Even though master record tables always include the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields, these fields are not shown on the master card pages in standard Microsoft Dynamics NAV 2013. If users require these fields, you can add them to master card pages.

On list pages for master records, the following two actions are located in the **Navigate** tab in the ribbon:

- **Dimensions-Single** – Shows and lets the user edit dimensions that are attached to the single selected master record.
- **Dimensions-Multiple** – Shows and lets the user edit dimensions that are attached to all master records in the selection.



**Note:** To enable the *Dimensions-Multiple* feature for a new master record type, you must customize the page 542, **Default Dimensions-Multiple**, by adding a new function that copies the default dimensions from the new master record type.

---

### Document and Posted Document Pages

Similar to card type pages for master records, all document and list type pages for documents contain an action named Dimensions. This action is located in the **Navigate** tab in the ribbon and shows the dimensions that are attached to the master record.

Document subpages contain the following dimension management functionality:

- An action named Dimensions in the Line action group shows the dimensions for the line in the edit mode.
- The OnValidate and OnLookup triggers of the Shortcut Dimension field controls call the **ValidateShortcutDimCode** and **LookupShortcutDimCode** functions of the underlying document line table.
- The OnAfterGetRecord trigger and the OnValidate trigger of fields that are related to the primary master record for the line (such as **No.** in the **Sales Line** table), call the **ShowShortcutDimCode** function of the underlying document line table.
- The OnNewRecord trigger clears the ShortcutDimCode array that keeps values for the shortcut dimensions of the current document line record.

Pages that show posted documents include the **Dimensions** action, which shows the dimensions for the posted document, or document line. Posted document pages show the action in the **Navigate** tab in the ribbon, whereas document subpages show the action in the Line action group.


### Journal and Ledger Entry Pages

The journal pages contain the same functionality as the document subpages, except that they show the **Dimensions** action in the **Navigate** tab of the ribbon. Ledger entry pages only contain the **Dimensions** action in the **Navigate** tab of the ribbon, and do not have any other functionality that is related to dimensions.

## DimensionManagement Codeunit

The codeunit 408, DimensionManagement is at the core of the dimension management process model in Microsoft Dynamics NAV 2013. This is where most of the dimension data processing occurs. Functions in this codeunit handle several groups of processes, such as managing dimension sets, checking posting rules, managing default dimensions, and so on. All tables and pages that include global or shortcut dimension fields, or enable users to show, edit, or look up dimension information, reference this codeunit and call its functions.


You must also call the following important functions from any custom-built master, document, journal, or ledger page.

Function	Remarks
<b>UpdateDefaultDim</b>	Makes sure that any default dimensions that are assigned to a specific table (master record type) are inserted for a new master record. It also updates any affected global dimension fields.
<b>DeleteDefaultDim</b>	Deletes all default dimensions for a master record when the master record is deleted.
<b>ValidateDimValueCode</b>	<p>Validates whether a value that is specified for a dimension is a valid value. If it cannot find the value that the user has entered, it tries to find the first dimension value that begins with the user's entry. If it finds it successfully, it substitutes the user's entry with the actual dimension value. In all other cases, it fails with an error.</p> <p>This function is called from all OnValidate triggers for global and shortcut dimension fields on tables and field controls on pages.</p> <hr/> <p> <b>Note:</b> For master record tables, the OnValidate trigger actually calls the <b>ValidateShortcutDimCode</b> function of the table. This in turn calls the <b>ValidateDimValueCode</b> function of the codeunit 408.</p>



Function	Remarks
<b>SaveDefaultDim</b>	<p>Inserts or updates an entry in the <b>Default Dimension</b> table for a master record when a user enters a value in any of the global dimension fields.</p> <p>It is called from the <b>ValidateShortcutDimCode</b> function in the master table, which is called from the OnValidate trigger of the <b>Global Dimension 1 Code</b> and <b>Global Dimension 2 Code</b> fields.</p>
<b>GetDefaultDimID</b>	<p>Uses up to ten master record types and numbers and an existing dimension set ID. Then it calculates the new dimension set ID from the combination of the existing dimension set and any of the ten dimension values that are not already included in the set. Finally, it returns the new dimension set ID to the caller.</p> <p>This function is typically called from <b>CreateDim</b> functions of document header tables.</p>
<b>ValidateShortcutDimValues</b>	<p>Calls the <b>ValidateDimValueCode</b> function, and then calculates the new dimension set.</p> <p>It is called from the <b>ValidateShortcutDimCode</b> function on document header tables when users change the value for one of the shortcut dimensions.</p>
<b>ShowDimensionSet</b>	<p>Runs the page 479, <b>Dimension Set Entries</b> to show all dimension values that consist of a dimension set in the read-only mode.</p> <p>This function is typically called from the <b>ShowDimensions</b> function on ledger entry or posted document line tables.</p>

Function	Remarks
<b>EditDimensionSet</b>	<p>Shows the page 480, <b>Edit Dimension Set Entries</b> to show all dimension values that consist of a dimension set in the edit mode. Then it calculates the new dimension set ID for the dimension values that were edited by the user.</p> <p>This function is typically called from the <b>ShowDimensions</b> function on document line tables.</p>
<b>EditDimensionSet2</b>	<p>Does the same as the <b>EditDimensionSet</b> function, except that it also updates the global dimension values through a call to the <b>UpdateGlobalDimFromDimSetID</b> function.</p> <p>This function is typically called from the <b>ShowDocDim</b> function on document header tables.</p>
<b>GetDeltaDimSetID</b>	<p>Combines the document header dimensions with document line dimensions and calculates and returns a new resulting dimension set ID.</p> <p>It is called from the <b>UpdateAllLineDim</b> function of document header tables when users update header dimensions and want to apply the changes to all document lines.</p>
<b>UpdateGlobalDimFromDimSetID</b>	<p>Updates values for <b>Global Dimension 1 Code</b> and <b>Global Dimension 2 Code</b> fields (or the first two shortcut dimensions) for a record by reading the values from the specified dimension set.</p> <p>This function is typically called from the <b>UpdateAllLineDim</b> function in document header tables.</p>
<b>LookupDimValueCode</b>	<p>Shows the available dimension values for the specified dimension to let the user select a value for a shortcut dimension field.</p> <p>This function is called from the <b>LookupShortcutDimCode</b> function on document line tables.</p>

Function	Remarks
<b>GetShortcutDimensions</b>	<p>Populates the array of values for eight shortcut dimensions from a dimension set ID.</p> <p>It is called from the <b>ShowShortcutDimCode</b> function of document line tables.</p>
<b>SetupObjectNoList</b>	<p>Populates the list of all account types (master record types) in the application. This lets users select the table type in the <b>Account Type Default Dim</b> page.</p> <hr/> <p> <b>Note:</b> <i>Whenever you add new master record types, you must customize this function to enable users to define default account type dimensions on your custom master tables.</i></p> <hr/>
<b>CheckDimIDComb</b>	<p>Checks dimension value combinations of a dimension that is set for invalid combinations according to the configuration in the <b>Dimension Combination</b> and <b>Dimension Value Combination</b> tables.</p> <p>This function is called from Jnl.-Check Line codeunits of journal posting routines, and Post codeunits of document posting routines.</p>
<b>CheckDimValuePosting</b>	<p>Checks dimension values of a dimension set against the <b>Value Posting</b> field settings for default dimensions of master records that are involved in the posting.</p> <p>This function is called from Jnl.-Check Line codeunits of journal posting routines, and Post codeunits of document posting routines.</p>

Function	Remarks
<b>GetDimCombErr</b>	Returns the message that describes the dimension combination error.  This function is called from Jnl.-Check Line codeunits of journal posting routines, and Post codeunits of document posting routines.
<b>GetDimValuePostingErr</b>	Returns the message that describes the dimension posting value error.  This function is called from Jnl.-Check Line codeunits of journal posting routines and Post codeunits of document posting routines.

### Journal and Document Posting Codeunits

Several codeunits that are connected with posting routines perform specific dimension-related tasks and call functions of the DimensionManagement codeunit.

The following table explains the dimension management features of posting-routine codeunits.

Codeunit	Dimension Management Features
<b>Jnl.-Check Line</b>	Calls the <b>CheckDimIDComb</b> and <b>CheckDimValuePosting</b> functions of the DimensionManagement codeunit on the dimension set of the journal line.
<b>Jnl.-Post Line</b>	Copies values of <b>Shortcut Dimension 1 Code</b> , <b>Shortcut Dimension 2 Code</b> , and <b>Dimension Set ID</b> fields from the journal line table to <b>Global Dimension 1 Code</b> , <b>Global Dimension 2 Code</b> , and <b>Dimension Set ID</b> fields in the ledger entry table.

Codeunit	Dimension Management Features
<b>Post</b>	<p>Calls the <b>CheckDimIDComb</b> and <b>CheckDimValuePosting</b> functions of the DimensionManagement codeunit on the dimension sets of the document header and document lines.</p> <p>Copies values of <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields from the document lines to the appropriate journal lines, before it starts the journal posting routine for the journal line.</p>

## Integrating Seminar Management with Dimensions

In Microsoft Dynamics NAV 2013, dimensions serve two purposes: they enable the expansion of the existing data model with additional attributes that meet a company's business need for simple multi-dimensional analysis of data; and they support simple data-driven business rules that relate to data validity during posting.

When you extend the application by using new application areas or functionality such as master records, documents, or journals, you should take advantage of the standard dimension management functionality to give users a consistent user experience.

You have already developed many features for the new Seminar Management application area. You now must integrate the dimension management features with Seminar Management functionality.

### Solution Design

The CRONUS International Ltd. functional requirements do not include specific details about dimensions. The legacy application that they used prior to the implementation of Microsoft Dynamics NAV 2013 has no similar functionality, and CRONUS International Ltd. is unfamiliar with the concept of dimensions.

However, two of the following functional requirements for CRONUS International Ltd. can apply to dimensions:

- The solution must use the standard functionality of Microsoft Dynamics NAV 2013 wherever possible. The solution must not duplicate functions that are already present in the application or cause redundant functionality.

- The solution must be consistent, user-friendly, easy to learn, and to use. Any custom-built functionality must follow the standards, principles, and best practices of Microsoft Dynamics NAV 2013, and must seamlessly integrate into the standard application.



**Best Practice:** *Whether your customers explicitly ask for standard Microsoft Dynamics NAV 2013 functionality in a custom-built module, or if any nonfunctional requirements request full dependency on standard functionality, you must always provide a level of integration with standard features that is consistent with any other functionally similar application area. Even though users might not know what dimensions are before they implement Microsoft Dynamics NAV 2013, they will find dimension functionality elsewhere in the application. Any functionality that you develop must not be an exception.*

---

### Solution Development

To provide a consistent user experience and to integrate with dimension management features of standard Microsoft Dynamics NAV 2013, you must change many of the objects that you have previously developed, including tables, pages, and codeunits. You must extend both your data model to integrate with the standard dimension model, and your application features to integrate with the standard dimension process model, as explained in the Prerequisite Knowledge lesson.

The following table summarizes the changes that you must make to the existing standard and custom objects.

Object	Summary of Changes
Table 352, <b>Default Dimension</b>	<ul style="list-style-type: none"><li>• Add a new function to update the global dimension code on a seminar record.</li><li>• Change the <b>UpdateGlobalDimCode</b> function to call the new function when it is necessary.</li></ul>
Table 123456700 <b>Seminar</b>	<ul style="list-style-type: none"><li>• Add <b>Global Dimension 1 Code</b> and <b>Global Dimension 2 Code</b> fields.</li><li>• Add the <b>ValidateShortcutDimCode</b> function to validate a user's entry and to store the default dimension values.</li><li>• Change the OnInsert and OnDelete triggers to call DimensionManagement codeunit where you can.</li></ul>

Object	Summary of Changes
Table 123456710 <b>Seminar Registration Header</b>	<ul style="list-style-type: none"> <li>• Add <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields.</li> <li>• Add the <b>SeminarRegLinesExist</b> function.</li> <li>• Add the <b>CreateDim</b> function.</li> <li>• Add the <b>ValidateShortcutDimCode</b> function.</li> <li>• Add the <b>ShowDocDim</b> function.</li> <li>• Add the <b>UpdateAllLineDim</b> function.</li> </ul>
Table 123456711 <b>Seminar Registration Line</b>	<ul style="list-style-type: none"> <li>• Add <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields.</li> <li>• Add the <b>ShowDimensions</b> function.</li> <li>• Add the <b>CreateDim</b> function.</li> <li>• Add the <b>ValidateShortcutDimCode</b> function.</li> <li>• Add the <b>LookupShortcutDimCode</b> function.</li> <li>• Add the <b>ShowShortcutDimCode</b> function.</li> </ul>
Table 123456718 <b>Posted Seminar Reg. Header</b>	<ul style="list-style-type: none"> <li>• Add <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields.</li> <li>• Add the <b>ShowDimensions</b> function.</li> </ul>
Table 123456719 <b>Posted Seminar Reg. Line</b>	<ul style="list-style-type: none"> <li>• Add <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields.</li> <li>• Add the <b>ShowDimensions</b> function.</li> </ul>
Table 123456731 <b>Seminar Journal Line</b>	<ul style="list-style-type: none"> <li>• Add <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields.</li> </ul>

## C/SIDE Solution Development in Microsoft Dynamics® NAV 2013

Object	Summary of Changes
Table 123456732 <b>Seminar Ledger Entry</b>	<ul style="list-style-type: none"> <li>• Add <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields.</li> <li>• Add the <b>ShowDimensions</b> function.</li> </ul>
Page 542 <b>Default Dimensions-Multiple</b>	<ul style="list-style-type: none"> <li>• Add a new function to select default dimensions for specified seminar records.</li> </ul>
Page 123456700 <b>Seminar Card</b>	<ul style="list-style-type: none"> <li>• Add the <b>Dimensions</b> action to the <b>Navigate</b> tab of the ribbon.</li> </ul>
Page 123456701 <b>Seminar List</b>	<ul style="list-style-type: none"> <li>• Add the Dimensions-Single and Dimensions-Multiple actions to the <b>Navigate</b> tab of the ribbon.</li> </ul>
Page 123456710 <b>Seminar Registration</b>	<ul style="list-style-type: none"> <li>• Add the Dimensions action to the <b>Navigate</b> tab of the ribbon.</li> </ul>
Page 123456711 <b>Seminar Registration Subform</b>	<ul style="list-style-type: none"> <li>• Add the Dimensions action to the Line action group.</li> <li>• Add field controls for the eight shortcut dimensions.</li> <li>• Change the OnAfterGetRecord and OnNewRecord page triggers.</li> <li>• Change the OnValidate trigger for the Bill-to Customer No. field control.</li> </ul>
Page 123456713 <b>Seminar Registration List</b>	<ul style="list-style-type: none"> <li>• Add the Dimensions action to the <b>Navigate</b> tab of the ribbon.</li> </ul>
Page 123456721 <b>Seminar Ledger Entries</b>	<ul style="list-style-type: none"> <li>• Add the Dimensions action to the <b>Navigate</b> tab of the ribbon.</li> </ul>
Page 123456734 <b>Posted Seminar Registration</b>	<ul style="list-style-type: none"> <li>• Add the Dimensions action to the <b>Navigate</b> tab of the ribbon.</li> </ul>
Page 123456735 <b>Posted Seminar Reg. Subform</b>	<ul style="list-style-type: none"> <li>• Add the Dimensions action to the Line action group.</li> </ul>



Object	Summary of Changes
Page 123456736 <b>Posted Seminar Reg. List</b>	<ul style="list-style-type: none"> <li>• Add the Dimensions action to the <b>Navigate</b> tab of the ribbon.</li> </ul>
Codeunit 123456700 <b>Seminar-Post</b>	<ul style="list-style-type: none"> <li>• Add the <b>CheckDim</b> function.</li> <li>• Add the <b>CheckDimComb</b> function.</li> <li>• Add the <b>CheckDimValuePosting</b> function.</li> <li>• Copy the values from the <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields from the document lines to the appropriate journal lines.</li> </ul>
Codeunit 123456731 <b>Seminar Jnl.-Check Line</b>	<ul style="list-style-type: none"> <li>• Call the <b>CheckDimIDComb</b> and <b>CheckDimValuePosting</b> functions of the DimensionManagement codeunit.</li> </ul>
Codeunit 123456732 <b>Seminar Jnl.-Post Line</b>	<ul style="list-style-type: none"> <li>• Copy the values from the <b>Shortcut Dimension 1 Code</b>, <b>Shortcut Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields from the journal line to the <b>Global Dimension 1 Code</b>, <b>Global Dimension 2 Code</b>, and <b>Dimension Set ID</b> fields in the ledger entry.</li> </ul>



**Note:** When you develop the customizations in this chapter's lab, you implement some of the changes that are specified in the previous table and import several changes from external files.

## Lab 8.1: Integrating with Dimension Management

### Scenario

Viktor is the business systems developer who works for the Microsoft Dynamics partner company that is implementing Microsoft Dynamics NAV 2013 for CRONUS International Ltd. His task is to integrate existing seminar management module features with dimension management.

Viktor reuses existing code from other objects. He makes sure to extend the master records, documents, journal and the ledger entry tables of the Seminar Management module by mimicking the functionality that he finds in the **Customer, Sales Header, Sales Line, Res. Journal Line, and Res. Ledger Entry** tables, the **Sales Invoice, Posted Sales Invoice, and Resource Ledger Entries** pages, and the Res. Jnl.-Check Line, Res. Jnl-Post Line, and Sales Post codeunits.

Because there are many objects that require customization, Isaac helps Viktor by customizing a subset of the Seminar Management objects. Isaac also prepares the set of object files that Viktor will import and review later.



**Note:** *Even though the labs contain the steps to develop most of the dimension management functionality in Sales Management module from the ground up, you may take a look at the objects that are listed here to see how they integrate with the dimension management features of the standard Microsoft Dynamics NAV 2013 application.*

---

### Exercise 1: Extending Master Data with Dimensions

#### **Exercise Scenario**

Viktor starts by changing several standard objects to make them ready for integration of the Seminar Management module with dimension management features. Then he extends the master tables and pages by using required dimension management features.

### Task 1: Modify the DimensionManagement Codeunit

#### High Level Steps

1. Add the Seminar option to the list of available account types, in the **SetupObjectNoList** function of the DimensionManagement codeunit.

#### Detailed Steps

1. Add the Seminar option to the list of available account types, in the **SetupObjectNoList** function of the DimensionManagement codeunit.
  - a. Design codeunit 408 DimensionManagement.
  - b. Locate the **SetupObjectNoList** function.
  - c. In the local variables for the function, on the TableIDArray variable set the Dimensions property to 100.
  - d. In the SetupObjectNoList function trigger, at the end of the block where TableIDArray values are assigned and before the Object.SETRANGE statement, enter the following code.

```
//CSD1.00>  
TableIDArray[100] := DATABASE::Seminar;  
//CSD1.00<
```

- e. Compile, save, and then close the codeunit.

### Task 2: Modify the Seminar Table

#### High Level Steps

1. Add the **ValidateShortcutDimCode** function to the **Seminar** table to call the **ValidateDimValueCode** and **SaveDefaultDim** functions of the DimensionManagement codeunit.
2. Add the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields to the table 123456700 **Seminar**. Make sure that you establish correct table relations to the **Dimension Value** table.
3. Add code to the OnInsert and OnDelete triggers to call the UpdateDefaultDim and DeleteDefaultDim triggers of the DimensionManagement codeunit.

## Detailed Steps

1. Add the **ValidateShortcutDimCode** function to the **Seminar** table to call the **ValidateDimValueCode** and **SaveDefaultDim** functions of the DimensionManagement codeunit.
  - a. Design table 123456700, **Seminar**.
  - b. Add a new function and name it **ValidateShortcutDimCode**. The function receives the following parameters.

Var	Name	Data Type	Length
No	FieldNumber	Integer	
Yes	ShortcutDimCode	Code	20

- c. Add a new global variable for the DimensionManagement codeunit, and name it DimMgt.
- d. In the ValidateShortcutDimCode function trigger, enter the following C/AL code.

```
DimMgt.ValidateDimValueCode(FieldNumber,ShortcutDimCode);


DimMgt.SaveDefaultDim(DATABASE::Customer,"No.",FieldNumber,ShortcutDimCode);

MODIFY;
```

2. Add the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields to the table 123456700 **Seminar**. Make sure that you establish correct table relations to the **Dimension Value** table.
  - a. Add the following fields.

Field No.	Field Name	Data Type	Length
21	Global Dimension 1 Code	Code	20
22	Global Dimension 2 Code	Code	20

- b. Set the TableRelation property on the **Global Dimension 1 Code** field to the **Code** field of the **Dimension Value** table, where **Global Dimension No.** is CONST 1.

 **Note:** You can use the Table Relation window to set the relation in a more user-friendly way, or you may set the property to the following text: "Dimension Value".Code WHERE (Global Dimension No.=CONST(1))

- c. Set the TableRelation property on the **Global Dimension 2 Code** field to the **Code** field of the **Dimension Value** table, where **Global Dimension No.** is CONST 2.



**Note:** "Dimension Value".Code WHERE (Global Dimension No.=CONST(2))

- d. Set the CaptionClass property on the following fields.

Field	CaptionClass
Global Dimension 1 Code	'1,1,1'
Global Dimension 2 Code	'1,1,2'

- 3. Add code to the OnInsert and OnDelete triggers to call the UpdateDefaultDim and DeleteDefaultDim triggers of the DimensionManagement codeunit.
  - a. At the end of the OnInsert trigger, add the following code.

```
DimMgt.UpdateDefaultDim(
    DATABASE::Seminar,"No.",
    "Global Dimension 1 Code","Global Dimension 2 Code");
```

- b. At the end of the OnDelete trigger, add the following code.

```
DimMgt.DeleteDefaultDim(DATABASE::Seminar,"No.");
```

- c. Compile, save, and then close the table.

### Task 3: Modify the Default Dimension Table

#### High Level Steps

- 1. Add a new function to the table 352, **Default Dimension**, and name it **UpdateSeminarGlobalDimCode**. This function assigns the value to the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields of the specified **Seminar** record.
- 2. Add code to the **UpdateGlobalDimCode** function to call the **UpdateSeminarGlobalDimCode** function when you can.

## Detailed Steps

1. Add a new function to the table 352, **Default Dimension**, and name it **UpdateSeminarGlobalDimCode**. This function assigns the value to the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields of the specified **Seminar** record.
  - a. Design table 352, **Default Dimension**.
  - b. In the **C/AL Globals** window for the table, create a new function, and name it **UpdateSeminarGlobalDimCode**.
  - c. Define the following parameters on the **UpdateSeminarGlobalDimCode** function.

Name	Data Type	Length
GlobalDimCodeNo	Integer	
SeminarNo	Code	20
NewDimValue	Code	20

- d. Define a local record variable for the function. Set the its Subtype to Seminar.
- e. In the function trigger for the function, enter the following code.

```
//CSD1.00>
IF Seminar.GET(SeminarNo) THEN BEGIN
    CASE GlobalDimCodeNo OF
        1:
            Seminar."Global Dimension 1 Code" := NewDimValue;
        2:
            Seminar."Global Dimension 2 Code" := NewDimValue;
    END;
END;
//CSD1.00<
```

2. Add code to the **UpdateGlobalDimCode** function to call the **UpdateSeminarGlobalDimCode** function when you can.
  - a. In the function trigger of the **UpdateGlobalDimCode** function, within the CASE block after the last case, enter the following code.

```
//CSD1.00>  
  
DATABASE::Seminar:  
  
    UpdateSeminarGlobalDimCode(GlobalDimCodeNo,"No.",NewDimValue);  
  
//CSD1.00<
```

- b. Compile, save, and then close the table.

### Task 4: Import the Default Dimensions-Multiple Page

#### *High Level Steps*

1. Import the Mod08\Labfiles\Lab 8.A - Starter - Page 542.fob object file.

#### *Detailed Steps*

1. Import the Mod08\Labfiles\Lab 8.A - Starter - Page 542.fob object file.
  - a. In Object Designer, click **File > Import**.
  - b. Browse to the Mod08\Labfiles\Lab 8.A - Starter - Page 542.fob file, and then click **Open**.
  - c. Click **Yes** to complete the import.
  - d. Click **OK** to complete the object import.
  - e. In Object Designer, click **File > Import**.
  - f. Browse to the Mod08\Labfiles\Lab 8.A - Starter - Page 542.fob file, and then click **Open**.
  - g. Click **Yes** to complete the import.
  - h. Click **OK** to complete the object import.

### Task 5: Modify the Seminar Card and Seminar List Pages

#### *High Level Steps*

1. Add the Dimensions action to the **Navigate** tab on the **Seminar Card** page.
2. Add the Dimensions action group to the **Navigate** tab on the **Seminar List** page.
3. Add the Dimensions-Single and Dimensions-Multiple actions to the Dimensions action group on the **Seminar List** page.

## Detailed Steps

1. Add the Dimensions action to the **Navigate** tab on the **Seminar Card** page.
  - a. Design page 123456700, **Seminar Card**.
  - b. In the Action Designer for the page, under the Seminar action group, add a new action, and set its caption to "Dimensions".
  - c. Set the following properties on the new action.

Property	Value
Image	Dimensions
ShortCutKey	Shift+Ctrl+D
RunObject	Page Default Dimensions
RunPageLink	Table ID=CONST(123456700),No.=FIELD(No.)

- d. Compile, save, and then close the page.

2. Add the Dimensions action group to the **Navigate** tab on the **Seminar List** page.
  - a. Design page 123456701, **Seminar List**.
  - b. In the Action Designer for the page, under the Seminar action group, add a new action group, and set its caption to "Dimensions".
3. Add the Dimensions-Single and Dimensions-Multiple actions to the Dimensions action group on the **Seminar List** page.
  - a. Under the Seminar action group, add a new action, and set its caption to "Dimensions-Single".
  - b. Set the following properties on the Dimensions-Single action.

Property	Value
Image	Dimensions
ShortCutKey	Shift+Ctrl+D
RunObject	Page Default Dimensions
RunPageLink	Table ID=CONST(123456700),No.=FIELD(No.)

- c. Under the Seminar action group, add a new action, and set its caption to "Dimensions-Multiple".



- d. In the OnAction trigger for the Dimensions-Multiple action, define the following local variables.

Name	Data Type	Subtype
Seminar	Record	Seminar
DefaultDimMultiple	Page	Default Dimensions-Multiple

- e. In the OnAction trigger for the Dimensions-Multiple action, enter the following code.

```

CurrPage.SETSELECTIONFILTER(Seminar);

DefaultDimMultiple.SetMultiSeminar(Seminar);

DefaultDimMultiple.RUNMODAL;

```

- f. Set the Image property on the Dimensions-Multiple action to "DimensionSets".
- g. Compile, save, and then close the page.

## Exercise 2: Extending Documents with Dimensions

### Exercise Scenario

Viktor extends the seminar registration tables and pages with required dimension management functionality. He then imports the posted seminar registration tables and pages from an external file that was prepared by Isaac, another developer working on the implementation project for CRONUS International Ltd.

### Task 1: Modify the Seminar Registration Header and Line Tables

#### High Level Steps

1. In the Seminar Registration Line table, add the **Shortcut Dimension 1 Code** and **Shortcut Dimension 2 Code**, and **Dimension Set ID** fields.
2. Add a global variable for the DimensionManagement codeunit.
3. Add the **ShowDimensions** function, and call it from the appropriate trigger.
4. Add the **CreateDim** function and call it from the appropriate trigger.
5. Add the **ValidateShortcutDimCode** function, and call it from the appropriate triggers.
6. Add the **LookupShortcutDimCode** function.
7. Add the **ShowShortcutDimCode** function.

8. In the **Seminar Registration Header** table, add the **Shortcut Dimension 1 Code** and **Shortcut Dimension 2 Code**, and **Dimension Set ID** fields.
9. In the **Seminar Registration Header** table, add a global variable for the DimensionManagement codeunit.
10. In the **Seminar Registration Header** table, add the **SeminarRegLinesExist** function.
11. Add the **CreateDim** function and call it from the appropriate trigger.
12. Add the **ValidateShortcutDimCode** function, and call it from the appropriate triggers.
13. Add the **ShowDocDim** function, and call it from the appropriate trigger.
14. Add the **UpdateAllLineDim** function.
15. Compile the Seminar Registration Line table.

### Detailed Steps

1. In the Seminar Registration Line table, add the **Shortcut Dimension 1 Code** and **Shortcut Dimension 2 Code**, and **Dimension Set ID** fields.
  - a. Design table 123456711, **Seminar Registration Line**.
  - b. Add the following fields to the table.

Field No.	Field Name	Data Type	Length
51	Shortcut Dimension 1 Code	Code	20
52	Shortcut Dimension 2 Code	Code	20
480	Dimension Set ID	Integer	

- c. Set the TableRelation property on the shortcut dimension fields to relate to the **Code** field of the matching record in the **Dimension Value** table.



**Note:** You may copy the values from the same property on the **Global Dimension 1 Code** and **Global Dimension 2 Code** fields in the **Seminar** table.

Otherwise, you may enter "Dimension Value".Code WHERE (Global Dimension No.=CONST(1)). Make sure that you replace "1" with "2" for the **Shortcut Dimension 2 Code** field.

---

- d. Set the TableRelation property on the **Dimension Set ID** field to the **Dimension Set Entry** table.

2. Add a global variable for the DimensionManagement codeunit.
  - a. In the **C/AL Globals** window, define a new variable, and name it DimMgt. Set Type to Codeunit, and subtype to DimensionManagement.
  
3. Add the **ShowDimensions** function, and call it from the appropriate trigger.
  - a. In the C/AL Globals window, create a new function, and name it **ShowDimensions**.
  - b. In the function trigger code for the function, enter the following code.

```
"Dimension Set ID" :=
DimMgt.EditDimensionSet(
    "Dimension Set ID",
    STRSUBSTNO('%1 %2',
    "Document No.",
    "Line No."));
DimMgt.UpdateGlobalDimFromDimSetID(
    "Dimension Set ID",
    "Shortcut Dimension 1 Code",
    "Shortcut Dimension 2 Code");
```

- c. In the OnLookup trigger for the **Dimension Set ID** field, enter the following code.

```
ShowDimensions;
```

4. Add the **CreateDim** function and call it from the appropriate trigger.
  - a. In the **C/AL Globals** window, create a new function, and name it **CreateDim**.
  - b. Define the following parameters for the **CreateDim** function.

Name	DataType	Length
Type1	Integer	
No1	Code	20

- c. Define the following local variables for the **CreateDim** function.

Name	Data Type	Subtype	Length
SourceCodeSetup	Record	Source Code Setup	
TableID	Integer		
No	Code		20

- d. Set the Dimensions property on the *TableID* and *No* local variables to 10.
- e. In the function trigger for the **CreateDim** function, enter the following code.

```
SourceCodeSetup.GET;

TableID[1] := Type1;

No[1] := No1;

"Shortcut Dimension 1 Code" := "";

"Shortcut Dimension 2 Code" := "";

GetSeminarRegHeader;

"Dimension Set ID" :=

DimMgt.GetDefaultDimID(

    TableID,No,SourceCodeSetup.Seminar,

    "Shortcut Dimension 1 Code",

    "Shortcut Dimension 2 Code",

    SeminarRegHeader."Dimension Set ID",

    DATABASE::Seminar);

DimMgt.UpdateGlobalDimFromDimSetID(

    "Dimension Set ID",

    "Shortcut Dimension 1 Code",

    "Shortcut Dimension 2 Code");
```

- f. At the end of the OnValidate trigger for the **Bill-to Customer No.** field, append the following code.

CreateDim(DATABASE::Customer,"Bill-to Customer No.");

5. Add the **ValidateShortcutDimCode** function, and call it from the appropriate triggers.
  - a. In the **C/AL Globals** window, create a new function, and name it **ValidateShortcutDimCode**.
  - b. Define the following parameters for the **ValidateShortcutDimCode** function.

Var	Name	DataType	Length
	FieldNumber	Integer	
Yes	ShortcutDimCode	Code	20

- c. In the function trigger for the **ValidateShortcutDimCode** function, enter the following code.

```
DimMgt.ValidateShortcutDimValues(
    FieldNumber,
    ShortcutDimCode,
    "Dimension Set ID");
```

- d. In the OnValidate trigger for the **Shortcut Dimension 1 Code** field, enter the following code.

ValidateShortcutDimCode(1,"Shortcut Dimension 1 Code");

- e. In the OnValidate trigger for the **Shortcut Dimension 2 Code** field, enter the following code.

ValidateShortcutDimCode(2,"Shortcut Dimension 2 Code");

6. Add the **LookupShortcutDimCode** function.
  - a. In the **C/AL Globals** window, create a new function, and name it **LookupShortcutDimCode**.
  - b. Define the following parameters for the **LookupShortcutDimCode** function.

Var	Name	DataType	Length
	FieldNumber	Integer	
Yes	ShortcutDimCode	Code	20

- c. In the function trigger for the **LookupShortcutDimCode** function, enter the following code.

```
DimMgt.LookupDimValueCode(  
  
    FieldNumber,  
  
    ShortcutDimCode);  
  
ValidateShortcutDimCode(  
  
    FieldNumber,  
  
    ShortcutDimCode);
```

7. Add the **ShowShortcutDimCode** function.
- a. In the **C/AL Globals** window, create a new function, and name it **ShowShortcutDimCode**.
- b. Define the following parameters for the **ShowShortcutDimCode** function.

Var	Name	Data Type	Length
Yes	ShortcutDimCode	Code	20

- c. Set the Dimensions property on the ShortcutDimCode parameter to 8.
- d. In the function trigger for the **ShowShortcutDimCode** function, enter the following code.

```
DimMgt.GetShortcutDimensions(  
  
    "Dimension Set ID",  
  
    ShortcutDimCode);
```

- e. Save the table without compiling, and then close the Table Designer.

8. In the **Seminar Registration Header** table, add the **Shortcut Dimension 1 Code** and **Shortcut Dimension 2 Code**, and **Dimension Set ID** fields.
  - a. Design table 123456710, **Seminar Registration Header**.
  - b. Add the following fields to the table.

Field No.	Field Name	Data Type	Length
51	Shortcut Dimension 1 Code	Code	20
52	Shortcut Dimension 2 Code	Code	20
480	Dimension Set ID	Integer	

- c. Set the TableRelation property on the shortcut dimension fields to relate to the **Code** field of the matching record in the **Dimension Value** table.
  - d. Set the TableRelation property on the **Dimension Set ID** field to the **Dimension Set Entry** table.
9. In the **Seminar Registration Header** table, add a global variable for the DimensionManagement codeunit.
  - a. In the **C/AL Globals** window, define a new variable, and name it DimMgt. Set Type to Codeunit, and subtype to DimensionManagement.
10. In the **Seminar Registration Header** table, add the **SeminarRegLinesExist** function.
  - a. Create a new function, name it **SeminarRegLinesExist**.
  - b. On the **Return Value** tab of the **C/AL Locals** window for the function, set the Return Type to Boolean.
  - c. In the function trigger for the **SeminarRegLinesExist** function, enter the following code.

```
SeminarRegLine.RESET;

SeminarRegLine.SETRANGE("Document No.,"No.");

EXIT(SeminarRegLine.FINDFIRST);
```

11. Add the **CreateDim** function and call it from the appropriate trigger.
  - a. In the **C/AL Globals** window, create a new function, and name it **CreateDim**.
  - b. Define the following parameters for the **CreateDim** function.

Name	Data Type	Length
Type1	Integer	
No1	Code	20
Type2	Integer	
No2	Code	20
Type3	Integer	
No3	Code	20

- c. Define the following local variables for the **CreateDim** function.

Name	Data Type	Subtype	Length
SourceCodeSetup	Record	Source Code Setup	
TableID	Integer		
No	Code		20
OldDimSetID	Integer		

- d. Set the Dimensions property on the *TableID* and *No* local variables to 10.
    - e. In the function trigger for the **CreateDim** function, enter the following code.

```
SourceCodeSetup.GET;

TableID[1] := Type1;

No[1] := No1;

TableID[2] := Type2;

No[2] := No2;

TableID[3] := Type3;

No[3] := No3;

"Shortcut Dimension 1 Code" := "";
```



```
"Shortcut Dimension 2 Code" := "";  
OldDimSetID := "Dimension Set ID";  
"Dimension Set ID" :=  
    DimMgt.GetDefaultDimID(TableID,No,  
    SourceCodeSetup.Seminar,  
    "Shortcut Dimension 1 Code",  
    "Shortcut Dimension 2 Code",0,0);  
IF (OldDimSetID <> "Dimension Set ID") AND  
    SeminarRegLinesExist  
THEN BEGIN  
    MODIFY;  
    UpdateAllLineDim("Dimension Set ID",OldDimSetID);  
END;
```

- f. At the end of the OnValidate trigger for the **Seminar No.** field, append the following code.

```
CreateDim(  
    DATABASE::Seminar,"Seminar No.",  
    DATABASE::Resource,"Instructor Resource No.",  
    DATABASE::Resource,"Room Resource No.");
```

- g. At the end of the OnValidate trigger for the **Instructor Resource No.** field, append the following code.

```
CreateDim(  
    DATABASE::Seminar,"Seminar No.",  
    DATABASE::Resource,"Instructor Resource No.",  
    DATABASE::Resource,"Room Resource No.");
```

- h. At the end of the OnValidate trigger for the **Room Resource No.** field, append the following code.

```

CreateDim(
    DATABASE::Seminar,"Seminar No.",
    DATABASE::Resource,"Instructor Resource No.",
    DATABASE::Resource,"Room Resource No.");
    
```

12. Add the **ValidateShortcutDimCode** function, and call it from the appropriate triggers.

- a. In the **C/AL Globals** window, create a new function, and name it **ValidateShortcutDimCode**.
- b. Define the following parameters for the **ValidateShortcutDimCode** function.

Var	Name	Data Type	Length
	FieldNumber	Integer	
Yes	ShortcutDimCode	Code	20

- c. Define a local integer variable for the **ValidateShortcutDimCode** function, and name it **OldDimSetID**.
- d. In the function trigger for the **ValidateShortcutDimCode** function, enter the following code.

```

OldDimSetID := "Dimension Set ID";
DimMgt.ValidateShortcutDimValues(
    FieldNumber,
    ShortcutDimCode,
    "Dimension Set ID");
IF "No." <> " THEN
    MODIFY;
IF OldDimSetID <> "Dimension Set ID" THEN BEGIN
    MODIFY;
IF SeminarRegLinesExist THEN
    
```

```
UpdateAllLineDim(
    "Dimension Set ID",
    OldDimSetID);
END;
```

- e. In the OnValidate trigger for the **Shortcut Dimension 1 Code** field, enter the following code.

```
ValidateShortcutDimCode(1,"Shortcut Dimension 1 Code");
```

- f. In the OnValidate trigger for the **Shortcut Dimension 2 Code** field, enter the following code.

```
ValidateShortcutDimCode(2,"Shortcut Dimension 2 Code");
```

13. Add the **ShowDocDim** function, and call it from the appropriate trigger.

- In the **C/AL Globals** window, create a new function, and name it **ShowDocDim**.
- Define a local integer variable for the **ShowDocDim** function, and name it **OldDimSetID**.
- In the function trigger of the **ShowDocDim** function, enter the following code.

```
OldDimSetID := "Dimension Set ID";

"Dimension Set ID" :=
DimMgt.EditDimensionSet2(
    "Dimension Set ID","No.",
    "Shortcut Dimension 1 Code",
    "Shortcut Dimension 2 Code");

IF OldDimSetID <> "Dimension Set ID" THEN BEGIN
    MODIFY;

    IF SeminarRegLinesExist THEN

        UpdateAllLineDim(
            "Dimension Set ID",
```

```
OldDimSetID);  
  
END;
```

- d. In the OnLookup trigger for the **Dimension Set ID** field, enter the following code.

```
ShowDocDim;
```

14. Add the **UpdateAllLineDim** function.

- a. In the **C/AL Globals** window, create a new function, and name it **UpdateAllLineDim**.
- b. Define the following parameters for the **UpdateAllLineDim** function.

Name	Data Type
NewParentDimSetID	Integer
OldParentDimSetID	Integer

- c. Define a local integer variable for the **UpdateAllLineDim** function, and name it NewDimSetID.
- d. Define a global text constant, name it Text009, and enter the following ConstValue for it: "You may have changed a dimension.\\Do you want to update the lines?"
- e. In the function trigger for the function, enter the following code.

```
IF NewParentDimSetID = OldParentDimSetID THEN  
  
    EXIT;  
  
IF NOT CONFIRM(Text009) THEN  
  
    EXIT;  
  
SeminarRegLine.RESET;  
  
SeminarRegLine.SETRANGE("Document No.,"No.");  
  
SeminarRegLine.LOCKTABLE;  
  
IF SeminarRegLine.FIND('-') THEN  
  
    REPEAT  
  
        NewDimSetID := DimMgt.GetDeltaDimSetID(  

```

```
SeminarRegLine."Dimension Set ID",
NewParentDimSetID,
OldParentDimSetID);
IF SeminarRegLine."Dimension Set ID" <> NewDimSetID
THEN BEGIN
    SeminarRegLine."Dimension Set ID" := NewDimSetID;
    DimMgt.UpdateGlobalDimFromDimSetID(
        SeminarRegLine."Dimension Set ID",
        SeminarRegLine."Shortcut Dimension 1 Code",
        SeminarRegLine."Shortcut Dimension 2 Code");
    SeminarRegLine.MODIFY;
END;
UNTIL SeminarRegLine.NEXT = 0;
```

f. Compile, save, and then close the table.

15. Compile the Seminar Registration Line table.

- a. In Object Designer, select the table **123456711, Seminar Registration Line**.
- b. Click **Tools > Compile** (or press F11).

### Task 2: Modify the Seminar Registration Pages

#### *High Level Steps*

1. Add the Dimensions action to the **Seminar Registration** and **Seminar Registration List** pages.
2. Add field controls for eight shortcut dimensions to the **Seminar Registration Subform** page.
3. Add the Dimension actions to the Line action group.
4. Modify the OnAfterGetRecord and OnNewRecord page triggers.
5. Modify the OnValidate trigger for the **Bill-to Customer No.** field.

## Detailed Steps

1. Add the Dimensions action to the **Seminar Registration** and **Seminar Registration List** pages.
  - a. Design page 123456710, **Seminar Registration**.
  - b. In the Seminar Registration action group, add a new page action, set its caption to "Dimensions", and set the following properties.

Property	Value
Image	Dimensions
ShortCutKey	Shift+Ctrl+D

- c. In the OnAction trigger for the Dimensions action, enter the following code.

```
ShowDocDim;  
  
CurrPage.SAVERECORD;
```

- d. Compile, save, and then close the **Seminar Registration** page.
      - e. Design page 123456713, **Seminar Registration List**.
      - f. In the Seminar Registration action group, add a new page action, set its caption to "Dimensions", and set the following properties.

Property	Value
Image	Dimensions
ShortCutKey	Shift+Ctrl+D

- g. In the OnAction trigger for the Dimensions action, enter the following code.

```
ShowDocDim;
```

- h. Compile, save, and then close the **Seminar Registration List** page.
2. Add field controls for eight shortcut dimensions to the **Seminar Registration Subform** page.
    - a. Design page 123456711, **Seminar Registration Subform**.
    - b. Create the ShortcutDimCode global variable of type Code and length 20, and set its Dimensions property to 8.

- c. Design page 46, **Sales Order Subform**.
- d. Copy to the clipboard all field controls with the following SourceExpr values: "Shortcut Dimension 1 Code", "Shortcut Dimension 2 Code", ShortcutDimCode[3], ShortcutDimCode[4], ShortcutDimCode[5], ShortcutDimCode[6], ShortcutDimCode[7], ShortcutDimCode[8].



**Note:** Select the specified controls, then press CTRL+C.

- e. Close the **Sales Order Subform** page.
  - f. In the page designer for the **Seminar Registration Subform** page, at the first empty line, paste the field controls from the clipboard (press CTRL+V).
3. Add the Dimension actions to the Line action group.
- a. In the Action Designer of the **Seminar Registration Subform** page, define the following actions.

Type	SubType	Caption
ActionContainer	ActionItems	<Action9>
ActionGroup		&Line
Action		Dimensions

- b. For the Dimensions action, define the following properties.

Property	Value
Image	Dimensions
ShortCutKey	Shift+Ctrl+D

- c. In the OnAction trigger for the Dimensions action, enter the following code.

```
ShowDimensions;
```

- 4. Modify the OnAfterGetRecord and OnNewRecord page triggers.
  - a. In the OnAfterGetRecord trigger, enter the following code.

```
ShowShortcutDimCode(ShortcutDimCode);
```

- b. In the OnNewRecord trigger, enter the following code.

```
CLEAR(ShortcutDimCode);
```

5. Modify the OnValidate trigger for the **Bill-to Customer No.** field.
  - a. In the OnValidate trigger for the **Bill-to Customer No.** field, enter the following code.

```
ShowShortcutDimCode(ShortcutDimCode);
```

- b. Compile, save, and then close the **Seminar Registration Subform** page.

### Task 3: Import Posted Seminar Registration Tables and Pages

#### High Level Steps

1. Import the Mod08\Labfiles\Lab 8.A - Starter.fob object file.

#### Detailed Steps

1. Import the Mod08\Labfiles\Lab 8.A - Starter.fob object file.
  - a. In Object Designer, click **File > Import**.
  - b. Browse to the Mod08\Labfiles\Lab 8.A - Starter.fob file, and then click **Open**.
  - c. Click **OK** to open the Import Worksheet.
  - d. Click **Replace All**, and then click **OK**.
  - e. Click **OK** to complete the object import.

### Exercise 3: Extending Ledger Entries and Posting Process with Dimensions

#### Exercise Scenario

Viktor extends the journal posting routines and imports the changes that were made by Isaac to ledger tables and pages and the document posting routine.

### Task 1: Modify the Seminar Jnl.-Check Line Codeunit

#### High Level Steps

1. In the Seminar Jnl.-Check Line codeunit, enter code to call the **CheckDimIDComb** and **CheckDimValuePosting** functions of the DimensionManagement codeunit.

#### Detailed Steps

1. In the Seminar Jnl.-Check Line codeunit, enter code to call the **CheckDimIDComb** and **CheckDimValuePosting** functions of the DimensionManagement codeunit.
  - a. Design codeunit 123456731, Seminar Jnl.-Check Line.
  - b. Create a global variable named DimMgt for the DimensionManagement codeunit.
  - c. Create the following global text constants.



Name	ConstValue
Text002	The combination of dimensions used in %1 %2, %3, %4 is blocked. %5
Text003	A dimension used in %1 %2, %3, %4 has caused an error. %5

d. In the **RunCheck** function, create the following local variables.

Name	DataType	Length
TableID	Integer	
No	Code	20

- e. On both variables, set the Dimensions property to 10.
- f. In the end of the **RunCheck** function, before the final END of the WITH block, enter the following code.

```

IF NOT DimMgt.CheckDimIDComb("Dimension Set ID") THEN

ERROR(

    Text002,

    TABLECAPTION,"Journal Template Name",

    "Journal Batch Name","Line No.",

    DimMgt.GetDimCombErr);

TableID[1] := DATABASE::Seminar;

No[1] := "Seminar No.";

TableID[2] := DATABASE::Resource;

No[2] := "Instructor Resource No.";

TableID[3] := DATABASE::Resource;

No[3] := "Room Resource No.";

IF NOT DimMgt.CheckDimValuePosting(TableID,No,"Dimension Set ID") THEN

    IF "Line No." <> 0 THEN

        ERROR(
    
```

```
Text003,  
  
TABLECAPTION,"Journal Template Name",  
  
"Journal Batch Name","Line No.",  
  
DimMgt.GetDimValuePostingErr)  
  
ELSE  
  
ERROR(DimMgt.GetDimValuePostingErr);
```

- g. Compile, save, and then close the codeunit.

### Task 2: Modify the Seminar Jnl.-Post Line Codeunit

#### High Level Steps

1. In the Seminar Jnl.-Post Line codeunit, add code to copy **Shortcut Dimension 1 Code**, **Shortcut Dimension 2 Code**, and **Dimension Set ID** fields from the journal line to the **Global Dimension 1 Code**, **Global Dimension 2 Code**, and **Dimension Set ID** fields in the ledger entry.

#### Detailed Steps

1. In the Seminar Jnl.-Post Line codeunit, add code to copy **Shortcut Dimension 1 Code**, **Shortcut Dimension 2 Code**, and **Dimension Set ID** fields from the journal line to the **Global Dimension 1 Code**, **Global Dimension 2 Code**, and **Dimension Set ID** fields in the ledger entry.
  - a. Design codeunit 123456732, Seminar Jnl.-Post Line.
  - b. In the Code function trigger, after assignment block of **SeminarLedgerEntry** table, and before the `SeminarLedgerEntry.INSERT` statement, enter the following code.

```
SeminarLedgerEntry."Global Dimension 1 Code" := "Shortcut Dimension 1 Code";  
  
SeminarLedgerEntry."Global Dimension 2 Code" := "Shortcut Dimension 2 Code";  
  
SeminarLedgerEntry."Dimension Set ID" := "Dimension Set ID";
```

- c. Compile, save, and then close the codeunit.

### Task 3: Import and Review the Seminar-Post Codeunit

#### **High Level Steps**

1. Import the Mod08\Labfiles\Lab 8.A - Starter - Seminar-Post.fob object file.
2. Review the code in the Seminar-Post codeunit.

#### **Detailed Steps**

1. Import the Mod08\Labfiles\Lab 8.A - Starter - Seminar-Post.fob object file.
  - a. In Object Designer, click **File > Import**.
  - b. Browse to the Mod08\Labfiles\Lab 8.A - Starter - Seminar-Post.fob file and then click **Open**.
  - c. Click **OK** to open the **Import Worksheet**.
  - d. Click **Replace All**, and then click **OK**.
  - e. Click **OK** to complete the object import.
2. Review the code in the Seminar-Post codeunit.
  - a. Design codeunit 123456700, Seminar-Post.
  - b. Review code in the **CheckDim** function, and see where it is called from.
  - c. Review code in the **CheckDimComb** and **CheckDimValuePosting** functions, and see where they are called from.
  - d. Review code in the **PostResJnlLine** and **PostSeminarJnlLine** functions, and how these functions assign dimension fields from the document lines to resource and seminar journals.
  - e. Close the Seminar-Post codeunit.

## Module Review

### *Module Review and Takeaways*

Dimensions let users extend the data models of master tables, documents, journals, and ledgers, with additional attributes that meet a company's business needs. They also let users establish rules that posting routines check before creating any posting entries. These posting routines make sure that any data that is entered by users complies with the business rules.

This chapter explored the role of dimensions in Microsoft Dynamics NAV 2013 and demonstrated how to integrate the dimensions management functionality in the existing features of the Seminar Management module. Therefore, CRONUS International Ltd. users can now work with dimensions in the Seminar Management module, in the same manner as they use dimensions in any other application area of Microsoft Dynamics NAV 2013.

### Test Your Knowledge

Test your knowledge with the following questions.

1. What is a dimension and what is its purpose in Microsoft Dynamics NAV 2013?

---

---

---

---

2. Which of the following types is not a dimension type in Microsoft Dynamics NAV 2013?

- ( ) Global Dimension
- ( ) Shortcut Dimension
- ( ) Budget Dimension
- ( ) Department Dimension

3. What table stores information about dimensions users define on master records, such as Customer, Vendor, or Item?

---

---

---

---

4. What table stores dimension combinations for documents, journals, or ledger entries?

---

---

---

---

5. How do you copy dimension information from a document line to the journal line during the posting process?

---

---

---

---

6. Which object in Microsoft Dynamics NAV 2013 contains most functionality related to managing dimensions?

---

---

---

---

## Test Your Knowledge Solutions

### Module Review and Takeaways

1. What is a dimension and what is its purpose in Microsoft Dynamics NAV 2013?

MODEL ANSWER:

A dimension is a value that is entered into a transaction to enable transactions with similar values to be grouped together for analysis.

2. Which of the following types is not a dimension type in Microsoft Dynamics NAV 2013?

- ( ) Global Dimension
- ( ) Shortcut Dimension
- ( ) Budget Dimension
- (v) Department Dimension

3. What table stores information about dimensions users define on master records, such as Customer, Vendor, or Item?

MODEL ANSWER:

Default Dimension

4. What table stores dimension combinations for documents, journals, or ledger entries?

MODEL ANSWER:

Dimension Set Entry

5. How do you copy dimension information from a document line to the journal line during the posting process?

MODEL ANSWER:

You assign the same value of the Dimension Set ID field of the document line to the Dimension Set ID field of the journal line.

6. Which object in Microsoft Dynamics NAV 2013 contains most functionality related to managing dimensions?

MODEL ANSWER:

Codeunit 408, DimensionManagement

