Presentation Design Guide

edocs®

ea**Suite Tools**

Table of Contents

About Customer Self-Service and eaSuite™

edocs has developed the industry's most comprehensive software and services for deploying Customer Self-Service solutions. **eaSuite**™ combines electronic presentment and payment (EPP), order management, knowledge management, personalization and application integration technologies to create an integrated, natural starting point for all customer service issues. eaSuite's unique architecture leverages and preserves existing infrastructure and data, and offers unparalleled scalability for the most demanding applications. With deployments across the healthcare, financial services, energy, retail, and communications industries, and the public sector, eaSuite powers some of the world's largest and most demanding customer self-service applications. eaSuite is a standards-based, feature rich, and highly scalable platform, that delivers the lowest total cost of ownership of any self-service solution available.

eaSuite is comprised of four product families:

- Electronic Presentment and Payment (EPP) Applications

- Advanced Interactivity Applications

- Enterprise Productivity Applications

- Development Tools
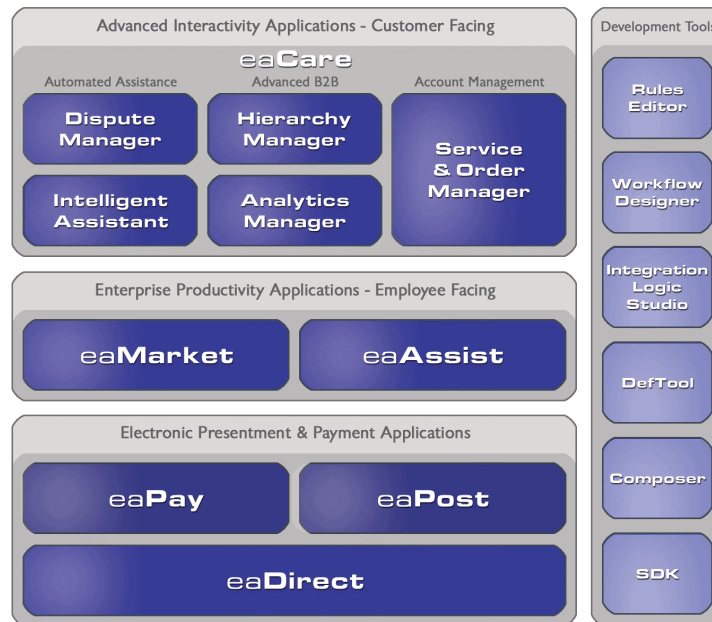
Electronic Presentment and Payment (EPP) Applications are the foundation of edocs' Customer Self-Service solution. They provide the core integration infrastructure between organizations' backend transactional systems and end users, as well as rich e-billing, e-invoicing and e-statement functionality. Designed to meet the rigorous demands of the most technologically advanced organizations, these applications power Customer Self-Service by managing transactional data and by enabling payments and account distribution.

eaDirect™ is the core infrastructure of enterprise Customer Self-Service solutions for organizations large and small with special emphasis on meeting the needs of organizations with large numbers of customers, high data volumes and extensive integration with systems and business processes across the enterprise. Organizations use eaDirect with its data access layer, composition engine, and security, enrollment and logging framework to power complex Customer Self-Service applications.

eaPay™ is the electronic payment solution that decreases payment processing costs, accelerates receivables and improves operational efficiency. eaPay is a complete payment scheduling and warehousing system with real-time and batch connections to payment gateways for Automated Clearing House (ACH) and credit card payments, and payments via various payment processing service providers.

eaPost® is the account content distribution system that handles all the complexities of enrollment, authentication and secure distribution of summary account information to any endpoint, while also bringing customers back the organization's Website to manage and control their self-service experience.

Advanced Interactivity Applications are a comprehensive set of advanced customer-facing self-service capabilities that enable the full range of business and consumer customer service activities. These sophisticated modules have the flexibility to completely customize the Customer Self-Service solution to meet vertical industry and specific company requirements.

eaCare™ consists of a rich set of sophisticated self-service modules – Dispute Manager, Intelligent Assistant, Hierarchy Manager, Analytics Manager, and Service and Order Manager - for automated assistance, advanced business-to-business applications and account management. These capabilities come together to create a web self-service dashboard for customers to access all service offerings from a single, easy-to-use interface. eaCare's modularity accelerates time to market with components that can be deployed incrementally in a phased approach.

Enterprise Productivity Applications are employee-facing solutions that empower customer service representatives, sales agents, account managers, marketing managers, broker-dealers and channel partners within an organization and external partner organizations to facilitate self-service and to support assisted service. Employees leverage edocs' Customer Self-Service solution to deliver customer service, access information, create and deploy marketing and customer service content, and perform activities for the benefit of customers.

eaAssist™ reduces interaction costs and increases customer satisfaction by enabling enterprise agents – customer service representatives (CSRs), sales agents, broker-dealers and others – to efficiently access critical account data and service-related information to effectively service customers. Through its browser interface designed especially for the enterprise agent, eaAssist enables agents to take advantage of customer-facing online capabilities to provide better service by more efficiently resolving customer account inquiries at the point of customer contact.

eaMarket™ is the personalization, campaign and content management solution that enables organizations to increase revenue and improve customer satisfaction by weaving personalized marketing and customer service messages throughout the Customer Self-Service experience. The transactional account data that provides the foundation for a Customer Self-Service solution – such as transaction activity, service or usage charges, current task and prior service history – bring valuable insight into customers and can help optimize personalized marketing and customer service campaigns. eaMarket leverages that data to present relevant marketing and customer service messages to customers.

edocs' **Development Tools** are visual development environments for designing and configuring edocs' Customer Self-Service solutions. The Configuration Tools encompass data and rules management, workflow authoring, systems integration, and a software development kit that makes it easy to create customer and employee-facing self-service applications leveraging eaSuite.

About This Guide

This guide describes the tasks required to design and create an eaDirect application for presenting statements online. It provides instructions on how to use the Composer design and development tool.

This guide is intended for the application developer and those involved in the process of designing an eaDirect application.

This guide assumes you have:

- Completed a Statement Mastering Plan.

- Installed the eaDirect tools, DefTool and Composer, on a Windows NT or 2000 system.

Related Documentation

Online Help and a PDF version of this guide are also available.

Online	How to Access
Help	In DefTool or Composer, select **Help>Help Topics**.
A PDF of this guide	A PDF of this guide is available on the eaDirect product CD-ROM.

This guide is part of the eaDirect documentation set. For more information about implementing your eaDirect application, see one of the following guides:

Print Document	Description
eaDirect Installation and Configuration Guide	How to install eaDirect and configure it in a distributed environment.
eaDirect Data Definition Guide	How to use DefTool to define the rules for data extraction in a DDF file.
eaDirect Data Presentation Production Guide	How to set up and run a live eaDirect application in a J2EE environment.

If You Need Help

Technical support is available to customers who have valid maintenance and support contracts with edocs. Technical support engineers can help you install, configure, and maintain your edocs application.

edocs provides global Technical Support services from the following Support Centers:

US Support Center
Natick, MA
Mon-Fri 8:30am – 8:00pm US EST
Telephone: 508-652-8400

Europe Support Center
London, United Kingdom
Mon-Fri 9:00am – 5:00 GMT
Telephone: +44 20 8956 2673

Asia Pac Rim Support Center
Melbourne, Australia
Mon-Fri 9:00am – 5:00pm AU
Telephone: +61 3 9909 7301

Customer Central
https://support.edocs.com

Email Support
mailto:support@edocs.com

When you report a problem, please be prepared to provide us the following information:

- What is your name and role in your organization?

- What is your company's name?

- What is your phone number and best times to call you?

- What is your e-mail address?

- In which edocs product did a problem occur?

- What is your Operating System version?

- What were you doing when the problem occurred?

- How did the system respond to the error?

- If the system generated a screen message, please send us that screen message.

- If the system wrote information to a log file, please send us that log file.

If the system crashed or hung, please tell us.

Introduction to Composer

What is Composer?

Composer is the GUI design tool you use to map the data extracted from your data input file to an HTML template for online presentation.

The result of using Composer is an Application Logic File (ALF) for your application and an edited version of one or more HTML templates.

When a customer makes a request to view a statement in a live eaDirect application, the eaDirect server uses the information in the ALF to locate the associated DDF, HTML template, digital images, and the data input file for extracting and present the statement.

Using the National Wireless Sample Application

A sample eaDirect application named "NatlWireless" is included with your system to let you demonstrate and practice using the eaDirect design tools (DefTool and Composer) with sample files and data. NatlWireless simulates telecommunication statements in an ASCII file format.

With some procedures in this guide, specific instructions for using NatlWireless are included so you can follow along (if you choose). The National Wireless input instructions appear in parenthesis after each step.

The sample application files can be found in EDCSbd\samples\NatlWireless.

The NatlWireless application consists of the following files:

NatlWireless File	Purpose
NatlWireless.txt	Sample data input file (source).
NatlWireless.htm	HTML template for presenting the summary statement view (named HtmlDetail).
NW_LocSummary.htm	HTML template for presenting the detail statement view (named NW_LocSummary).
NWEmailDefault.htm	Default HTML template for notification emails.
NWEmail.htm	Alternate layout for notification emails (its use is determined by business logic).
IndexerJob\ NatlWirelessIndexer.DDF	Defines fields for indexing the NatlWireless data.
NatlWireless.DDF	Contains the customized rules and business logic for extracting each field from each page of the original statement file for presenting different views of the data in online statements.
NatlWireless.ALF	Contains the customized rules and business logic for mapping and presenting the extracted data in an online summary statement.
NW_EmailDefault.ALF	Contains the customized rules and business logic for creating notification email.

After Creating an eaDirect Application

When you are finished using DefTool and Composer to create your application, refer to the *eaDirect Data PresentationProduction Guide* for instructions on how to use the eaDirect Command Center to implement and run an application in a live production environment.

Overview of Creating an ALF

Before Creating an Application Logic File (ALF)

You must create the following files for your application before you can begin using Composer to create the rules for statement presentment:

- The Data Definition File (DDF) using DefTool

- The HTML templates you plan to use for presentment

You must create a separate ALF for each application view. Become familiar with the different views your application requires before starting to create ALFs.

Steps to Create an ALF

To create an ALF:

1. **Run Composer**. From the Windows Start menu, select **Programs>eaDirect>Composer**.

2. **Create a new ALF**. Associate the new ALF with the DDF.

3. **Define the default template** for the view (ALF).

4. **Map field, table, and group definitions to your template.** Composer displays all the data definitions from your DDF in a tree and lets you click and drag the definitions onto your HTML template. Be sure to work in the WYSIWYG template view.

5. **Format mapped fields and tables.** Composer lets you specify font and cell properties.

6. **Customize the ALF** to present the email notification or statement data as needed:

   - Add content. Content can include images, text, hyperlinks, HTML code, HTML statement templates, and certain table formatting.

   - Create any simple or complex data-driven conditions (business logic, called logic elements) on sections or tables in the statement. A condition contains the data-based business logic that determines which image, text, HTML code, HTML template, or conditional table formatting to display.

   - Specify the HTML template to use for each occurrence in a statement.

   - Create any sort, paging, filter, combine, or arithmetic elements and add corresponding element hyperlinks.

   - Add any conditional templates or switch ALFs.

7. **Use the Simulator to view the composed sample statements.** Simulator lets you see how the composed view will appear in a browser. This lets you make necessary changes without running the production jobs to test the application.

8. **Save your ALF and the HTML templates.**

9. **Create any additional ALF your application requires (if any)**

A completed ALF contains all the information about how to assemble an email or statement view. Composer adds eaDirect-specific tags into the HTML templates during the ALF creation process, indicating where data should be modified and where logic arguments need to be applied. The actual logic arguments, as well as other assembly instructions, are stored in the ALF.

Once you have finished creating your application ALFs, you can proceed to test and implement your application in a live production environment. See the *eaDirect Production Guide*.

Saving the ALF, HTML Template Files

Whenever you make changes to the composition rules in an ALF, you must save the ALF. Whenever you save the ALF, Composer automatically saves the open template file as well. You can save the template independently, however.

See "Completed ALF and HTML Application Files" on Page 19 for details about the features saved in each file.

To save the ALF (and HTML template):

- Select `File>Save Alf` or click 🖫 on the ALF toolbar.

Tip	Since the ALF is an XML file, you can edit it in a text or XML editor. However, make sure syntax is correct, or you won't be able to open the an ALF in Composer and problems can arise during production.

To save the HTML template only:

- Click 🖫 on the Template toolbar. (Be careful not to click the ALF save icon, shown above; the Save Template icon has the letter "T" on it.)

Tip	Giving each template revision a "template ID" comment in the HTML template that indicates (matches) the dated directory in which the template resides can be helpful when troubleshooting.

Completed ALF and HTML Application Files

Be sure to save your ALF whenever you make additions or changes to composition rules. Whenever you save an ALF, Composer automatically saves the open template as well. (You can save the template independently.)

A complete ALF contains the following information:

- File path/name of the associated DDF
- Version of eaDirect

- File path/name of the associated default HTML template

In addition, a complete ALF includes the following information for any features you add, including:

- Content name for:
  - Switch ALFs
  - Alternate HTML statement templates (conditional templates)
  - HTML group templates
  - Business logic-driven content containing images, text, or HTML files
- Conditional statements containing business logic
- Logic elements (pieces of comparative logic you use in conditions)
- Group assembly statements
- Table formatting information
- Sorting configuration details (sort elements)
- Paging configuration details (page elements)
- Filter configuration details, including top/bottom filters (filter elements)
- Arithmetic elements to sum a column or count rows in a table
- Configuration details to combine tables for paging or sorting (combine elements)

The completed HTML template can contain the following eaDirect application features:

- Tags for mapped fields, tables, groups, and charts
- Formatting (except table formatting, which is saved in the ALF)
- Open and close tags for sections of HTML on which you create one or more conditions
- Links to sort, page, and filter elements
- Links to subsequent views, such as a detail view.

See also "Saving the ALF, HTML Template Files" on Page 19.

The Composer Interface

The Composer interface consists of:

- Menus

- Toolbars

- Application and Definition Trees on the left

- The open HTML template on the right

The Trees, menus, and toolbars often provide more than one way to select a Composer feature or perform a task.
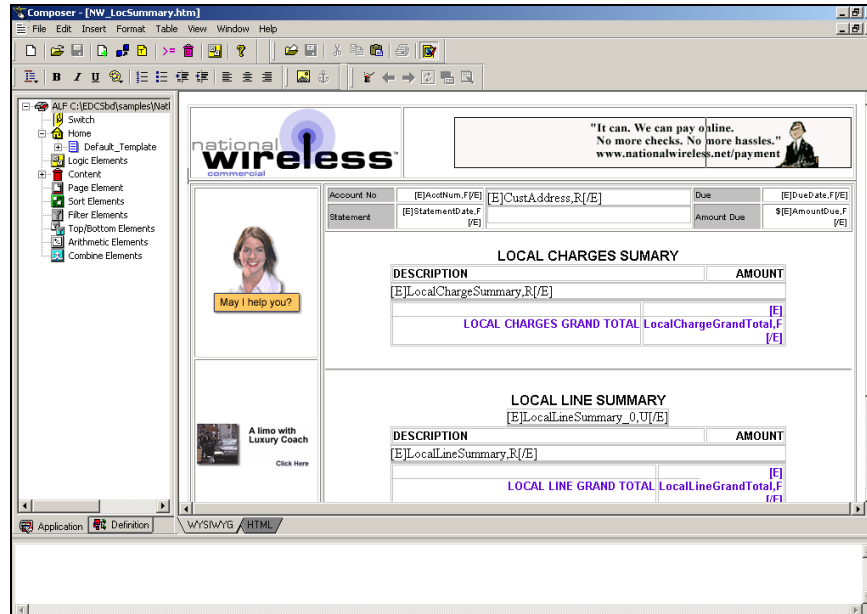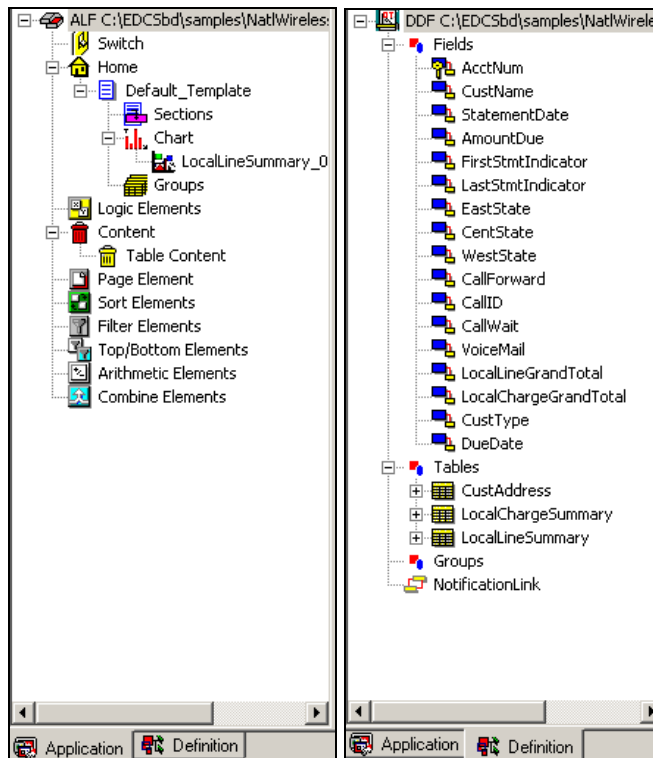

Application and Definition Trees

The left side of the Composer window lets you alternately display either of two trees that let you add data to your statement or create additional features for presentment:

- **Application Tree** – The Application tab displays a list of composition rules and features that have been created in the Application Logic File (ALF), such as any switch ALFs or alternate HTML templates added, conditions on sections, content, logic elements, and sorting, paging, filtering, arithmetic, or combine elements. You can right-click on categories and items in the Application tree for a shortcut to creating new features (such as new content) or other design tasks. The Application tree is empty for a new ALF.

- **Definition Tree** – The Definition tab displays an expandable list of all data fields, tables and groups defined in the associated Data Definition File (DDF). You can click and drag items from the Definitions tab to the HTML Template area to map these items to the statement for presentation.

The Application Tree tab appears on the left, the Definition Tree tab on the right:

To switch between trees:

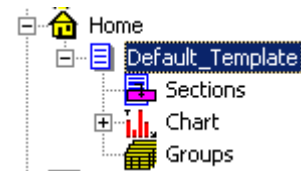- Click the [🔲 Definition] or [🔲 Application] tab.

To expand or collapse a tree node:

- Double-click the node name or click the plus (+) or minus (-) sign in front of the node to expand or collapse the list of items.
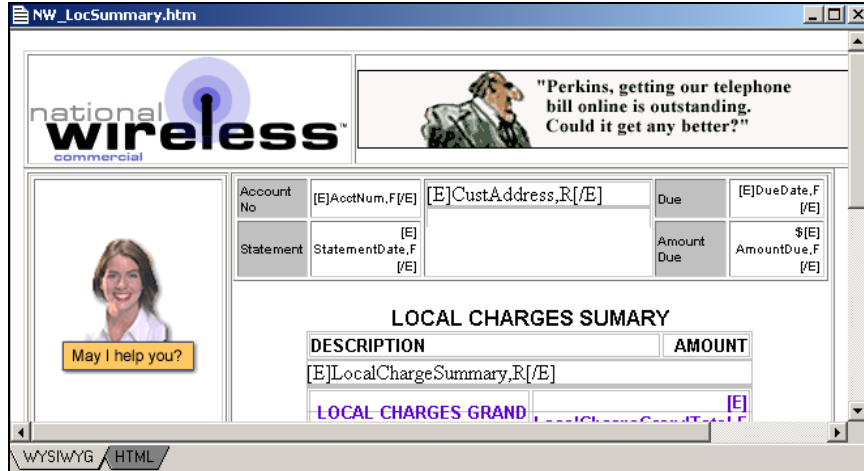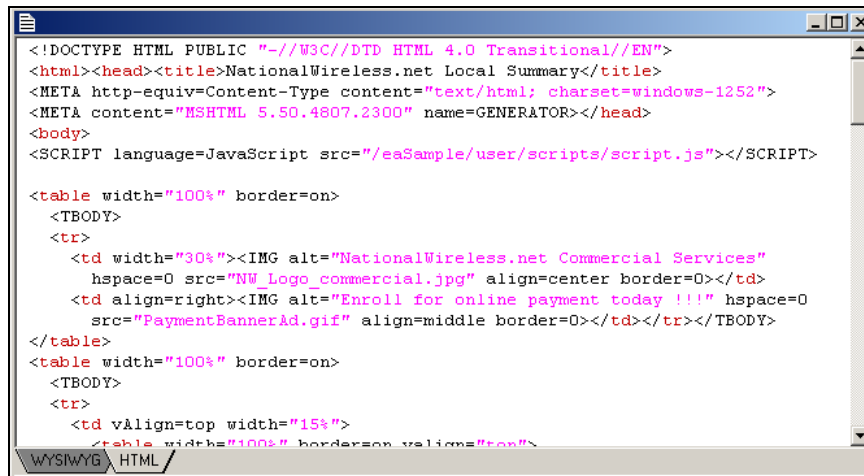
Collapsed:

Expanded:

HTML Template Area

The open HTML template appears in the right pane of the Composer window. Use this work area to view and edit your statement graphically in WYSIWYG (what-you-see-is-what-you-get) mode:

You can switch to HTML mode to view and edit the source HTML:

To switch HTML template edit mode:

- Click the HTML or WYSIWYG tab.

Menus

File Menu

The Composer File menu let you open, close, or save your current ALF and associated HTML template files, or access recently opened ALFs directly.

Edit Menu

Use the Edit menu for standard edit and text search functionality in HTML or WYSIWYG mode, including Undo and Redo actions on the template.

Insert Menu

The Insert menu lets you insert images, anchors (bookmarks), or HTML text in the template, or create content or logic elements. You can also insert the character set tag into the HTML template for internal reference (see "Using Encoded Character Sets in the HTML Templates" on Page 76 for details).

Format Menu

Use the Format menu for standard formatting options such as style, font, alignment, indent and bulleted and numbered lists. You can use it to directly edit the HTML template.

Table Menu

Provides standard table options to insert and edit HTML tables in your template (not the data tables you map from your DDF).

View Menu

Use the View menu to define which Composer panes and toolbars you want to view (toggle on/off), view a list of conditions, logic elements, or content, go into edit mode, or select the Simulation options.

The ALF Objects Bar option displays/hides the Composer Application and Definition Trees. (You can also right-click on the white space in the trees and click the Hide button that appears.)

Window Menu

The Window options provide standard tile and cascade capabilities.

Help Menu

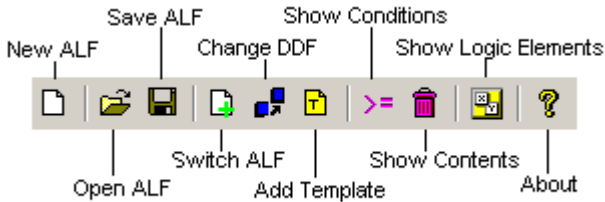The Help menu provides access to online help topics for Composer and the product About box.

Toolbars

The Composer includes several toolbars, shown below, containing icons that let you quickly access many key functions. To find out what an icon represents, hold the mouse over the icon and the Quick Help description appears.
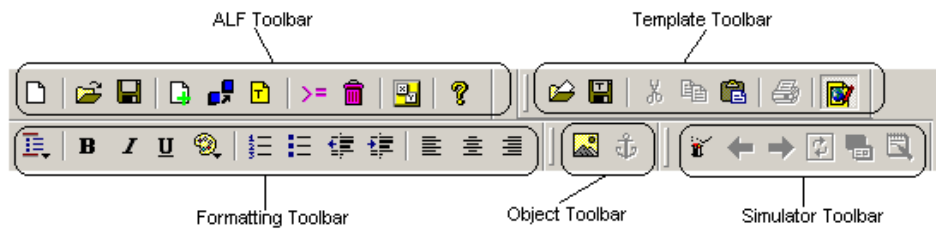

ALF Toolbar

The ALF toolbar icons let you to perform functions associated with the ALF. You can create a new ALF, opening an existing ALF, save the file, create a switch ALF condition, change the associated DDF, view a list of conditions, contents, or logic elements, add a conditional template, or view the Composer About box.

New ALF   Save ALF   Change DDF   Show Conditions   Show Logic Elements

Open ALF   Switch ALF   Add Template   Show Contents   About
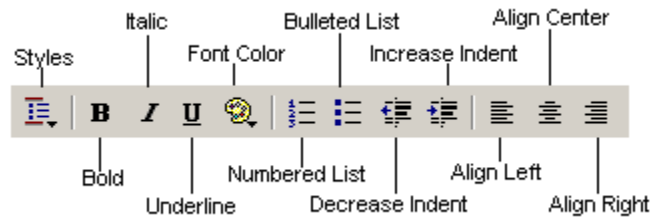
Template Toolbar

The Template toolbar icons affect the open template file. You can open a template, save changes, cut/copy/paste items to it, print, or toggle in and out of edit mode.

Save Template   Copy   Print

Open HTML Template   Cut   Paste   Toggle Edit Mode
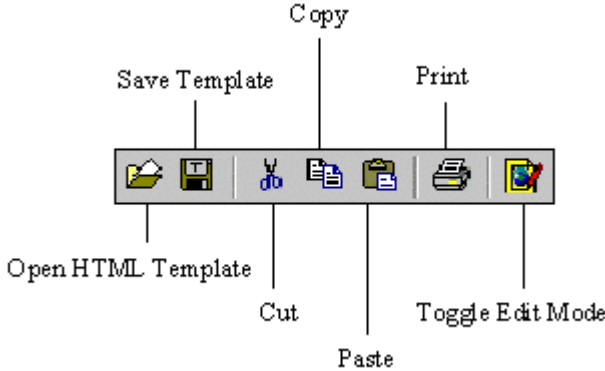
Formatting Toolbar

Use the Formatting toolbar icons to format the template. You can apply various character and paragraph formatting, including bold, italics, underline, color, numbering, bullets, justification, and indents. Use the Style Format icon to apply a variety of formatting.

Object Toolbar

The Object toolbar icons let you insert graphics or hyperlinks into the template (first select an object on the template to associate a hyperlink with it).
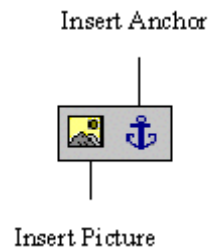
Simulator Toolbar

The Simulator toolbar icons let you use a test data file to view the statements as they will appear composed in a browser. You can page backward and forward through multiple file documents, verifying that the conditional logic is defined correctly and that all values map correctly.
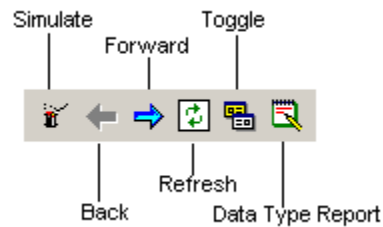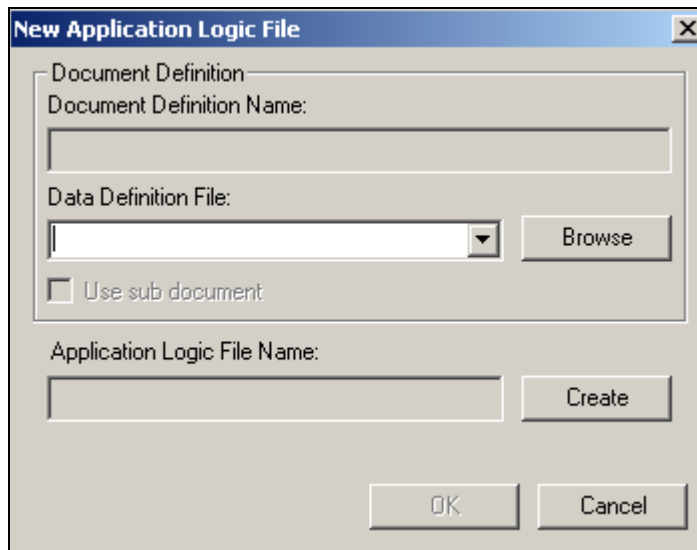
Creating a New Application Logic File

For a View that is Not a Sub-Account

To create a new ALF (statement or email notification):

1. Maximize the Composer window.

2. Select **File>New ALF** or click the **New Document** ⬜ icon. The New Application Logic File window opens.

3. Click **Browse** and select the appropriate path to the DDF associated with the ALF. (If the selected DDF has a subkey group field, the Use Sub-document option becomes active; you must select this option for a view designed to present a sub-document. See "For a Sub-Account View" on page 30.)

4. Click **Create**. The Create Alf File As dialog opens.
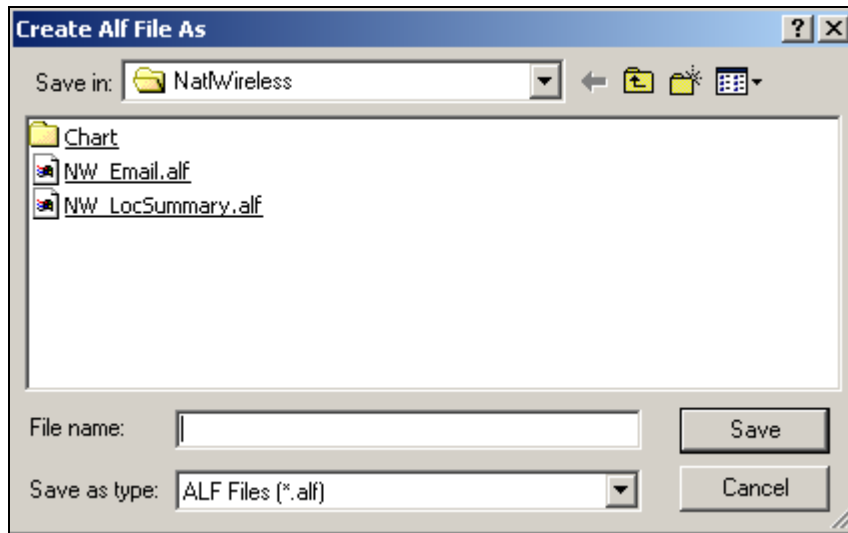
5. Verify that the Save In field points to the correct folder and enter a meaningful name for the ALF.

6. Click **Save**. The New Application Logic File dialog reappears.

7. Click **OK**. The new ALF appears in the Application Tree.

For a Sub-Account View

To use the sub-document indexing feature you **must** create and use a separate Application Logic File (ALF) for presenting a sub-document view.

You must also map the group and subkey to the default template. (You do not need to specify a group template for the sub-documents; the group becomes the main document in this view.)
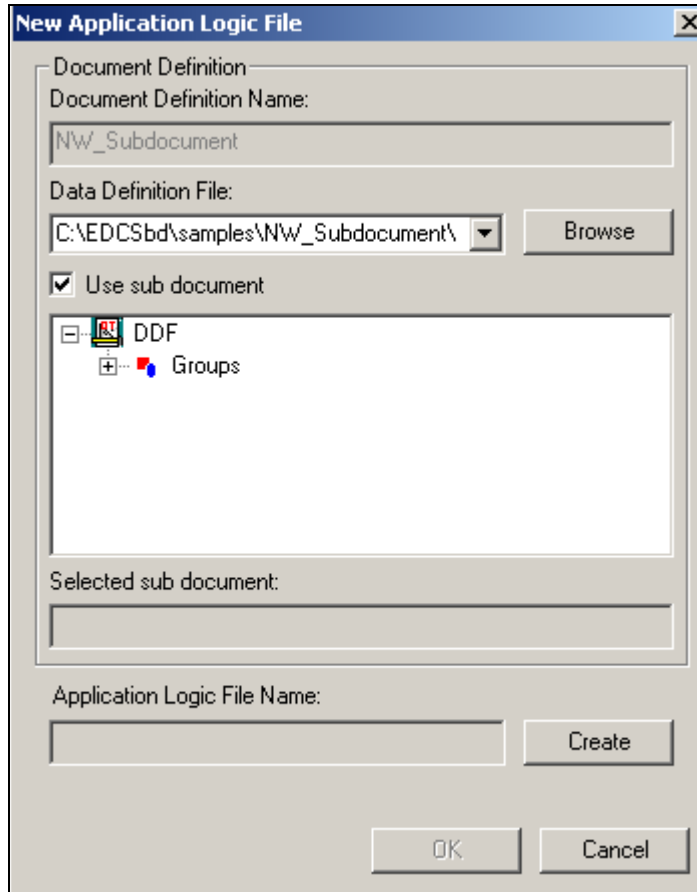
To create a new sub-account ALF and map the data for presentment:

1. In Composer, select **File>New ALF**. Composer displays the New Application Logic File dialog. Select the application DDF you created for this view.
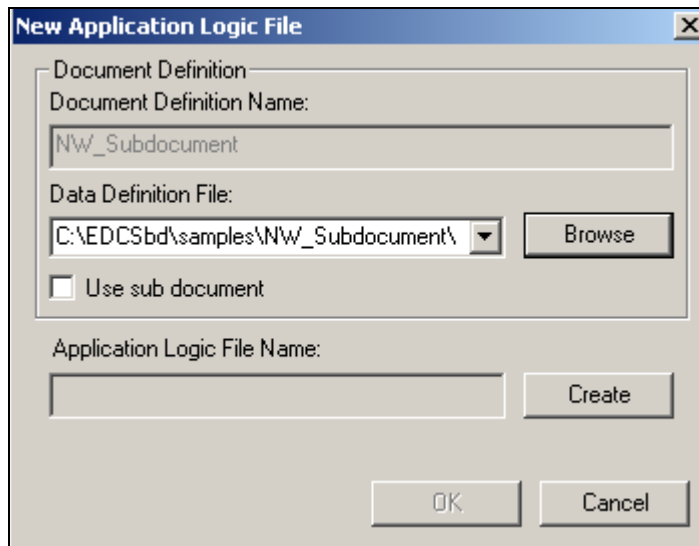
2. Select the **Use sub-document** option. Composer displays a tree showing the sub-document groups in the DDF.

3. Click on the Groups node to expand the list. Select the group containing the subkey for the sub-account you want to use in this ALF view.

4. Click **Create**. The Create Alf File As dialog opens.

5. Verify that the Save In field points to the correct folder and enter a meaningful name for the ALF.

6. Click **Save**.

7. Click **OK**. Note that in the Definition Tree, Composer converts the sub-document group into pure tables, fields into pure fields, and the subkey to the primary key. Composer treats the sub-document as a normal document, not a group (and there is no group template for the group).

Specifying a Default Template

To specify a default template for a statement or email notification ALF:

1. Right-click the 🏠 **Home** icon in the Application Tree.

2. Select **Add Default Template** from the right-click menu. The New Default Template dialog opens.
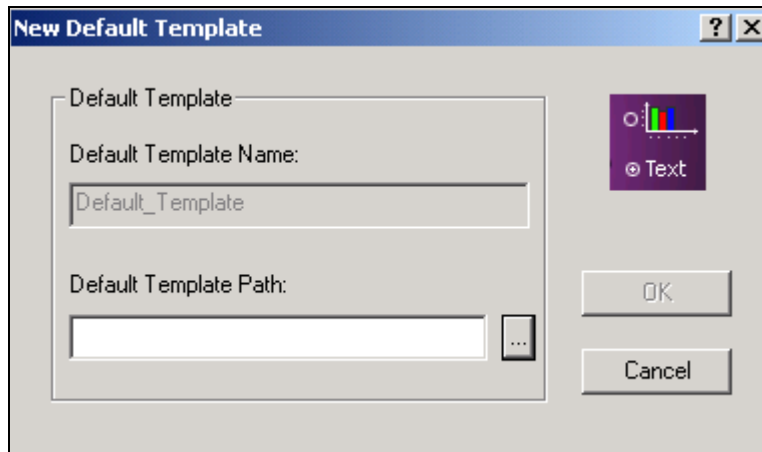
3. Click the ⬚ button next to the Default Template Path. The Open dialog appears.

4. Select the HTML template you want to use as the default.
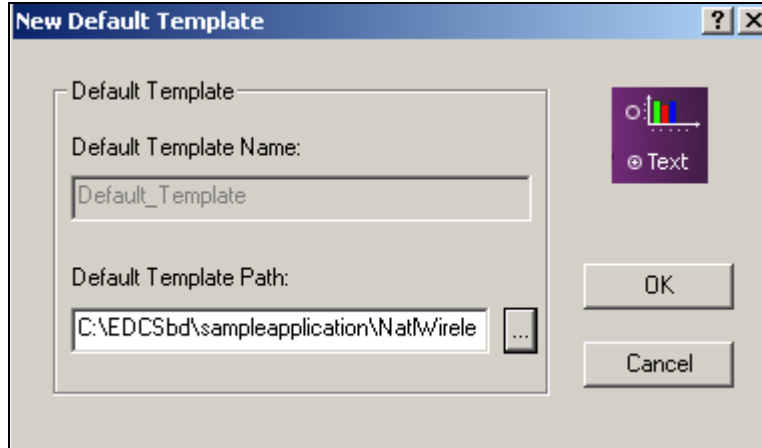
5. Click OK. The template appears in the work area of Composer, and Default_Template appears under the Home node in the Application Tree.

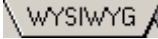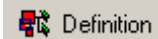| Tip	To open ALFs created in version 1.8 of Composer or earlier, convert the ALF to the new XML format using ALFConverter.exe, found in the *EDCSbd/bin* directory. To open XML files from version 1.9, rename the files with an ALF extension.

Defining Statement Presentment

Mapping Fields to the Template

After creating an ALF and specifying the default template, you can map the data fields defined in the DDF to the template for presentment. (You can also map fields to any conditional HTML templates added).

To map data fields to a statement or email notification template:

1. Click the WYSIWYG tab in the Work Area (if not there already) to display the Default HTML template graphically.

2. Click the Definition tab in the Tree area and double-click the **Fields** node to expand the list of items in your DDF, for example:

3. Click the field you want to map, then drag and drop it where you want it to appear in the template. (Fields in the template appear as `[E]Fieldname,F[/E]`. The field assumes the properties of the area in which you place it. "F" indicates the element is a field.)

4. Add any field formatting you want. To format a field, highlight the entire field tag, including the `[E]` and `[/E]` tags, and use the Format menu or the Formatting toolbar to apply HTML formatting.

5. You can optionally add a notification link to the template. A notification link creates a direct link to the production web server, based on the configuration settings in the eaDirect Command Center. See "Mapping a Notification Link" on Page 75 for details.

6. If your HTML template used "XX" placeholders for fields or tables, delete them. Select `Edit>Find Text` to search the HTML for any placeholders; verify that you've mapped all the required values to the template and deleted all the placeholders.

7. Click the [icon] icon to run the Simulator and check your work. Make any changes necessary, running Simulator as often as needed. (The NatlWireless statement template should look like this:)
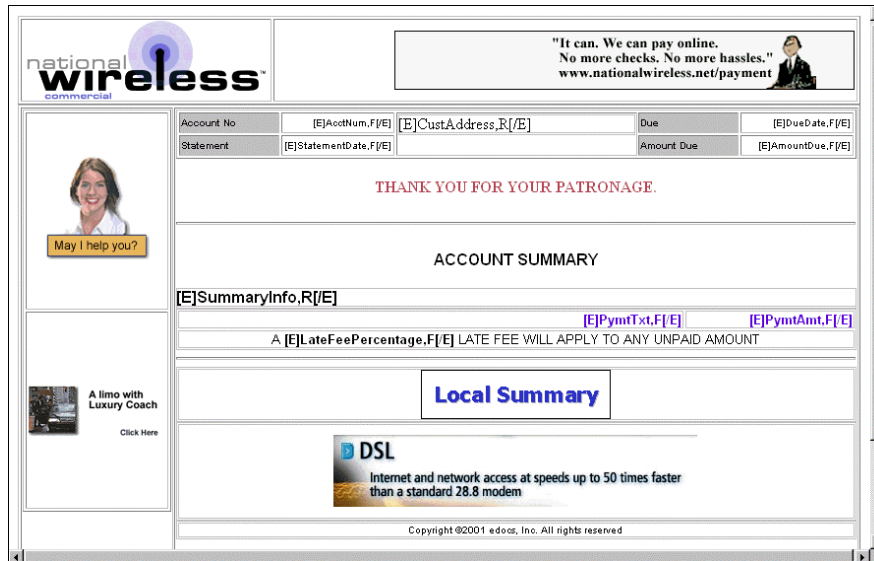
8. Click the Save Template [icon] icon to save the changes to your template.

Note that if you delete a field, table, or group from the DDF (using DefTool) after mapping it to an HTML template in Composer, you must manually delete the associated tag in the template.

Inserting Images, HTML Code, Hyperlinks, or HTML Tables

You can insert images, HTML code, hyperlinks or HTML tables into the statement template.

To insert an image:

1. With the ALF and statement template open in Composer, place the cursor on the template where you want to insert the image, and click ▨ or select **Insert>Picture.** The Picture dialog appears.
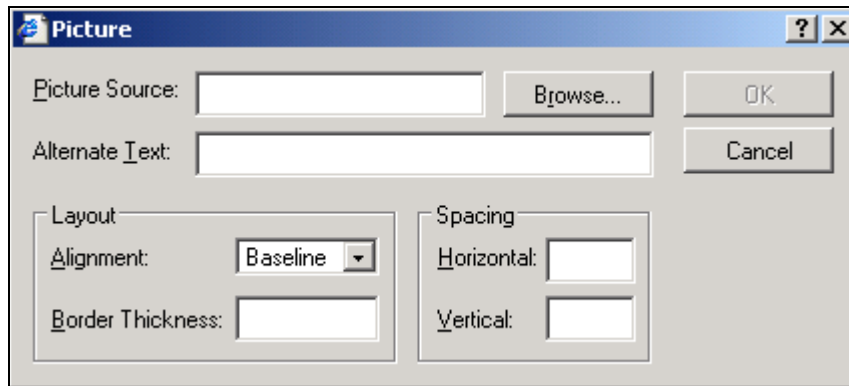
2. Click **Browse** and select the image.

> **Caution** When referencing images or other files in the template, don't use literal paths, such as "img src=c:\images\logo.gif". Instead, use URL addresses, such as:
>
> `http://www.edocs.com/images/logo.gif`

3. Enter any text to use as a possible alternative to the image.

4. Specify the layout and spacing information and click **OK**.

To insert hyperlinks:

1. With the ALF and statement template open in Composer, select the text or object to use for the link.

2. Click ⚓ or select **Insert>Anchor**. The Hyperlink dialog appears.
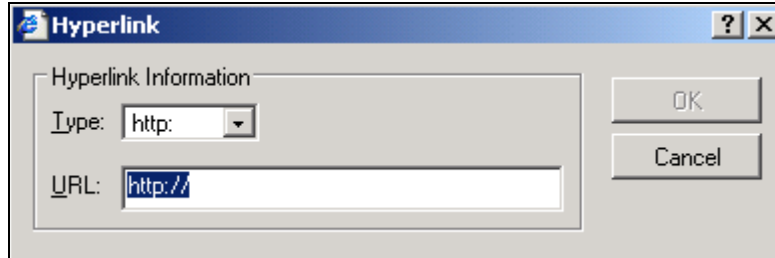
3. Choose the hyperlink type and enter its URL information.

4. Click **OK**. Text links appear underlined and blue in the template.

To insert HTML tables:

1. With the ALF and statement template open in Composer, place the cursor in the desired location and select `Table>Insert Table`. The Insert Table dialog appears.
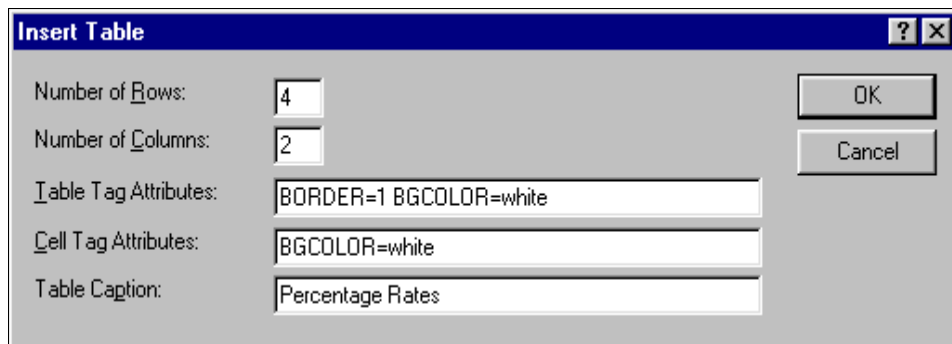
2. Specify the number of rows and columns.

3. Specify the table attributes.

4. Specify the cell attributes.

5. Enter a table caption, if desired.

6. Click **OK**.

To insert HTML text:
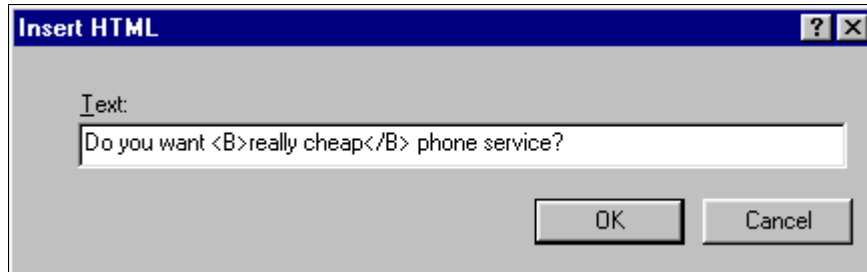
1. With the ALF and statement template open in Composer, place the cursor in the desired location and select `Insert>HTML`. The Insert HTML dialog appears.

2. Enter the text, with any HTML commands.

3. Click `OK`. The text appears.

Creating Conditions

Composer lets you specify data-driven conditions that can instruct eaDirect to do one of the following when composing an individual statement or email:

*   Use an alternate ALF (switch ALF condition)

*   Use an alternate HTML template for presenting the statement

*   Insert or use alternate images, text, or HTML file in a section of a statement or in an individual table row (conditions on a table can also include font and cell formatting)

For example, you could define a condition that causes a different HTML template to be used depending on the value in the total amount field of a statement. You can target marketing campaigns and promotions to specific users based on the data in their statement.

In addition, a non-data driven condition on a table can apply formatting and content to a select pattern of rows.

The Composer interface lets you build the conditional statements by selecting the individual operators, logic elements, content, etc. Similarly, you can pick and choose components to create individual logic elements to use in your conditions. You can create and name logic elements anytime, making them available to choose from a drop-down list when creating conditional statements.

Creating a condition on a table lets you add images, links, formatting, etc. directly into the line of a statement, letting you focus your marketing campaign at the table data level. Creating a "condition on a line item" lets you format the cells or rows in an alternating pattern you define. See Chapter 4 for information about using conditions on tables.

A simple conditional statement takes the form:

```
IF (logic element) THEN content
```

For example:

```
IF (B2Ccustomer) THEN B2CMessage
```

Creating a condition on a section adds **<EDOCSSECTION = SECTIONNAME>** open and close tags to the HTML template.

Creating a Logic Element

A logic element is an expression that compares the values in two data fields or between a data field and a specific value. You use one or more logic elements to build conditional statements in Composer.

You can create logic elements while defining a condition or create logic elements first, making them available to select from a drop-down list when creating the condition. You define and save logic elements with a name, making them available for reuse in other conditions.

A logic element can compare:

- The values in two fields, group fields, or table columns
- The value in a field, group field, or table column to a specific value such as a date, time, amount, NULL or EMPTY, or a regular expression
- A table or group with NULL or EMPTY (to test for existence)

Content is what eaDirect inserts or uses in place of something else in a statement when the condition is true. Content can be an image, text string, HTML section or HTML template within a section of a template. In tables, content can also include font and cell formatting; see "Creating Conditions on a Table (Line Item)"on Page 83 for a description of content you can apply to tables (table content). (Any alternate HTML templates or ALFs are "content" for those types of conditions only.)

You can create conditions with multiple logic elements using the AND operator or the OR operator, for example:

```
IF (B2CCustomer AND AmountDue1000) THEN B2CMessage
```

Additional conditions in a statement must start with ELSE IF.

Logic element syntax

Each logic element requires the following information and syntax:

- **The data type;** DATE, TIME, CURRENCY, INTEGER, or DECIMAL (not necessary for a string).

- **An open parenthesis** (not needed if the data type is a string).

- **The field, group field, table, column, or group** in format **/BillDate[].**

- **The data format** if the data type is DATE, TIME, CURRENCY, or DECIMAL. For DATE or TIME you must manually enter any separators, such as forward slashes in the date: %d/%m/%Y.

- **A close parenthesis** (not needed if the data type is a string).

- **An operator for the condition**; ==, != (not equals), <, >, <=, >=, ~, or TRUE, AND, OR or NOT. (Use the tilde "~" before a regular expression.)

- **A value to compare with the first part of the logic element**. This can be another field, group field, table, column, specific date, time, integer, or string, or a regular expression, along with the data type and format (plus separators), where appropriate.

The following examples of logic elements show a variety of logic comparisons:

```
/FirstStmtIndicator[]!=NULL
```

```
/CustType[]== "B2B"


/SummaryInfo[]/SummaryInfoCR[]=="CR"


DATE (/StatementDate[],"%d/%m/%Y")>= DATE
("11/18/2002","%d/%m/%Y")


CURRENCY (/CurrentCharges[],'.', "")> CURRENCY("200.00",
'.', "")
```

You can use a relative path if the section (condition) is on extracted data in the table/group HTML. You can use always use an absolute path.

## Creating logic elements on group fields

You can create a logic element on sub-account group fields, including a subkey, or on a group field and a table column. These elements can be at different levels, but must both be within the same sub-account group hierarchy.

In the case where you have nested sub-account groups, any conditions you create on the nested sub-account group apply only to that group and not to the higher-level, or parent sub-account group. Any condition you create on the parent sub-account group likewise applies to the parent sub-account only, and not to any nested sub-account groups. (Nested sub-account groups act like regular groups since they are just a subset of the parent group view.)

If a condition on a group field adds a section to a group template, the group field must belong to that group or to a child group of the one where the section appears. The condition cannot add a section to a group that is a child of the group the field belongs to.

To create a logic element:

1. With the ALF open in Composer, right-click the ▣ Logic Elements node in the Application Tree and select `Create Logic Element` from the right-click menu (or click the appropriate Create button while creating a condition). Composer displays the Create Logic Element dialog.
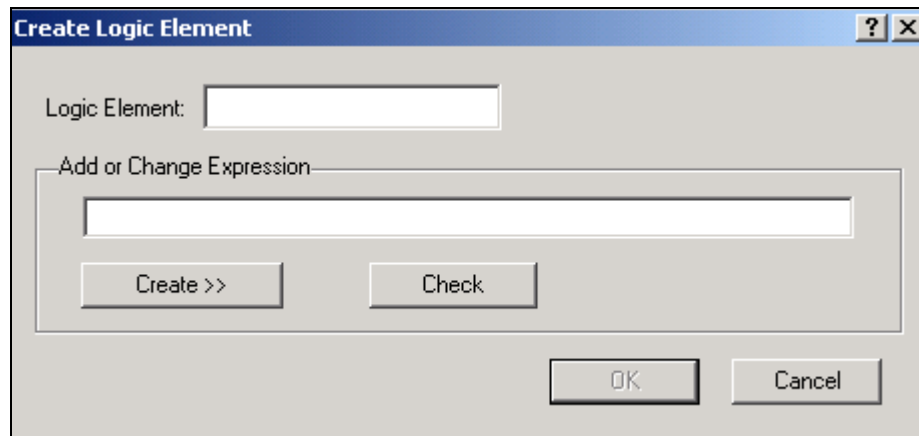
2. Type a meaningful name for the logic element in the Logic Element box.

3. Click [Create >>] to display the buttons for selecting fields, operators, dates, etc. to add to the logic element. Select or enter items in the sequence you want them to appear in the logic element. (You can manually enter the logic element in the Add or Change Expression box, but choosing from the menus reduces typing and syntax errors.)
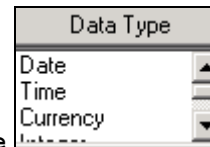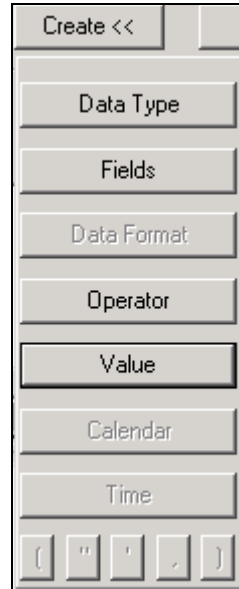
4. If comparing a field or column, click **Data Type** and
    select the type of data. It is not necessary to select a data type for strings.
    Composer adds the data type and an open parenthesis to the expression:
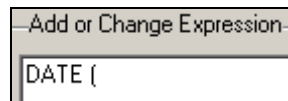
5. Click **Fields,** expand the list, and select the field,
    table, column, group, or group field you want to compare. Composer adds
    the column name (and a comma, if data type is not a string) to the
    expression:

Add or Change Expression

DATE [/DueDate[],
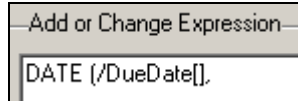
6. If you choose a table column, the Set Occurrence Value dialog appears. Specify the row occurrences you want eaDirect to check. To specify a basis other than the default (empty brackets default to Current Occurrence), click on the table name and edit the value between brackets [] as described here:

- [] Current Occurrence – Evaluate each row in the table and apply the content to each qualifying row

- [?N] Any N Occurrence – If N rows of qualifying data exist in the table, then apply the content

- [*] Every Occurrence – Every row must meet the condition, then apply the content

- [N] N Group/Table Occurrence or Column Line – If the Nth table row qualifies, then apply the content

Be sure and specify the same row occurrences on both sides of a logic element if you are comparing two column values.

**Select Occurrence value**

LocalChargeSummary[]
  LocalChargeAmt[*]

[ ]   - Current Occurrence

[? N] - Any N Occurrence

[*]   - Every Occurrence

[N]   - N Group/Table Occurrence
              or Column Line

OK   Cancel
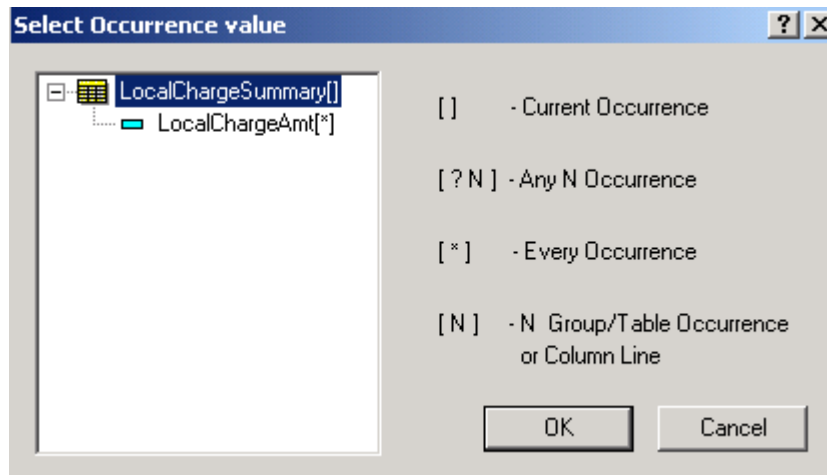
7. If the data type of the comparison field or column is a date, time, currency, or

> Select a date format
>
> %d%m%Y
> %d%m
> %Y%m%d
> %d%b%Y
> %d%b%y
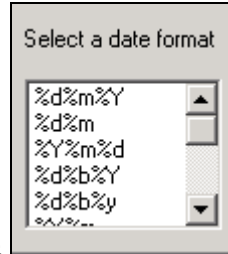
decimal, click **Data Format** and select the format of the data. (For additional information about valid format strings you can enter, see Appendix A.) Composer adds the data format and a close parenthesis to the expression. Manually add any required separators.

> Add or Change Expression
>
> DATE (/DueDate[],"%d/%m/%Y")

> ==
> !=
> <
> >
> <=
> >=
> ~
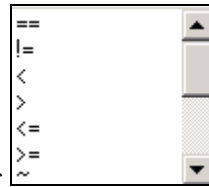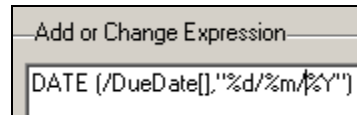
8. Click **Operator** and select an operator from the drop-down list. (Choose the tilde "~" operator with a regular expression.) Composer adds the operator to the expression.
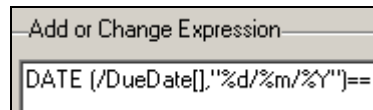
> Add or Change Expression
>
> DATE (/DueDate[],"%d/%m/%Y")==

9. Click **Data Type** and select the same data type you selected in the first part of the logic element (if any). Composer adds the data type and an open parenthesis to the second part of the expression.

> Add or Change Expression
>
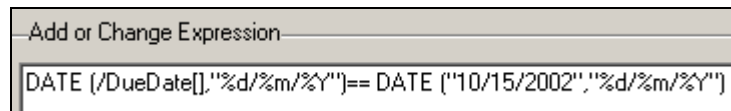> DATE (/DueDate[],"%d/%m/%Y")== DATE (
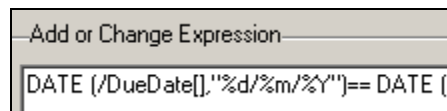
10. Specify the field, group field, column, date, time, or value for comparison on the right side of the logic element expression. You can only compare groups and tables with NULL or EMPTY.

- **To compare the data to a field or column**, click `Fields` and choose the field or column, adding any necessary separators. (Note that if you are comparing group fields, this field must be higher in the hierarchy than the one on the left side of the expression.) Composer adds the column name in quotation marks, followed by a comma, to the expression.

- **To specify a date for comparison**, click `Calendar` and choose a specific month, year, and day. Composer adds the selected date in quotation marks, followed by a comma, to the expression.
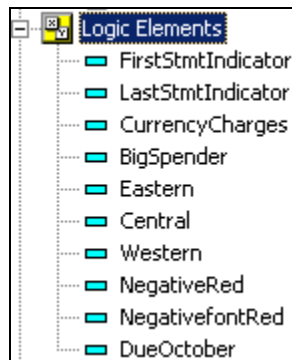

- **To specify a time for comparison**, click `Time`. Click the hour, minutes, seconds, or AM/PM to highlight, and use the up and down arrows to specify the value: . When the time you want appears, double-click to enter the time in the expression. Composer adds the selected time in quotation marks to the expression.

- **To specify an integer, decimal, or string value for comparison**, manually type the value between quotation marks followed by a comma. Add a close parenthesis after an integer value. (Click buttons or type directly into the expression.)

- **To compare a string, table, or group to NULL or EMPTY**, click
  `Value` and select NULL or EMPTY. Composer adds the word NULL or
  EMPTY to the expression.

11. If the data type is a date, time, currency, or decimal, click `Data Format` and
    select the same format you selected in the first part of the expression,
    manually entering any required separators. `Composer adds the data
   format, followed by a close parenthesis, to the
    expression.`

```
─Add or Change Expression─────────────────────────────────────────
DATE (/DueDate[],"%d/%m/%Y")== DATE ("10/15/2002","%d/%m/%Y")
```

12. When the logic element expression is complete, click `Check` to verify that
    the logic element is valid. Correct or complete the logic element as needed,
    checking again to verify any changes.

13. Click `OK`. Composer adds the logic element to the list in the Application
    Tree.

```
Logic Elements
    ─ FirstStmtIndicator
    ─ LastStmtIndicator
    ─ CurrencyCharges
    ─ BigSpender
    ─ Eastern
    ─ Central
    ─ Western
    ─ NegativeRed
    ─ NegativefontRed
    ─ DueOctober
```

To view logic elements:

- **To view a particular logic element expression**, with the ALF open in
  Composer, expand the ⊞ Logic Elements node and right-click the logic
  element name in the Application tree. Select `View Logic Element` from
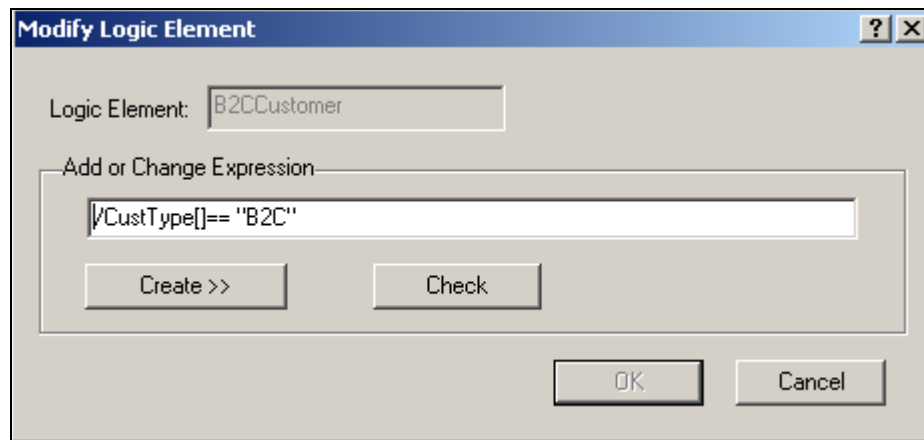 the right-click menu.
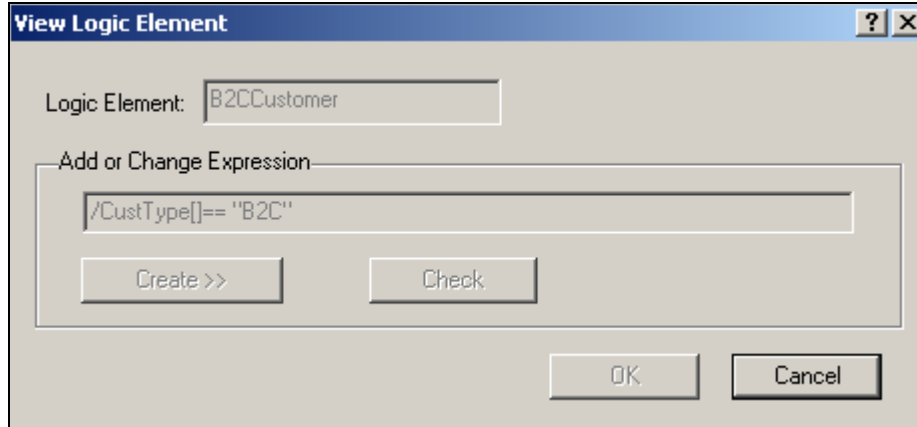
- **To view a list of all logic element expressions**, with the ALF open in Composer, right-click the ⊞ 📇 Logic Elements node in the Application Tree and select **Logic Elements List** from the right-click menu. Composer displays the Logic Elements List.
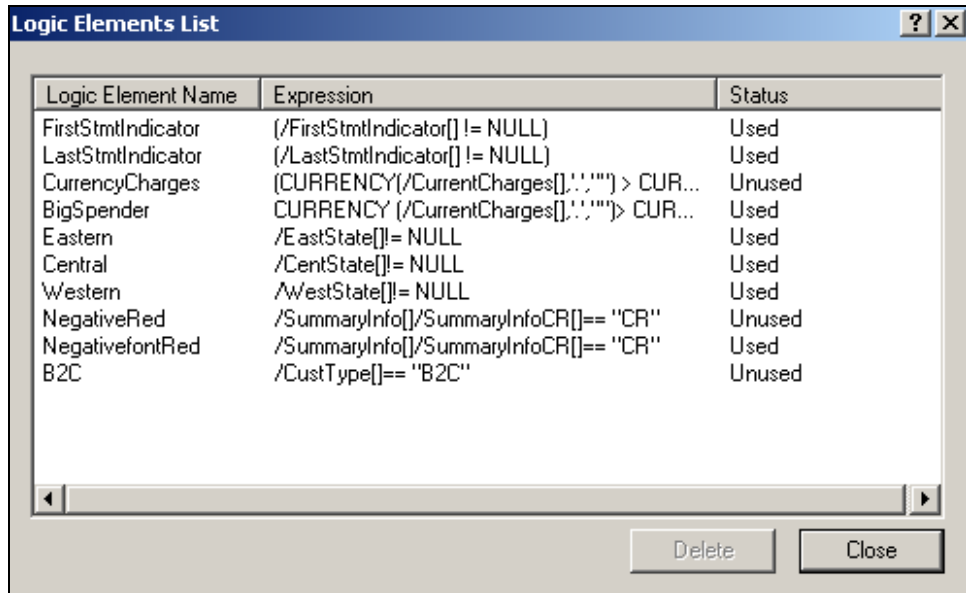
To edit a logic element expression:

1. With the ALF open in Composer, expand the ⊞ 🔣 Logic Elements node and right-click the logic element name in the Application tree.

2. Select **Edit Logic Element** from the right-click menu. Composer displays the Modify Logic Element dialog.

3. Type your edits or use the Create drop-down menu. See Creating a Logic Element on Page 41 for details.

4. Click **Check** to verify that the edited expression is valid.

5. Click **OK**.

To delete an unused logic element:

1. With the ALF open in Composer, right-click the ⊞ 🔣 Logic Elements node and select **Logic Elements List** from the right-click menu. Composer displays the Logic Elements List.

2. Select the unused logic element and click **Delete**.

3. Click **Close**.

Creating an ALF Switch Condition

You can create a data-driven condition that instructs eaDirect to compose and present a customer's statement using an entirely different ALF.

An ALF switch condition gives you the greatest level of conditional control over what you present to a customer. Using the same DDF, you can create a completely different-looking statement and present an alternate set of data for targeted customers. For example, you could create a condition to specify switching ALFs for larger customers whose data requires different table or charting formats.

Before creating a switch ALF condition, you must determine whether it makes more sense to use entirely separate views or to create a condition on a view.

Each ALF switch condition requires one or more logic elements. A logic element is an expression that compares the values in two data fields or between a data field and a specific value. You can create logic elements while defining a condition or create logic elements first, making them available to select from a drop-down list when creating the condition. See Creating a Logic Element on Page 41 for details.

To create an ALF switch condition:

1. Right-click the ![icon] Switch node in the Application Tree and select **Add ALF** from the right-click menu, or click the **Switch ALF** ![icon] icon in the ALF toolbar. The Add Switch ALF dialog appears.
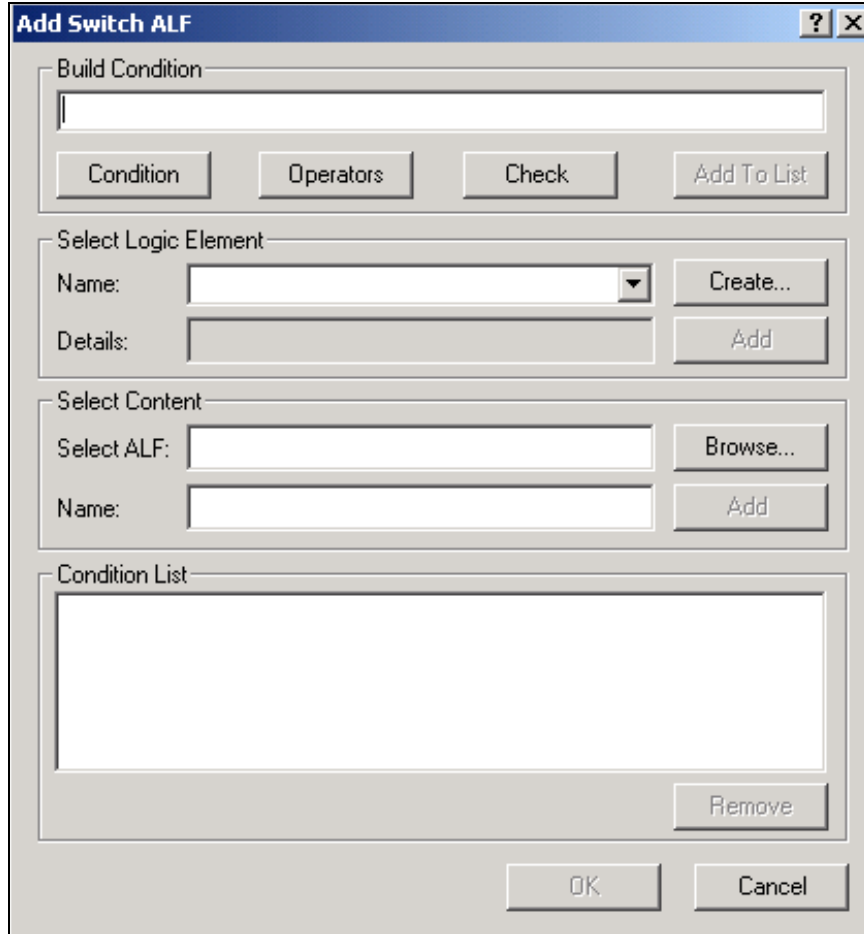
2. Use the options on this dialog to "build" a conditional statement in the Build Condition field. (You can manually enter the text in the Build Condition field, but choosing from the menus reduces typing and syntax errors.)

```
IF (           ELSE IF (
  THEN           THEN
  ELSE           ELSE
}              }
(              (
```
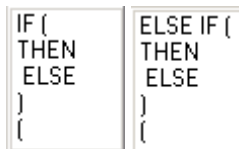
3. Click **Condition** If this is the first ALF switch you're creating, select **IF** from the drop-down list. If you've already added one or more ALF switch conditions, select **ELSE IF**.

4. Under Select Logic Element, click **Create** to create a logic element for the ALF switch; see "Creating a Logic Element" on Page 41 for details. If you have already created the logic element, choose it from the Name drop-down list and click **Add** to add it to the condition.

5. Click **Operators** to select any operators your conditional statement requires

    ```
    AND
    OR
    NOT
    ```
    .
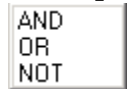
6. Add any additional logic elements the condition requires.

7. Click **Condition** again and select the right parenthesis from the drop-down list. One set of parenthesis must enclose all logic elements in a condition.

8. Click **Condition** and select **THEN**.

9. Click **Browse** to select the ALF (content) to switch to.

10. Type a name for the switch ALF content in the Name field and click **Add** to add it to the condition.

11. When the switch ALF condition is complete, click **Check** to verify that it is valid. Correct or complete the statement as necessary, checking the condition again.

12. When the condition is complete and valid, click **Add to List**. Composer displays the condition in the Condition List at the bottom of the dialog.
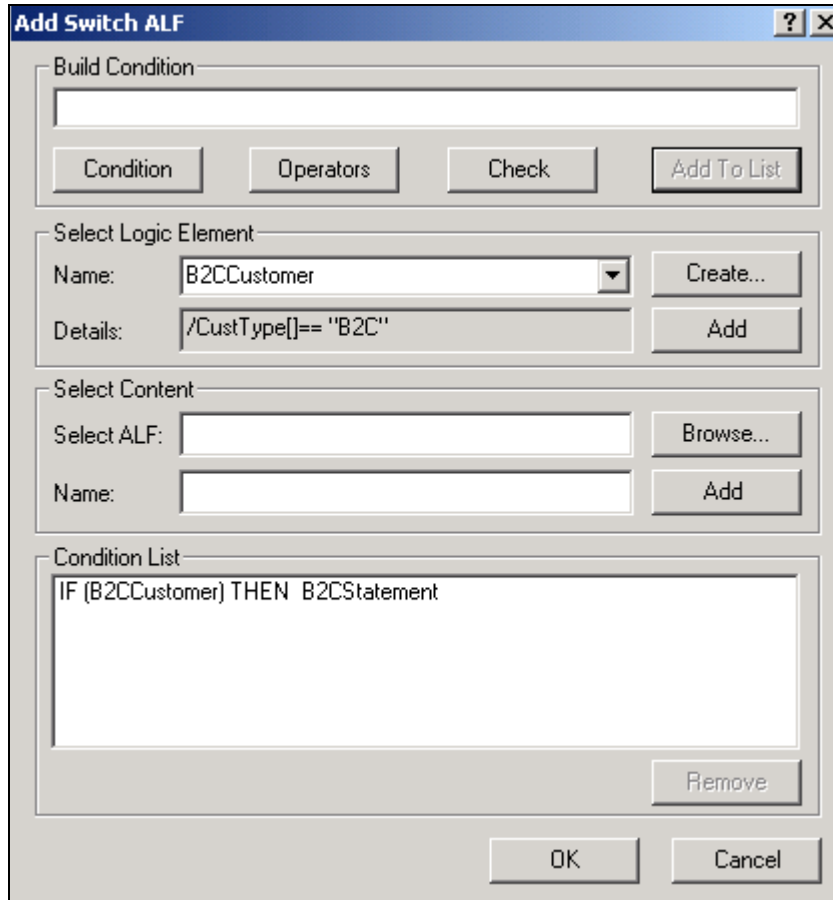
13. Click OK. The added switch condition appears in the Application Tree:

To view properties of all ALF switch conditions:

1. With the ALF open in Composer, right-click **IF=** Condition below the
   Switch node in the Application Tree.

2. Choose **`View Properties`** from the right-click menu. Composer displays the Switch Condition dialog showing all ALF switch conditions defined.

3. Click **`OK`**.

To edit an ALF switch condition:

1. With the ALF open in Composer, right click the ![Switch] Switch node in the Application Tree.

2. Select **`Modify Condition`** from the right-click menu. Composer displays the Modify Switch ALF dialog.

3. Double-click the condition from the Condition List and edit it as necessary.

To change the ALF used in a switch condition only:

1. With the ALF open in Composer, right click the content name of the ALF listed under **IF= `Condition`** below the ![Switch] Switch node in the Application Tree.

2. Select **`Change`** from the right-click menu.

3. Composer asks if you are sure you want to change the ALF. Click **`Yes`**.

4. Choose a new ALF from the Change ALF dialog and click **`OK`**.

To delete an ALF switch condition:

1. With the original ALF open in Composer, right click the content name of the ALF listed under **IF= `Condition`** below the ![Switch] Switch node in the Application Tree.

2. Select **`Delete`** from the right-click menu.

Adding Conditional Templates

You can create a condition that instructs eaDirect to compose a statement or document using an alternate template (not the default template) under specific data-driven circumstances. This lets you customize the presentation of statements based on the data they contain.

Each condition requires one or more logic elements. A logic element is an expression that compares the values in two data fields or between a data field and a specific value. You can create logic elements while defining a condition or create logic elements first, making them available to select from a drop-down list when creating the condition. See Creating a Logic Element on Page 41 for details.

To add a conditional HTML template:

1. With the ALF open in Composer, right-click the Home node in the Application Tree and select **Add Template** from the right-click menu, or click . Composer displays the Add Conditional Template dialog.
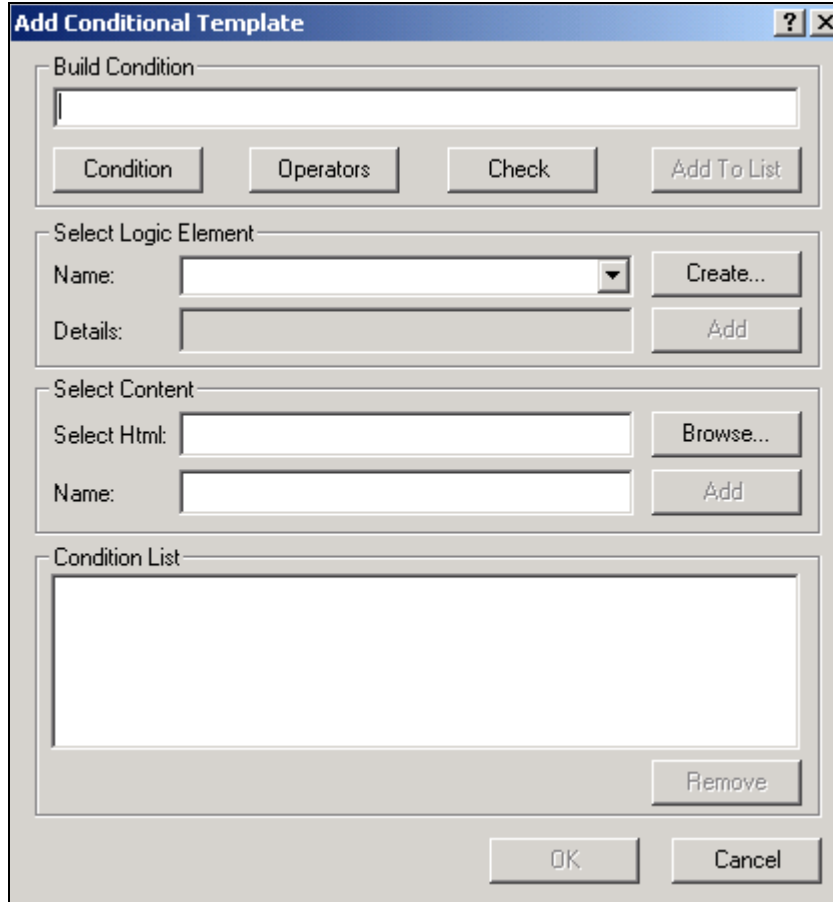
2. Use the options on this dialog to "build" a conditional statement in the Build Condition field. (You can manually enter the text in the Build Condition field, but choosing from the menus reduces typing and syntax errors.)
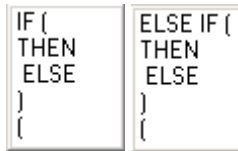
3. Click **Condition:** If this is the first alternate template you're adding to the ALF, select **IF** from the drop-down list. If you've already added any additional templates, select **ELSE IF**.

4. Under Select Logic Element, click **Create** to create a logic element for the template condition; see Creating a Logic Element on Page 41 for details. Click **Add** to add it to the condition.

5. Click **Operators** to select any operators your conditional statement requires

    ```
    AND
    OR
    NOT
    ```
   .
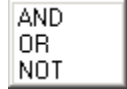
6. Add any additional logic elements the statement requires.

7. Click **Condition** again and select the right parenthesis from the drop-down list. One set of parenthesis must enclose all logic elements in a condition.

8. Click **Condition** and select **THEN** from the drop-down list.

9. Click **Browse** to select the alternate HTML template file (content).

10. Type a name for the alternate HTML template in the Name field and click **Add** to add it to the condition. Names must not contain spaces or special characters, and must start with an alpha character. The names must not exceed 30 characters in length.

11. When the condition is complete, click **Check** to verify that it is valid. Correct or complete the statement as necessary, checking the condition again.

12. When the condition is complete and valid, click **Add to List**. Composer displays the condition in the Condition List at the bottom of the dialog.
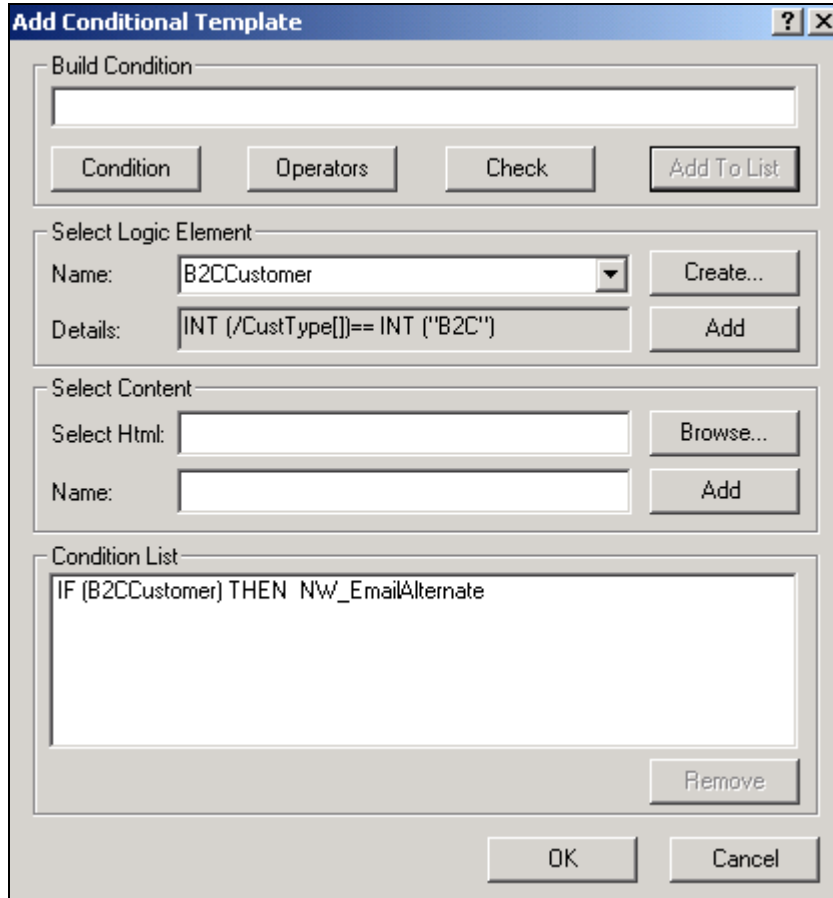
13. Add any additional conditional templates, if necessary.

14. Click **OK**. Composer adds an **IF= Condition** node listing the template conditions (along with the default template) for the ALF:
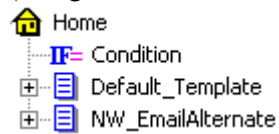
To view properties of all template conditions:

1. With the ALF open in Composer, expand the Home node in the Application Tree and right-click IF= Condition .

2. Choose **View Properties** from the right-click menu. Composer displays the Template Condition dialog showing all template conditions defined.

3. Click **OK**.

To edit a template condition:

1. With the ALF open in Composer, right-click the Home node.

2. Select **Modify Condition** from the right-click menu. Composer displays the Modify Conditional Template dialog.

3. In the Condition List, double-click the condition you want to change and edit as necessary.

To change the default template file or the template specified as a conditional alternate:

You can change the HTML template file you want to use as the default for a view. You can also change the template file specified in a condition as an alternate template.

1. With the ALF open in Composer, right-click the content name of the template in the Application Tree.

2. Select **Change** from the right-click menu. Composer displays the following message:
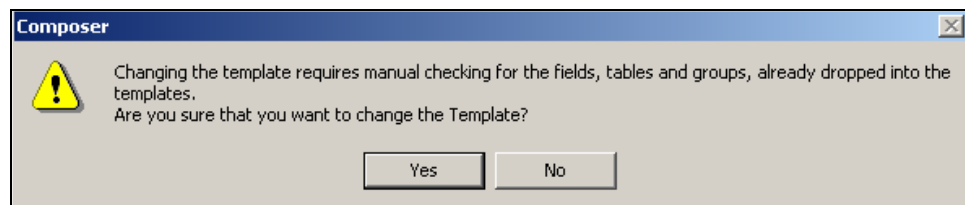
3. Click **Yes**. Composer displays the Change Template File dialog.

4. Choose the new template you want to use. If you added sections and groups to the old template in the ALF, Composer displays the following warning:
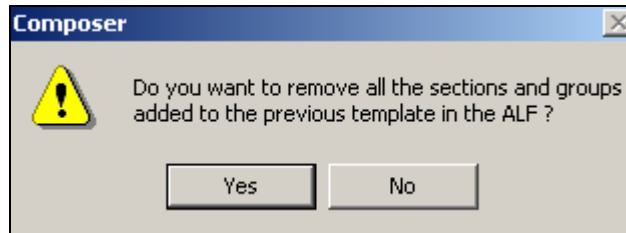
5. Click **Yes.** Composer removes all the sections and groups defined on that template from the ALF.

To delete a conditional template from an ALF:

1. With the ALF open in Composer, right-click the template you want to delete.

2. Select **Delete** from the right-click menu. Composer asks if you are sure you want to delete the template.

3. Click **Yes** to delete the template. Composer removes the template from the Application Tree.

Creating a Condition on a Statement Section

You can create a data-driven condition on a selected area of a statement that instructs eaDirect to replace the section with a different image, HTML file, text content, or NULL when composing the statement. This lets you tailor advertisements or marketing messages based on the data in a customer's statement.

You can define a condition on any section of a template, such as a block of text, or an inserted field, or image.

Each condition requires one or more logic elements. A logic element is an expression that compares the values in two data fields or between a data field and a specific value. You can create logic elements while defining a condition or create logic elements first, making them available to select from a drop-down list when creating the condition. See "Creating a Logic Element" on Page 41 for details.

You must identify each image, block of text, and HTML file you plan to use in a conditional statement as content in Composer. You must name each item you want to conditionally insert in the template. The content name acts as a pointer to the image, text, or HTML file. You can create content while building a condition on a section or create content independently; see "Creating Content" on Page 70.

You can create compound conditions on a section (IF THEN…ELSE IF) and add additional conditions at any time. To create a compound condition on a section, you must build, check, and add the first part of the expression and the ELSE IF portion separately. For example, to create the compound statement:

```
IF (StateFL) THEN OrangeJuiceAd ELSE IF (StateGA) THEN
PecanPieAd
```

you must build and check "`IF (StateFL) THEN OrangeJuiceAd`" and add it to the conditions list. Then you can build, check, and add "`ELSE IF (StateGA) THEN PecanPieAd`" to the list of conditions for the section.

To create a condition on a statement section:

1. With the ALF open in Composer, select the section (text, image, etc.) on which you want to create a condition. (Select **THANK YOU FOR YOUR PATRONAGE.** on the default template in NatlWireless.alf).

2. With the section highlighted, right-click and select **Add Section** from the right-click menu. The Create Section dialog appears:
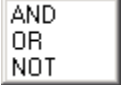
3. Enter a name for the section. Names cannot contain spaces or special characters, and must start with an alpha character. Maximum length is 30 characters. Use meaningful names; "BigSpenderInsertAirlineOffer" is more descriptive and easier to identify than "ALFsection1".

4. Use the options on this dialog to build a conditional statement in the `Build Condition` field. You can manually enter the text in the Build Condition field, but choosing from the menus reduces typing and syntax errors. (You can optionally choose a previously saved condition from the drop-down list if you've already created a condition you want to reuse in this section. If so, skip to Step 13.)

```
IF (
THEN
ELSE
)
(
```

5. Click `Condition` and select `IF` from the drop-down list.

6. Under Select Logic Element, click `Create` to create a logic element for the condition; see "Creating a Logic Element" on Page 41 for details. If you have already created the logic element, choose it from the Name drop-down list and click `Add` to add it to the condition.

7. Click `Operators` to select any operators your conditional statement requires

```
AND
OR
NOT
```
.

8. Add any additional logic elements the conditional statement requires.

9. Click `Condition` again and select the right parenthesis from the drop-down list. One set of parenthesis must enclose all logic elements in a condition.

10. Click `Condition` and select `THEN`.

11. Choose the content to insert from the `Name` drop-down list under Select Content, or click `Create` to create new content. See "Creating Content" on Page 70 for details.

12. Click `Add` to insert the content name in the condition.

Create Section

Section Name: NewCustMsg   ☐ eaMarket enabled

Build Condition

IF (BigSpender) THEN NewCustMsgText

[Condition]   [Operators]   [Check]   [Add to List]

Condition Name: [▼]   [Save]

Select Logic Element

Name: BigSpender   ▼   [Create...]

Details: [/FirstStmtIndicator[] != NULL)   [Add]

Select Content

Name: NewCustMsgText   ▼   [Create...]

Type: Text   [Add]

Details: as WELCOME TO NATIONAL WIRELESS!
WE LOOKFORWAR

Condition List

[   ]

[Remove]

[OK]   [Cancel]

13. When the section condition is complete, click **Check** to verify it is valid. Correct or complete the statement as necessary, checking the condition again.

14. If you want the conditions on this section to be available in eaMarket, click the **eaMarket enabled** checkbox.

15. When the condition is complete and valid, click **Add to List**. Composer displays the condition in the Condition List at the bottom of the dialog.

16. If you want to save the condition for possible reuse in another section, click the condition in the Condition List and type a name for the condition in the Condition Name drop-down box (next to the Add to List button). Click **Save**.

17. If necessary, build any additional conditions to create a compound condition on the section. Begin subsequent conditions with ELSE IF.

18. Click **OK**. The condition appears in the Sections node for the template. (If you eaMarket enabled a section, it appears with the eaMarket enabled section icon instead.)
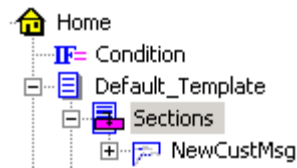
Deleting, Modifying, and Viewing Conditions

To display a list of saved conditions:

- Click the Show Conditions icon on the ALF toolbar. (Note table conditions are not included.)
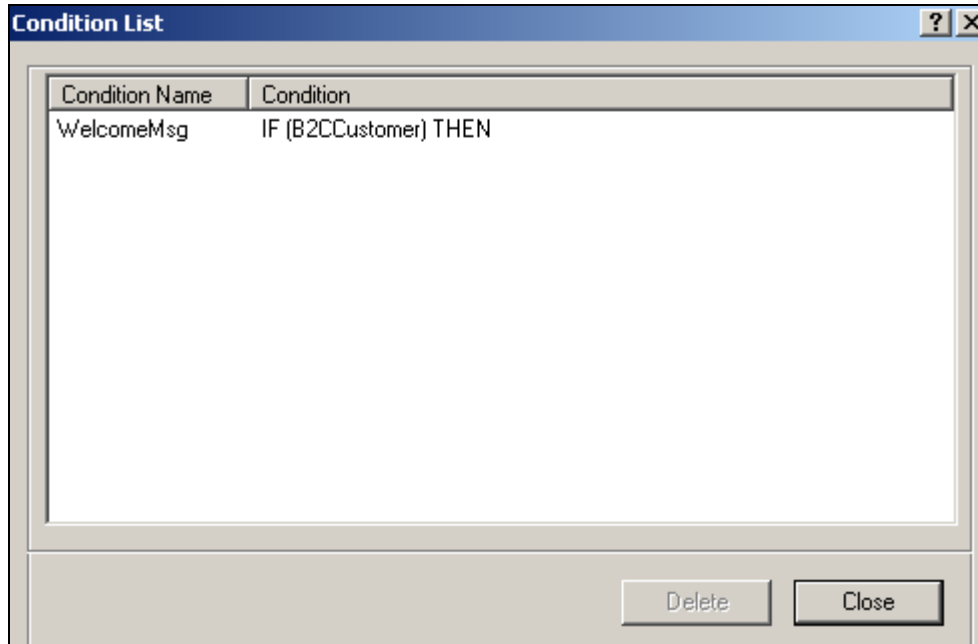
To view the properties of a condition:

1. With the ALF open in Composer, double-click the ![Sections icon] Sections node (below the ![Home icon] Home node). Composer displays the conditions (sections) defined in this template.

2. Double-click the name of the condition (section) you want to view.

3. Right-click on the ![IF= Condition icon] Condition node and select **View Properties** from the right-click menu. Composer displays the Section Condition dialog with the conditional statement.

Section Condition   ? X

Condition :

IF (NewCustomer) THEN WelcomeLogo

                                                    OK
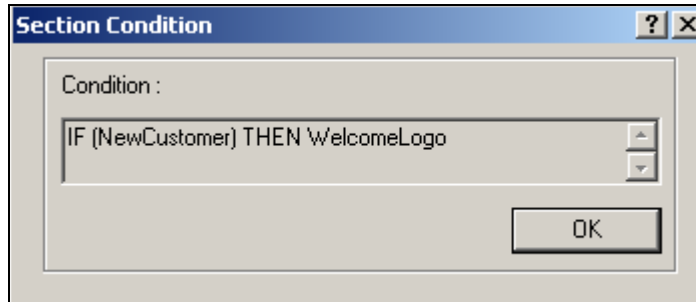
To modify a condition on a section:

You can modify a condition on a section or add one or more ELSE IF conditions to a section (creating compound conditions).

1. With the ALF open in Composer, double-click the ⊞ Sections node (below the 🏠 Home node).

2. Right-click the name of the condition you want to modify.

3. Select **Modify Section** from the right-click menu. The Modify Section dialog appears.

4. To modify an existing condition, double-click on the condition and edit as needed. To add a new condition to the section, follow the procedures above to add an ELSE IF condition.

To delete a condition on a section:

1. With the ALF open in Composer, double-click the ⊞ Sections node (below the 🏠 Home node).

2. Right-click the name of the condition you want to delete.

3. Select **Delete Section** from the right-click menu.

4. Composer asks if you're sure you want to delete the section. Click **Yes**.

Creating Content

Content is any image, "block" of text, or HTML that you conditionally insert in a section of a statement based on the user's data. You must identify each item you plan to use in a data-driven condition as content, and give each a name. The content name acts as a pointer to the image, text, etc.

You must define each of the following items as content to use them in a condition:

- Images

- Blocks of text

- HTML files

- Alternate (switch) ALFs

- Alternate HTML template files

By naming content, it becomes available to select from a drop-down list when building conditions on a section. (You can also create new content when building a condition.)

You must create content to use in table conditions separately; table content is not available for use in general (non-table) conditions and vice-versa. Composer ties table content directly to the particular table condition, and it can include font and cell formatting, unlike non-table content. (You can create some reusable table content for use with other table conditions only.) See "Creating Conditions on a Table (Line Item)" on Page 83 for information.

Creating New Content

Follow these instructions to create content for use in conditions on non-table sections of a statement. (See "Creating Table Content" on Page 87 for information about creating table content.)

To create new (non-table) content:

1. In the Application Tree, right-click the 🗑 Content node and select **Create** from the right-click menu. The Create Content dialog appears.
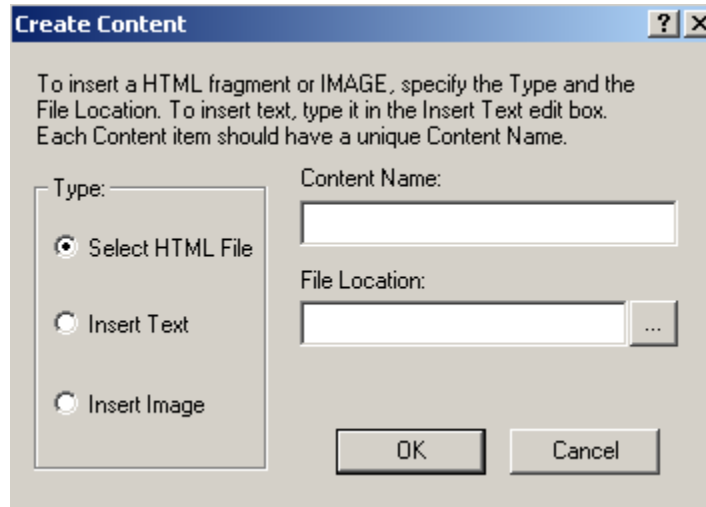
2. Click the content type: Select HTML File, Insert Text, or Insert Image. (Click Insert Text for NatlWireless.)

3. Enter a unique and meaningful name for the content. Names cannot contain spaces or special characters, must start with an alpha character, and cannot exceed 30 characters in length. (Enter NewCustMsgTxt.)

4. Browse the **File Location** field to select an HTML or image file, or type the text content in the **Insert Text** box. (Enter "**WELCOME TO NATIONAL WIRELESS!
WE LOOK FORWARD TO PROVIDING YOU WITH THE BEST WIRELESS SERVICE.**")
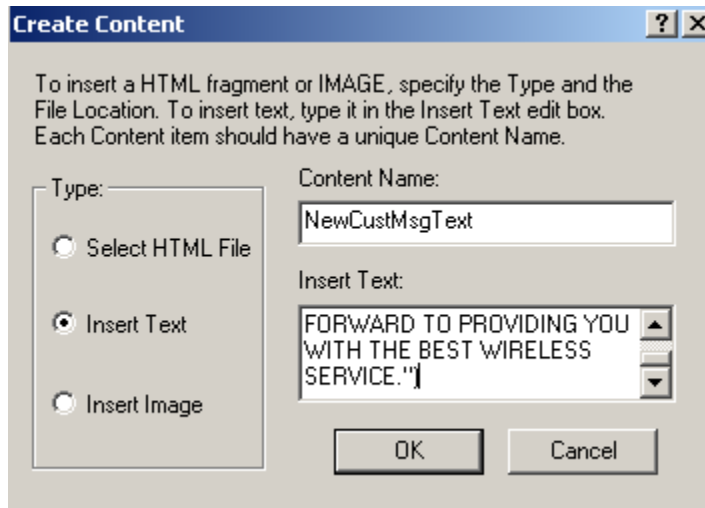
5. Click **OK**. The content name appears in the Application Tree:
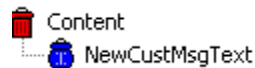
Viewing Content

You can display a list of all contents defined for an ALF showing the content type, details, and status (used or unused). You can optionally display a list of table content only.

You can also view the properties of individual contents.

To view a list of all content and properties:

1. With the ALF open in Composer, right-click the node (or click the node for a list of table content only).

2. Select **Content List** from the right-click menu. Composer displays the Content List dialog.
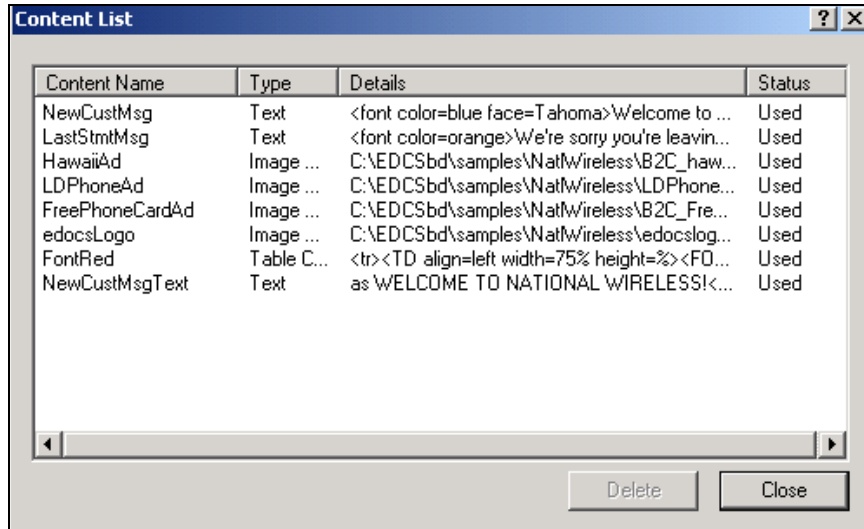
To view properties of individual content:

1. With the ALF open in Composer, double-click the 🗑 Content node to
   display a list of content names in the Application Tree.

2. Right-click the content name and click **View Properties** from the right-
   click menu. Composer displays the View Content dialog (or the View
   Reusable Content dialog for reusable table content).
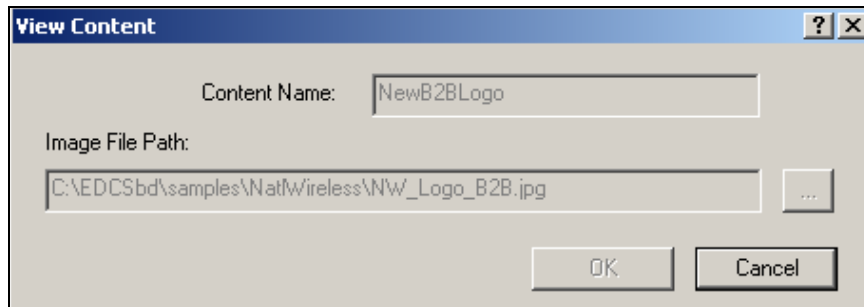
Changing Content

You can modify change the HTML template file, text, or image file associated with a content name by editing the content properties. (You can also modify table content and reusable table content.)

To edit content properties:

1. With the ALF open in Composer, double-click the 🗑 Content node to display a list of content names in the Application Tree. (Double-click the ⊞ 🗑 Table Content node to display a list of table content names, if necessary.)

2. Right-click the content name you want to edit.

3. Select **Edit Properties** from the right-click menu. Composer displays the Modify Content dialog (or the Modify Reusable Content dialog for reusable table content).

4. Edit the content as necessary and click **OK**.

Deleting Unused Content

You can delete any content that is not in use in a condition, including reusable table contents.

To delete unused content:

1. With the ALF open in Composer, right-click the 🗑 Content node in the Application Tree. (Click the ⊞ 🗑 Table Content node to view a list of table content only.)

2. Select **Content List** from the right-click menu. The Content List dialog appears.

3. Select the unused content item you want to delete and click **Delete**.

4. Click **Close**.

Mapping a Notification Link

You can map a notification link into an HTML email template. eaDirect adds a direct link to the production web server, based on the configuration settings in the eaDirect Command Center. In production, this link is embedded in the delivered email and provides the recipient with a convenient and customized link to their login page.

To map an email notification link into the email template:

1. With the ALF and email template open in Composer, click the **Definition** tab in the Application Tree area.

2. Click the Notification Link icon and drag and drop the link where you want it to appear in the template. (The email notification link appears in the template as **[E]NotificationLink,L[/E]**. The "L" indicates this is a link.)
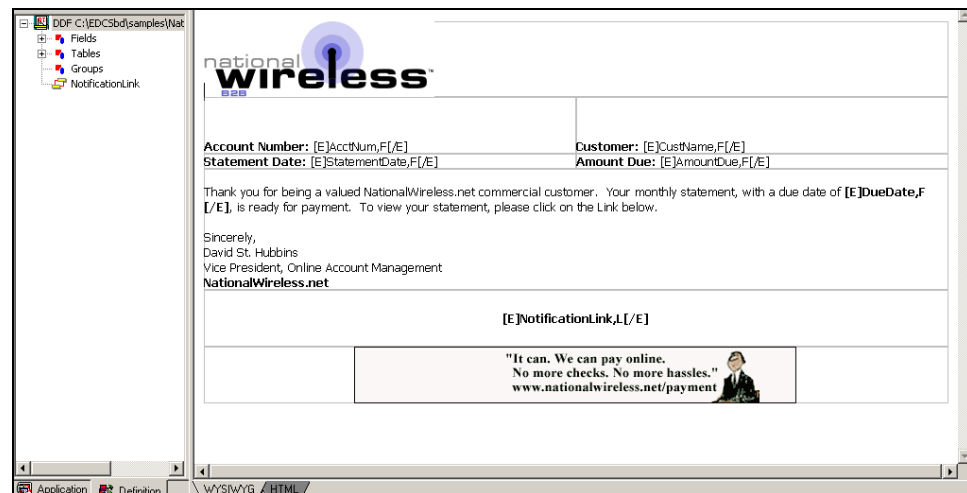
3. If the HTML template uses placeholders, delete them.

4. Click the **Save Template** icon.

Using Encoded Character Sets in the HTML Templates

To enable composed HTML pages to properly display the character set defined in the DDF, you must add the character set tag to the HTML template.

(If you want to use the default character set name specified in the code for presentment, do not complete the following procedure.)

To insert an [E] tag with the character-set name in the HTML templates:

1. On the right side of the Composer screen, click the **HTML** tab.

2. Select **Insert > Character set**. Composer inserts the character set name specified in the DDF into the default template and any conditional templates associated with the ALF as an [E] tag in the form charset=[E]Character-set[/E], for example:

   ```
   <META content="text/html; charset=[E]ISO-8859-1[/E]"
   http-equiv=Content-Type>
   ```

3. To enable the character set name you specified in the HTML template to override the default character set name in the code, your application designer must manually edit your Validation.jsp (included in Detail.jsp to support eaDirect user validation). They must remove the line shown here in bold:

   ```
   <%
       response.setContentType("text/html; charset=ISO-
   8859-1");
       validator.validateParameterNames(request);
       validator.validateParameterValues(request);
   %>
   ```

Mapping Text Tables and Configuring Elements

5

Mapping Tables and Groups

To present the groups and tables you defined in the DDF in your statement, you must map them to the location in the template where you want them to appear. Composer lets you click-and-drag tables and groups to the template.

You can present tables in one of two formats:

- As text in an HTML table
- As a chart that you design when mapping

A group consists of one or more tables that recur in a visual pattern. The simplest group can consist of one table of which there are multiple occurrences. Groups can be used also display information that repeats together. Each set of tables in a pattern is a group occurrence.

For details about mapping and formatting tables as charts, see Chapter 5.

Mapping a Table as Text

To map a table as text in a statement template:

1. With the ALF open in Composer, add an HTML table to the statement HTML template (if you haven't already); select **Table>Insert Menu**.

2. Click the Definition tab in Tree area and expand the list of DDF tables by clicking on the + sign next to "Tables."

3. Drag and drop the table you want to map to the location in the HTML table where you want it to appear.
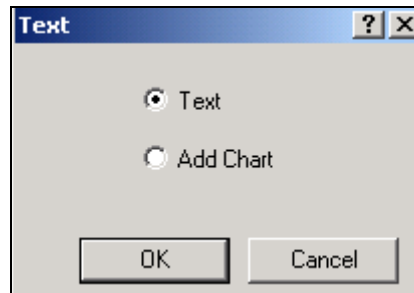
4. Select **Text** and click **OK.** "**[E]TableName, R[/E]**" appears in the template.

5. If your HTML template used placeholders, delete them.

6. Click to save the template.

Tip   If you delete a field, table, or group from the DDF, you must manually delete the associated tag from the HTML template.

Mapping a Group
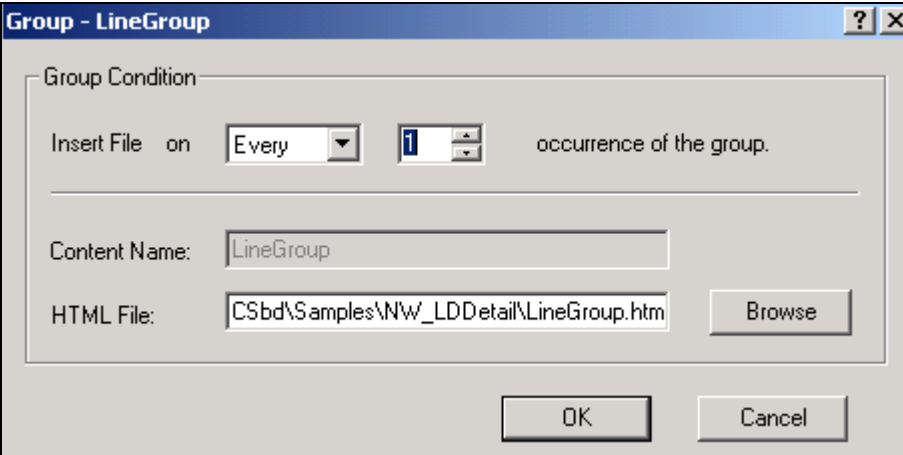
To map a group to a statement template, you drag-and-drop the group to the template, identify the group template to use for display, and specify the condition for using this template (every occurrence or some other basis).

Because you specify an external group template for presenting the group, you don't need to create an HTML table for the group; you don't need to encapsulate the group with any HTML tags. The HTML group template contains the **[E]** tags for the group's components, and the ALF contains the formatting rules for all tables and groups.

Correct	Incorrect
`[E]Group,G[/E]`	`<table width="100%"> <tr> <td> [E]Group,G[/E]</tr> </td> </table>`
`<table width="100%"> [E]Table,R[/E] </table>`	`[E]Table,R[/E]`

To map a group to a statement template:

1. Click the ![Definition] tab in the Tree area and double-click the ![Groups] node to expand the list of groups defined in the DDF.

2. Drag-and-drop the group you want to map to the HTML template. The Group dialog appears.

3. Select the "condition" for inserting the group template; in most cases you want to specify **Every** and **1** to insert the group template for each occurrence. (Specify a different frequency and number of times to insert the group template if required.)

4. Specify the group template or click **Browse** and select a template.

5. Click **OK**. The group HTML appears with a blue "content" icon in the Application Tree under the template where you mapped the group:
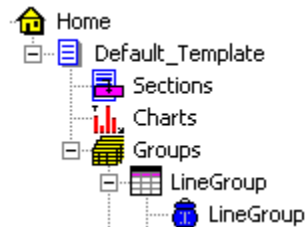
6. Click ⊞ to save the ALF.

Modifying the Group Template

You can change the group HTML template used to display a group in the statement or modify the "condition," or frequency to apply the group template.

To modify the group template or condition:

1. With the ALF open in Composer, expand the ▥ groups icons in the Application Tree to display the group you want to modify.

2. Right-click the ▦ icon for the group and select **Modify** from the right-click menu. Composer displays the Group dialog.

3. Make the necessary modifications.

4. Click **OK**.

Deleting a Group

To delete a group from a statement you should delete both the group tag and the group HTML association.

To delete a group from a statement:

1. With the ALF open in Composer, select the group tag in the statement and delete it.

2. In the Application Tree, right-click the group name and select `Delete` from the right-click menu. Composer asks if you're sure you want to delete the group.

3. Click `Yes`.

Formatting Tables

You can apply various formatting to enhance the appearance and readability of tables and to customize customer content.

You can apply formatting or add content to:

- **An entire table** – Use the Composer Format menu or Formatting toolbar to apply font and paragraph formatting such as styles, bolding, indentations, alignment, etc. eaDirect applies the formats to all columns and rows in the table. You apply table-wide formatting in the HTML template.

- **Individual columns** – In the temporary HTML displayed for configuring a table, use the Format menu or the Formatting toolbar to apply formatting to columns.

- **Customer-specific rows based on their data (using conditions)** – In the temporary HTML displayed for configuring a table, you create a section and specify the data-driven condition that (a column of) the customer's data must meet for eaDirect to apply the formatting or content. Conditional table formatting is useful, for example, to highlight a row containing a negative amount in red, add an advertising image, text, or link in a cell, or to remove a link to a detail drill-down view if a data amount equals zero.

- **A patterns of rows** – In the temporary HTML displayed for configuring a table, you create a section and define a pattern of lines and the table formatting (or other content, if useful) you want to apply to the pattern. For example, after every third row you could highlight two rows in blue for readability.

Formatting an Entire Table

To apply formatting to an entire table:

1. With the ALF open in Composer, select the table and use the Format menu or the Formatting toolbar to choose formatting. (If a table is part of a group, first click the group name shown under the ⊟·▦ Groups node in the Application Tree to list the tables in the group.)

2. Click 🖫 to save the formatting to the template (or wait and save the entire ALF at the end of your editing session, if you prefer).

Formatting Individual Columns

To format table columns:

1. With the ALF open in Composer, select the table you want to edit. Be sure to select the start and ending E tags with the table name. (If a table is part of a group, first click the group name shown under the ⊟·▦ Groups node in the Application Tree to list the tables in the group.)

2. Right click on the selection, and select **Configure Table** from the right-click menu. A new window opens with a temporary HTML page.

3. Select individual columns to format in either WYSIWYG or HTML mode. In WYSIWYG mode, use the Format menu or the Formatting toolbar to add formatting. In HTML mode, you can enter HTML formatting code directly. (You must add all structural formatting, such as modifications to td width, height, or vAlign in HTML mode.

4. Click 🖫 to save the ALF.

Creating Conditions on a Table (Line Item)

Conditions on a table let you tailor advertising or marketing messages based on the data in the individual lines of a customer's statement, or format the line based on a key data value.

You can create data-driven conditions on a table or selected columns instructing eaDirect when to apply table content to that selection. eaDirect checks the condition for each line of data and, if true, applies the content to that line only. Table content can include any combination of font and cell formatting (color, bold, italics, alignment, etc.), or an image, text, or hyperlink that eaDirect adds to the section when composing the statement. You can optionally specify different content for each column in a condition. The data driving a table condition must be within the same table.

For example, you could create a condition on a table to:

- Highlight a row containing a negative amount in red (content can include or be limited to font or cell formatting).

- Add a targeted advertising image to a line item in a credit card statement based on the nature of the purchasing data, such as the name of the clothing retailer, type of clothing purchased, or the purchase amount. (You can optionally *replace* the section with an image using reusable table content in the condition.)

- Add a hyperlink, such as one offering a discount purchase coupon for the linked retail Web site.

- Remove a link to a detail drill-down view if a data amount equals zero

- Add a text message with targeted offers or information to the customer based on the transaction. (You can optionally *replace* the section with text using reusable table content in the condition.)

- Replace the section with a null. (You must use reusable table content in the condition to replace a section with a null.)

- Suppress section display. (You must use reusable table content in the condition to suppress display.)

When creating a condition on a table, you must define the content specifically for that condition, called "table content." You cannot use content created for non-table sections in a table condition and vice-versa. You can, however, create reusable table content and apply it to multiple conditions. You must create and apply reusable table content in the condition if you want to insert a null or suppress section display; regular table content cannot do this. Also, if you want to replace a section with an image or text instead of adding it before or after existing section content, you must create and apply it as reusable table content in the condition. See "Creating Table Content" on Page 87 for details.

Each condition on a table requires one or more logic elements. A logic element is an expression that, in a table condition, can compare:

- The values in two columns

- The value in a table column to a specific value such as a date, time, amount, NULL or EMPTY, or a regular expression

- A table or group with NULL or EMPTY (to test for existence)

You can create logic elements while defining a condition or create logic elements first, making them available to select from a drop-down list when creating the condition. See "Creating Conditions" on Page 40 for details on condition syntax and creating logic elements.

To create a condition on a table:

1. With the ALF open in Composer, select the table on which you want to create a condition. (If a table is part of a group, first click the group name shown under the ⊟📊 Groups node in the Application Tree to expose the tables in the group.)

2. Right click and select **Configure Table** from the right-click menu.

3. Select the section you want the condition to apply to (one or more, or all columns), then right-click and select **Add Section** and **Conditional Table Format** from the right-click menu. The Create Section dialog appears.

4. Enter a name for the section. Names cannot contain spaces or special characters, and must start with an alpha character. Maximum length is 30 characters. Use meaningful names; "BigSpenderInsertAirlineOffer" is more descriptive and easier to identify than "ALFsection1".

5. Use the options on this dialog to build a conditional statement in the **Build Condition** field. You can manually enter the text in the Build Condition field, but choosing from the menus reduces typing and syntax errors. (You can optionally choose a previously saved condition from the drop-down list if you've already created a condition you want to reuse in this section. If so, skip to Step 13.)

```
IF (
THEN
ELSE
)
(
```
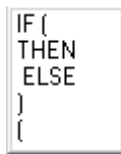
6. Click **Condition** ⎿⎾ and select **IF** from the drop-down list.

7. Under Select Logic Element, click **Create** to create a logic element for the condition; see Creating a Logic Element on Page 41 for details. If you have already created the logic element, choose it from the Name drop-down list and click **Add** to add it to the condition.

8. Click **Operators** to select any operators your conditional statement requires

```
AND
OR
NOT
```
.
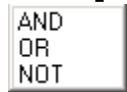
9. Add any additional logic elements the conditional statement requires.

10. Click **Condition** again and select the right parenthesis from the drop-down list. One set of parenthesis must enclose all logic elements in a condition.

11. Click **Condition** and select **THEN**.

12. Do one of the following:

   • To create content for this table, click **Create**. (See the instructions for creating table content in "Creating Table Content" on Page 87.)

   • To use an existing reusable table content, select it from the Name drop-down list under Select Content.

   • To create new, reusable table content, click **Reusable**. (See the instructions for creating reusable table content in "Creating Table Content" on Page 87.)

13. Click **Add** to insert the content name in the condition.

14. When the condition is complete, click **Check** to verify it is valid. Correct or complete the statement as necessary, checking the condition again.

15. If you want this condition to be available in eaMarket, click the **eaMarket enabled** checkbox. (If you eaMarket enabled a section, it appears with the eaMarket enabled section icon.)

16. When the condition is complete and valid, click **Add to List**. Composer displays the condition in the Condition List at the bottom of the dialog.

17. If you want to save the condition for possible reuse in another section, click the condition in the Condition List and type a name in the Condition Name drop-down box (next to the Add to List button). Click **Save**.

18. If necessary, build any additional conditions to create a compound condition on the section. Begin subsequent conditions with ELSE IF.

19. Click **OK**. The condition appears in the node for the template. Composer also adds a number to the table tag in the template, as in [E]LineCallList_0,R[/E], indicating a condition is defined on the table.

Creating Table Content

When creating a condition on a table, you must define new table content for use with the condition. You cannot use content created for non-table sections of a template in a table condition. In addition, the table content you create does not automatically become available for use in other table conditions. Composer ties table content directly to a particular table condition. You can create reusable table content, however, to define certain types of content, and these are available for reuse in other table conditions.

Table content can specify:

- Font formatting - Bolding, italics, underlining, and text color.

- Cell formatting – Alignment (left, right, center) and cell background color.

- Text and an optional hyperlink on the text.

- An image and an optional hyperlink on the image.

- Hyperlinks

In addition, you can specify different content and format for individual columns in the condition (and these specifications are kept together as table content for the condition).

You must use reusable table content to:

- Suppress display

- Insert a null

- Replace a section with text (use regular table content to *add* text)

- Replace a section with an image (use regular table content to *add* an image)

To create table content:

1. You create table content by clicking the Content button on the Create Section dialog when creating a condition on a line item or on a pattern of rows. (eaDirect displays the Create Content dialog.) You cannot create table content independently of a table condition, except to create reusable content (see below). On the Create Content dialog, enter a name for the content in the Content Name field.
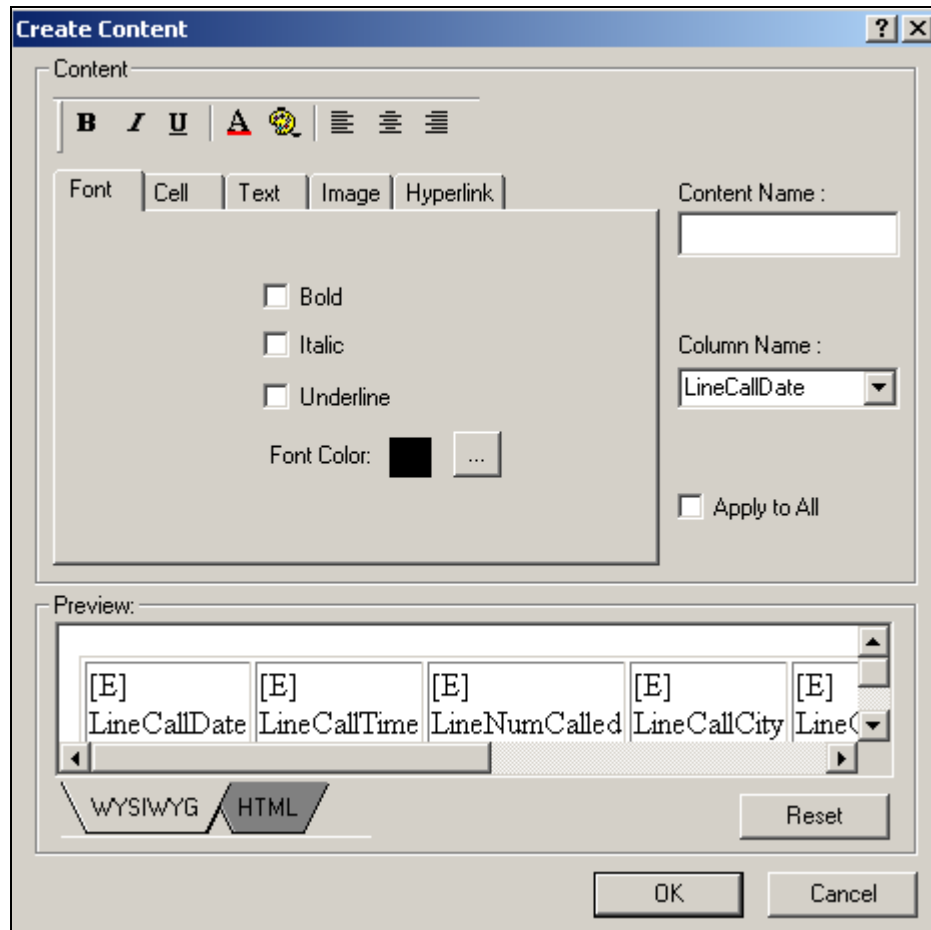
2. If you want the table content you specify to apply to all columns selected for the condition, click the Apply to All option. If you want to apply different content to individual columns, select the column names one at a time and proceed to specify the content.

3. Specify content use the tab options and/or the formatting toolbar(you can optionally click the HTML tab to edit the HTML directly):

   • **Font formatting** – For bold, italics, underlining, and font color, use the Font tab and/or the **B** *I* <u>U</u> **A** toolbar buttons.

- **Cell formatting** – To align left, center, or right, and apply background color to cells, use the Cell tab and/or the ▣ toolbar buttons.

- **Text** – Use the Text tab to define a block of text to add and optionally specify a hyperlink to use on the text. You must specify whether you want eaDirect to insert the text before or after the existing section content; select either the **Prefix** or **Suffix** option on the Text tab.

- **Image** – Use the Image tab to specify the path of an image file to insert and optionally specify a hyperlink to use on the image. You must specify whether you want eaDirect to insert the image before or after the existing section content; select either the **Prefix** or **Suffix** option on the Image tab.

- **Hyperlink** – Use the Hyperlink tab to specify an address to add as a link.

4. If you specify content in the Text, Image, or Hyperlink tabs, click **Apply** on that tab.

5. Use the Reset button if you want to clear any formatting you've selected.

6. Click **OK**. The name of the table content appears with a blue icon under the Table Content node in the Application Tree.

To create reusable table content:

You can create reusable table content while creating a condition on a table or at any time.

1. If you clicked the Reusable button on the Create Section dialog when creating a condition on a table (including on a pattern of rows), eaDirect displayed the Create Reusable Content dialog.

   If not, with the ALF open in Composer, right-click the 🗑 Table Content node in the Application Tree and select **Create Reusable**. Composer displays the Create Reusable Content dialog.

2. Specify a name for the reusable content.

3. Click the type of content you want to define: Suppress (to suppress display), Insert Null, Replace Text, or Replace Image.

4. For a null, text, or image, specify the number of cells.

5. To add a hyperlink to text or an image, click the **Hyperlink** option and specify a link.

6. Click **OK**. The name of the reusable content appears with a yellow icon under the Table Content node in the Application Tree:
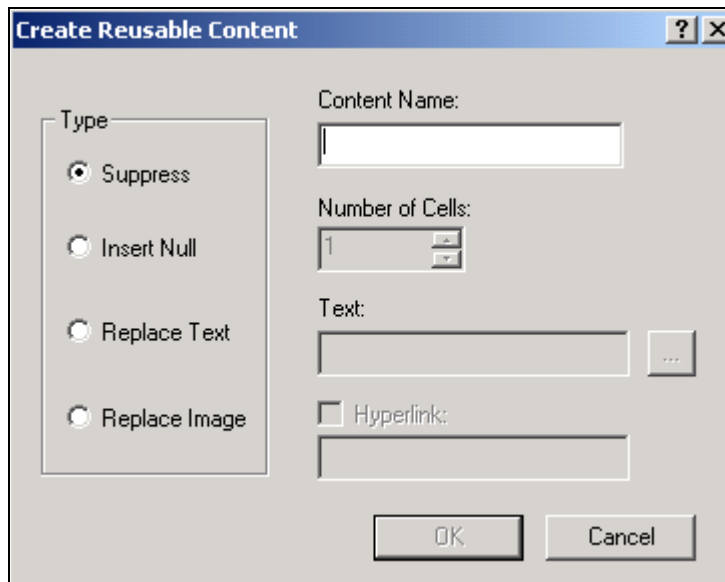
To view a list of table and reusable content:

- See "Viewing Content" on Page 72.

To change table content:

- See "Changing Content" on Page 74.

To delete (unused) reusable table content:

- See "Deleting Unused Content" on Page 74.

Deleting, Changing, or Viewing Table Conditions

To view the properties of a table condition:

1. With the ALF open in Composer, double-click the ▦ Sections node (below the 🏠 Home node). Composer displays the list of conditions (sections).

2. Double-click the name of the condition (section) you want to view.

3. Right-click on the IF= Condition node and select **View Properties** from the right-click menu. Composer displays the Section Condition dialog with the conditional statement.
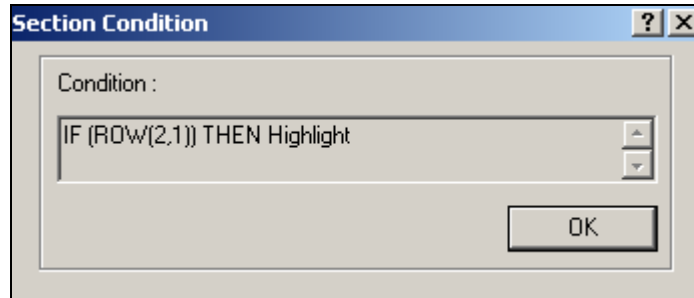
4. Click **OK**.

To modify a condition on a table:

You can modify a table condition or add one or more ELSE IF conditions to a section (creating compound conditions). You can also modify the condition on a pattern of rows.

1. With the ALF open in Composer, double-click the ▤ Sections node (below the 🏠 Home node) for the table.

2. Right-click the name of the condition you want to modify.

3. Select **Modify Section** from the right-click menu. The Modify Section dialog appears.

4. For a regular table condition, double-click on the condition and edit as needed. To add a new condition to the table section, follow the procedures above to add an ELSE IF condition. For a condition on a pattern of rows, edit the condition as needed.

5. Click **OK**.

To delete a table condition:

1. With the ALF open in Composer, double-click the ▤ Sections node (below the 🏠 Home node) for the table.

2. Right-click the name of the condition you want to delete.

3. Select **Delete Section** from the right-click menu.

4. Composer asks if you're sure you want to delete the section. Click **Yes**. Composer updates the table tag on the template (removes "_0," for example).

Applying Formatting or Adding Content to a Pattern of Rows

You can apply table content to a defined pattern of rows, letting you apply color highlight to alternating rows in a statement, for example. Color or other formatting can often make the lines of a statement easier to read.

You can apply the same types of table formatting and content to a row pattern as you can to regular table conditions on a line item. The "condition" in this case is simply the pattern of rows you define, and is not based on extracted data. For details on the types of formatting and content you can add to a pattern of rows, see "Creating Table Content" on Page 87.

You specify a pattern by the starting row number and the number of rows to repeat. For example, if you specify the row number to start at as 3 to repeat for 2 rows, and define the content as font color highlighting in yellow, then eaDirect composes every other two rows in yellow:

Row 1: Default color
Row 2: Default color
Row 3: Yellow
Row 4: Yellow
Row 5: Default color
Row 6: Default color

To apply a somewhat more complex highlighting, you can define multiple patterns for a table. For example, you would need to create two separate sections to achieve the following color highlight pattern:

Row 1: Yellow
Row 2: Green
Row 3: Blue
Row 4: Yellow
Row 5: Green
Row 6: Blue

In this example, you apply yellow as the default table color in the default template. Then you would create one section on the table with a pattern that starts on row 2 and repeats for 1 row in green, and another section on the table with a pattern that starts on row 3 and repeats for 1 row in blue.

To format a pattern of rows in a table:

1. With the ALF open in Composer, select the table you want to edit. (If a table is part of a group, first click the group name shown under the ⊟ 🗄 Groups node in the Application Tree to expose the tables in the group.)

2. Right click on the selection, and select `Configure Table` from the right-click menu. A new window opens with a temporary HTML page.

3. Select the table you want to format, and right click to select `Format Table` from the right-click menu. The Create Section dialog appears.
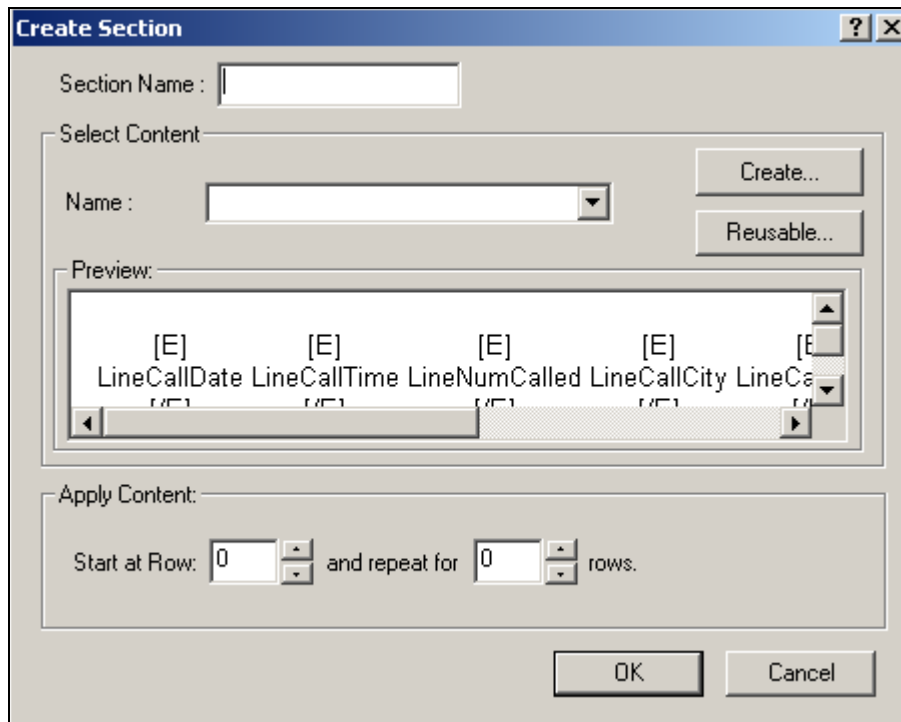
4. Enter a name for the section, such as HighlightEveryOtherRow.

5. Do one of the following:

    •   To create content for this table, click **Create**. (See the instructions for creating table content in "Creating Table Content" on Page 87.)

    •   To use an existing reusable table content, select it from the Name drop-down list under Select Content.

    •   To create new, reusable table content, click **Reusable**. (See the instructions for creating reusable table content in "Creating Table Content."

6. Specify the row number where you want the pattern to begin.

7. Specify the number of rows you want to apply the table content to.

8. Click **OK**. The condition appears in the node for the template. Composer also adds a number to the table tag in the template, as in [E]LineCallList_0,R[/E], indicating a condition is defined on the table.

To delete, modify, or view a condition on a pattern of rows:

- See "Deleting, Changing, or Viewing Table Conditions" on Page 92.

Sorting Tables and Groups

You can enable Web users to sort the data in their statements by creating sort elements and adding a link to the element in an appropriate place in the statement. You can sort on a table or a group.

You can create multiple sort elements within a statement ALF; you can also use the same link for multiple sort (or filter) elements.

The Simulator lets you validate the format of sort elements; see "Simulating Sorting and Filtering Elements" on Page 143.

Table Sorting

You can enable Web users to sort rows in a table in ascending or descending order by the values in a particular column. To do this, you create a sort element in table mode identifying the table and column to sort on, specify whether to sort in ascending or descending order, then create a sort element link and add it to the template.

When a Web user clicks a sort element link, the statement page redisplays with the table rows reordered according to the specifications of the sort element. When the user clicks the sort element again, sorting order automatically reverses.

Example - View of unsorted table (rows of a credit card statement ordered by Posting Date):

Posting Date	Transaction Date	Reference Number	Transactions	Charges
5/20	5/19	10384	Illuminations Lighting	166.79
5/23	5/22	10298	Springfield Getty	18.50
5/25	5/24	10273	American Airlines	1680.00
5/30	5/29	00578	Ocean Grill	65.00
6/06	6/04	20182	Harbor Line Cruises	126.06

Above table sorted by "Charge" column:

Posting Date	Transaction Date	Reference Number	Transactions	Charges
5/23	5/22	10298	Springfield Getty	18.50
5/30	5/29	00578	Ocean Grill	65.00
6/06	6/04	20182	Harbor Line Cruises	126.06
5/20	5/19	10384	Illuminations Lighting	166.79
5/25	5/24	10273	American Airlines	1680.00

If a table is part of a group, eaDirect sorts within each occurrence of the table. To enable sorting across occurrences of a table or group, create a combine element on the table or group in addition to the sort element. Creating both a sort and combine element on a table or group sorts and displays the rows from all table or group occurrences in sequence according to the sort column value. To create a combine element, see "Using Combine Elements to Page, Sort, & Filter Across Table or Group Occurrences" on Page 129.

Group Sorting

You can enable Web users to sort occurrences of group data in ascending or descending order. To sort a group, you create a sort element in group mode identifying the table and column (or the group field), specify whether to sort in ascending or descending order, then create a sort element link and add it to the template. When a Web user clicks the link, the statement page redisplays with the group occurrences reordered according to the specifications of the sort element. When the user clicks the link again, sorting order automatically reverses.

eaDirect sorts group occurrences according to the sort column value in the **first row of the first occurrence of the table** in the group. The first occurrence of the table on which you define the sort element acts as the "lead table," and the first row of this table occurrence as a "lead line" for sorting the group.

This can be useful, for example, when the tables in a group contain only one row, such as the lines of an address, or wherever it makes sense to sort on just one column and the first row of a table.

Example - View of unsorted group (tables of calling card line item data):

				Duration	Charge
$G1_{(0)}$	$T1_{(0)}$	10/21/02	12:35 pm	0:24:18	10.00
	$\mathbf{T2_{(0)}}$	Rye, NY	Atlanta, GA		
$\mathbf{G1_{(1)}}$	$\mathbf{T1_{(0)}}$	10/22/02	4:08 pm	0:05:46	1.60
	$\mathbf{T2_{(0)}}$	Atlanta, GA:	Chicago, IL		
$\mathbf{G1_{(2)}}$	$\mathbf{T1_{(0)}}$	10/24/02	6:00 am	1:59:04	34.90
	$\mathbf{T2_{(0)}}$	Chicago, IL	Rye, NY		

Above group sorted by call charge column:

				Duration	**Charge**
$G1_{(1)}$	$T1_{(0)}$	10/22/02	4:08 pm	0:05:46	1.60
	$T2_{(0)}$	Atlanta, GA:	Chicago, IL		
$G1_{(0)}$	$T1_{(0)}$	10/21/02	12:35 pm	0:24:18	10.00
	$T2_{(0)}$	Rye, NY	Atlanta, GA		
$G1_{(2)}$	$T1_{(0)}$	10/24/02	6:00 am	1:59:04	34.90
	$T2_{(0)}$	Chicago, IL	Rye, NY		

Creating a Sort Element

To create a sort element:

1. With the ALF open in Composer, right-click Sort Elements in the Application Tree. (Open NW_LocSummary.alf.)

2. Select `Create Sort Element` from the right-click menu. The Configure Sort dialog appears.
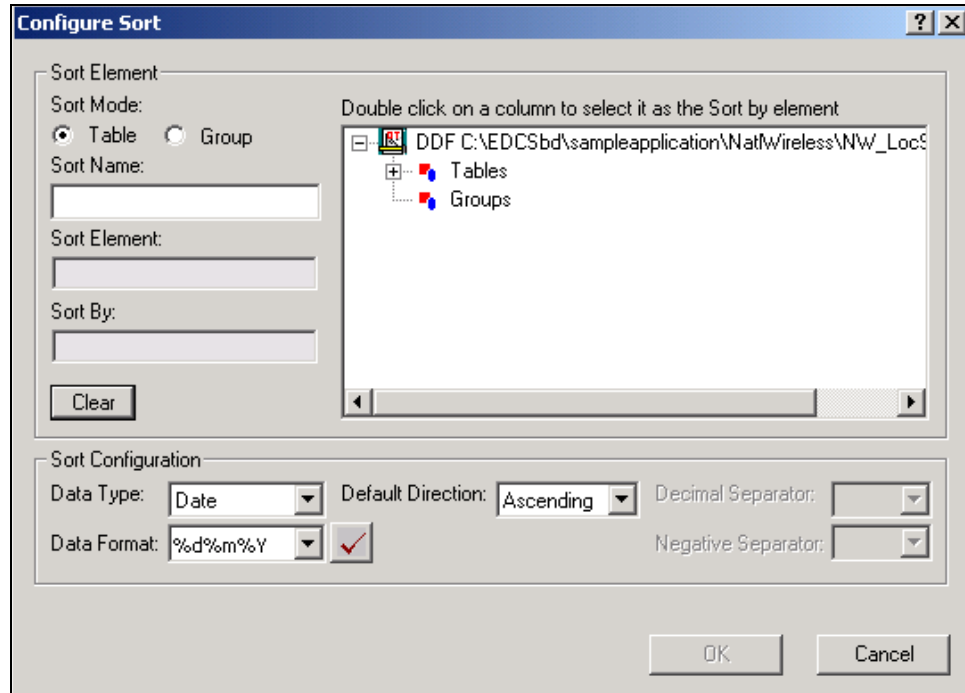
3. Under Sort Mode, click **Table** to create a sort on a table, or **Group** to create a sort on the occurrences of a group.

4. Enter a descriptive name for the sort. The name cannot contain spaces or special characters, must start with an alpha character, and cannot exceed 30 characters in length. (Enter **SortLocChargeDesc**.)

5. On the tree of table and groups, expand the list and select the table or group you want to sort. (Expand the **LocalChargeSummary** table.)

6. Double-click the table and column you want to sort by. If sorting a group, you can select a group field. If you make a mistake, click **Clear** and reselect the table and column. (Select **LocalChargeDesc**.)

7. Select the data type of the column you're sorting. (Select **String**.)

8. If the data type is date or time, select the appropriate sort format from the **Data Format** drop-down list or type one of your own. Manually enter any separators within the data format, if necessary. (You can copy, cut, and paste selected values from the list of formats to build the one you want. Right click the format to display the Edit menu.)

9. In the Default Direction field, choose the direction to sort: ascending or descending.

10. If the data type is currency or decimal, specify a separator (decimal point or comma) in the Decimal Separator field.

11. If the data type is currency, you can optionally select a symbol to represent a negative value: -, CR, cr, DB, db, or ().

12. If the data type is a date or time, click the Validate button to verify that the format you've entered complies with the data type.

13. Click **OK** to create the sort element.

14. Click to save the ALF.

15. Follow the instructions in the next section to create a link to the sort element.

Creating a Link to a Sort Element

After creating a sort element, you must create a link users can click to perform the sort. You can create a link on static text, an image, or an **[E]** tag.

You can optionally choose to add a Reset URL function to the link. By default, the sort would be cumulative on any previous URLs the user chose in the session. Adding this function enables you to clear any previous URL actions the Web user took, such as filtering data, before applying the sort (except if you sort the same table or group using a different column from a previous sort, eaDirect removes the previous sort and applies the new one).

To create a link to a sort element:

1. With the ALF open, select the item you want to use as a link (be sure to completely highlight the item). (Highlight the word "Description" above the **LocalChargeSummary** table.)

2. Right-click and select **Create Link** from the right-click menu. The Create Link dialog appears.
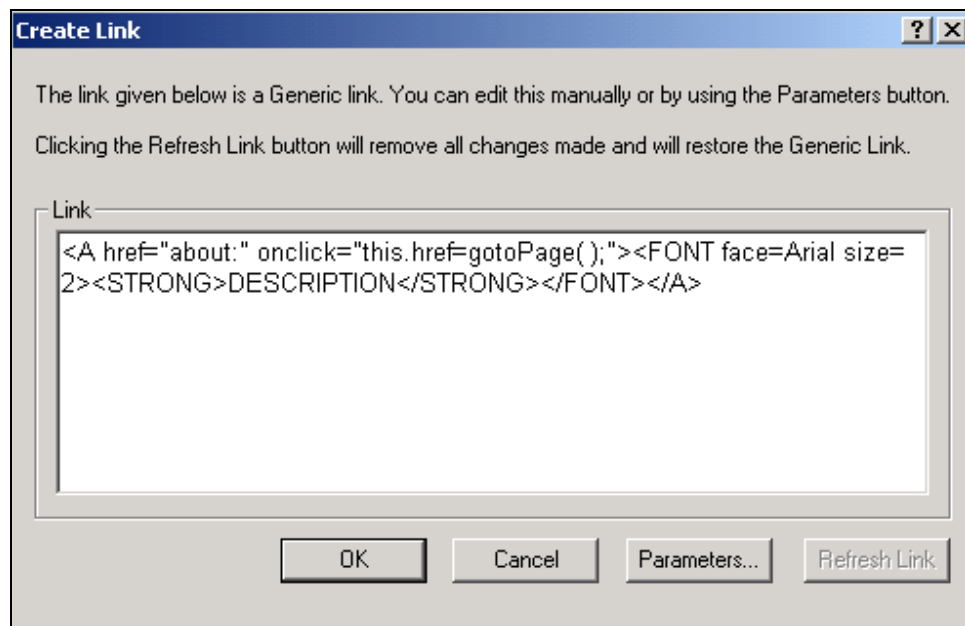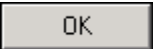
3. Click **Parameters**. The Add Parameters dialog appears.
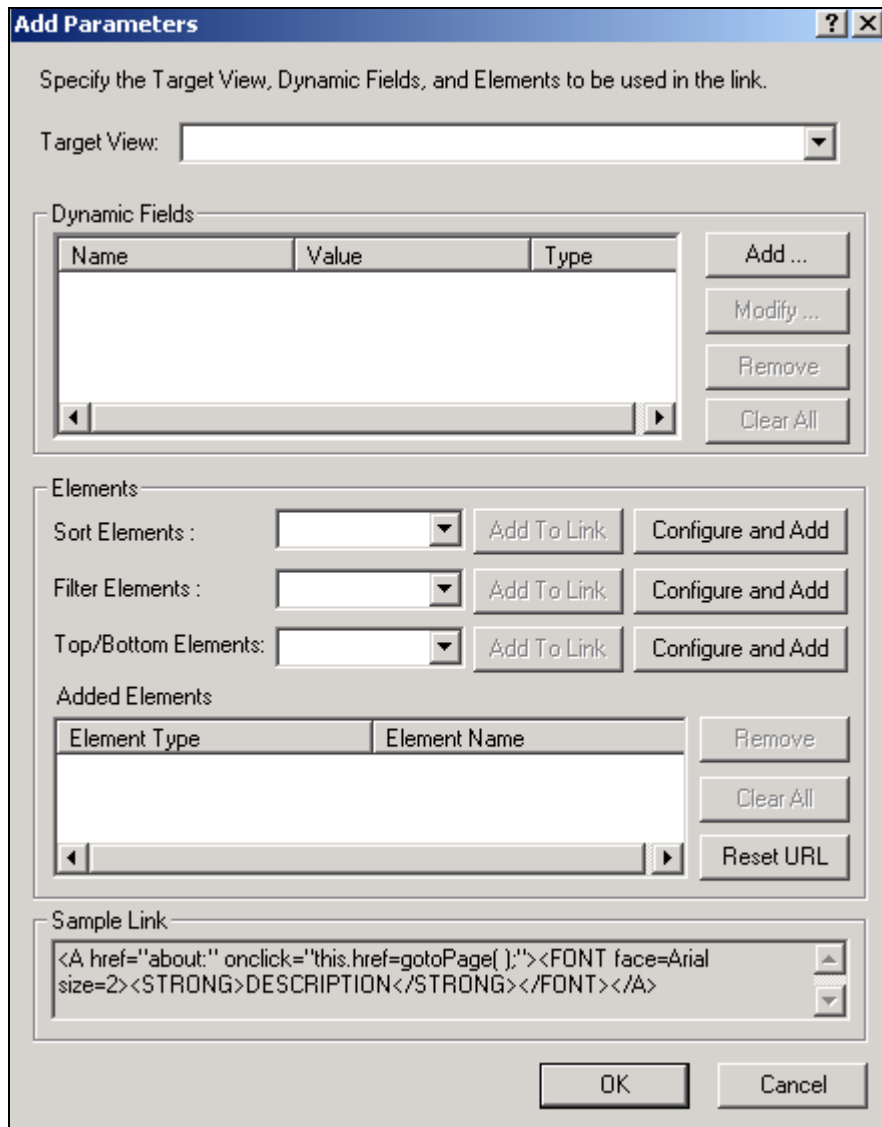
4. If the link is to a sort that appears in a different view, enter the name of the target view. (For NatlWireless, leave this blank.)

5. Click the **Sort Elements** drop-down list and select the sort element you want to link. (Select **SortLocChargeDesc.**)

6. Click [Add To Link]. The sort element appears in the Added Elements area.
   (You can use the Remove and Clear All buttons on items in this list.)

7. If you want to clear the actions of any previous URLs before displaying this
   sort, click [Reset URL] to add the Reset URL function to the link.
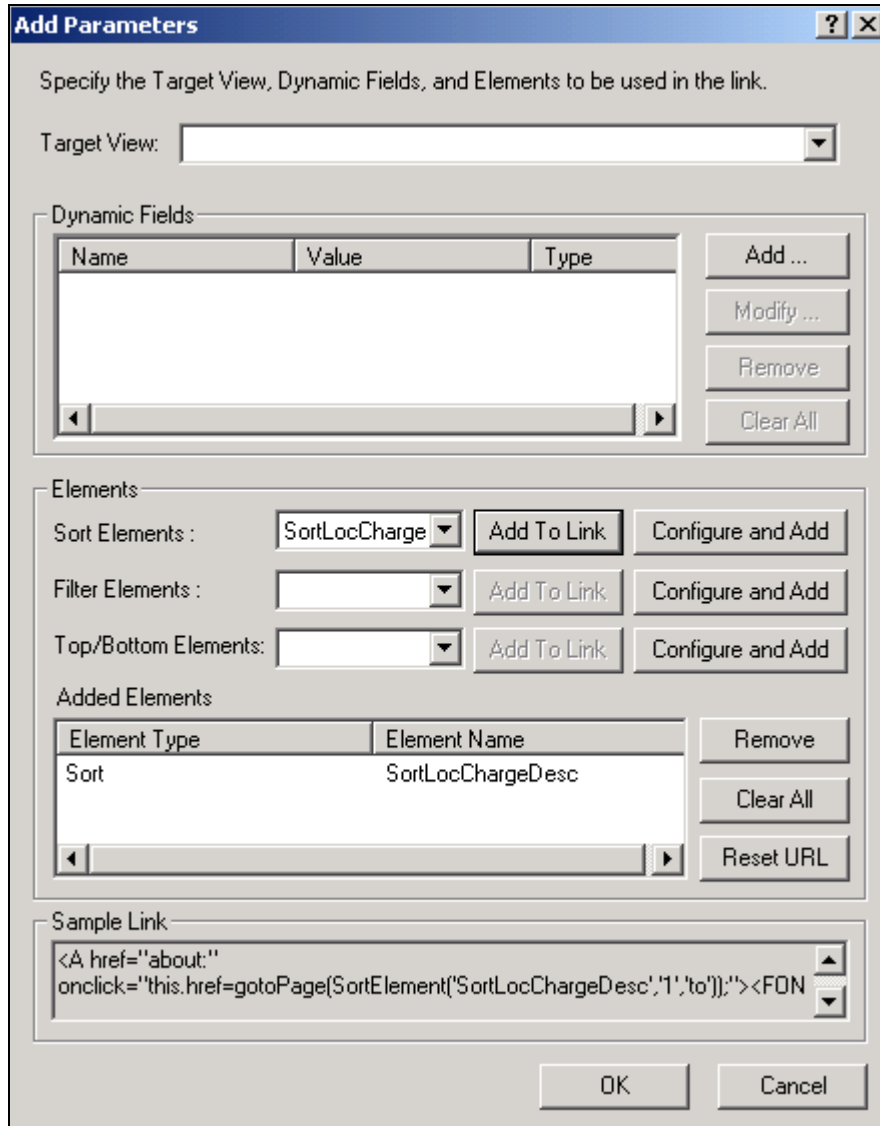
8. Click [OK]. Composer displays the view name and sort element in
    the Create Link dialog.
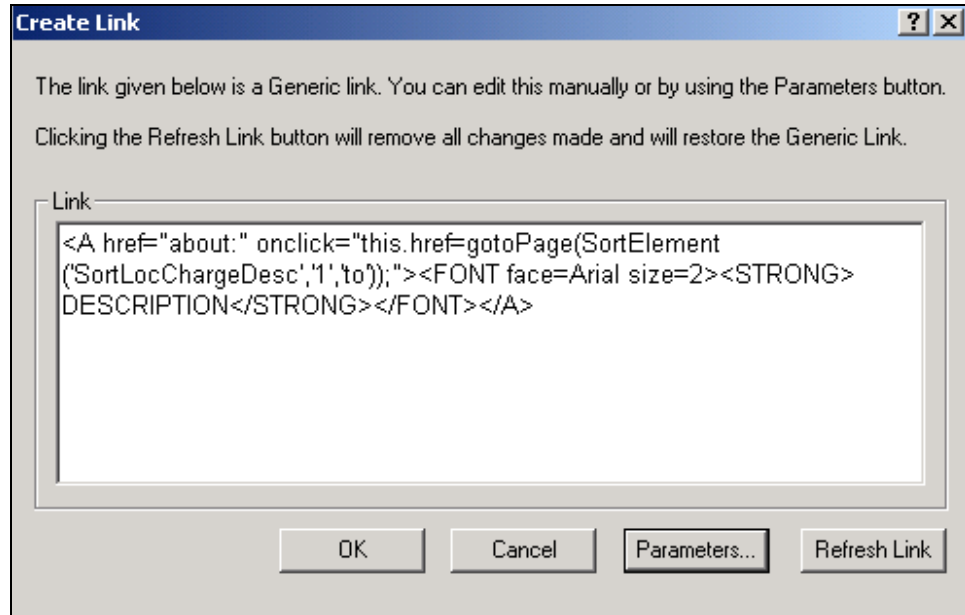
9. Click ![OK] to create the link. Composer adds a blue hyperlink to the statement.

10. Click ![save] to save the template.

11. If you added a link to a table column, click ![save] to save the ALF.

Tip   You can create a sort or filter element from the Add Parameters dialog; click ![Configure and Add]. See "Sorting Tables" on Page 97 or "Filtering Data" on Page 113 for details.

Modifying a Sort Element

To modify a sort element:

1. With the ALF open in Composer, right-click the sort element and select **Modify Sort Element** from the right-click menu. Composer displays the Configure Sort dialog.

2. Edit the sort element.

3. Click **OK**.

Deleting a Sort Element

To delete a sort element:

1. With the ALF open in Composer, right-click the sort element and select **Delete Sort Element** from the right-click menu. Composer asks if you're sure you want to remove this sort element.

2. Click **Yes**. Composer removes the sort element from the ALF.

3. You must manually remove the link to the sort element in the HTML template.

Paging Statements

Creating a page element lets you present large documents in multiple browser pages each with a fixed (configurable) number of rows. You can configure paging on a group or a table. You can also define page sets to display a fixed (configurable) number of pages in each set.

Using a paging element and page set can speed up the display of statement data to the user. Paging uses progressive data extraction so the user does not have to wait for all data to extract before starting to view it.

Once you create a page element and any page sets you must add one or more links to the statement for navigating between pages and page sets.

By default, eaDirect starts a new page for each occurrence of a table or group, regardless of how many rows there are. To eliminate the new page start and display table or group information contiguously, define a combine element for the table or group. Web users can then view the same amount of information in fewer (full) pages. See "Using Combine Elements to Page, Sort, & Filter Across Table or Group Occurrences" on Page 129.

Note that any lines containing group fields appear in addition to the number of lines, or rows, you specify for paging.

Also note that you cannot simulate paging in Composer; a simulation returns the first page of a document only. Also, paging links call a JavaScript function that is only available in a fully installed live environment.

Creating a Page Element

To create a page element:

1. With the ALF open in Composer, select the table or group to use for the paging element. Be sure to select the entire tag.

2. Right click on the tag and select `Paging, then Configure Paging Element from the` right-click `menus.` The Configure Paging dialog appears.
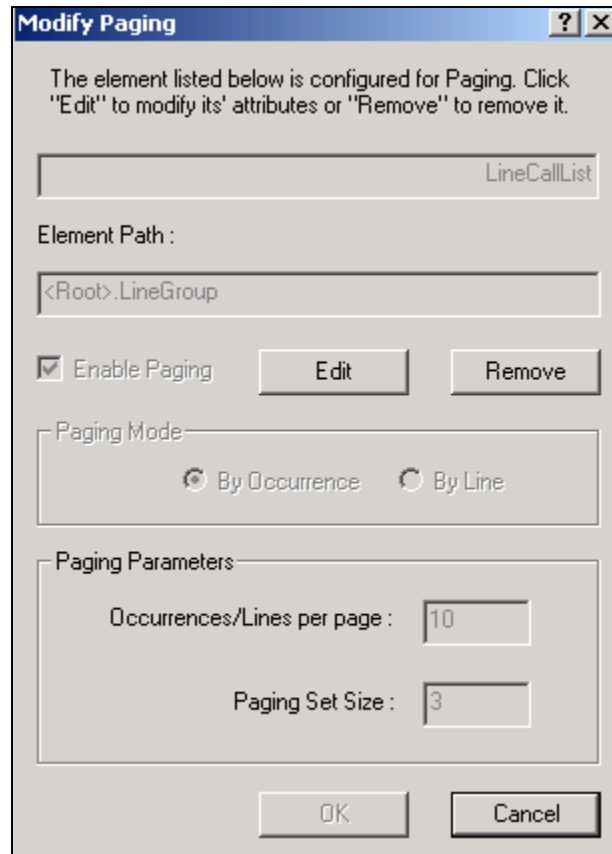
3. Select a paging mode, either **By Occurrence** (for a group) or **By Line** (for a table or a group). Paging groups by line counts all the contents of the group by the number of lines (or rows) returned. Paging groups by occurrence displays the entire group, regardless of the number of lines returned.

4. Enter the number of occurrences (for a group) or lines that will be shown on each HTML browser page (for tables and groups). Note that any lines containing group fields are not included in this count, but appear in addition to the number of rows you specify here. You may want to take this into account when planning your page size.

5. Enter the number of pages to extract at a time (page set) in **Paging Set Size**.

6. Click **OK**.

Creating a Page Set

To create a page set:

You must create a paging element before you can create a page set.

1. Enter text in the template to serve as a placeholder for the paging set tag.

2. Select the text.

3. Right-click and select **Paging and Insert Paging Set** from the right-click menu. Composer inserts the paging set tag.

4. Delete the placeholder text and "%P" from the template. The remaining tag should look something like: `[E]PageSet[/E]`.

Inserting Navigational Paging Links

You can create navigation links in the template to move to a previous or next extracted page, to extract the previous or next page set, and to go to the first page.

To insert page navigational links:

1. Insert text (such as "Next") or a graphic to use as a linking object.

2. Select the object.

3. Right-click and select **Paging and Paging Links** from the right-click menu. The Select Paging Link Type dialog appears.

4. Select the appropriate radio button for the paging link type.

5. Create any additional links.

6. Save your work.

Modifying a Page Element

To modify a page element:

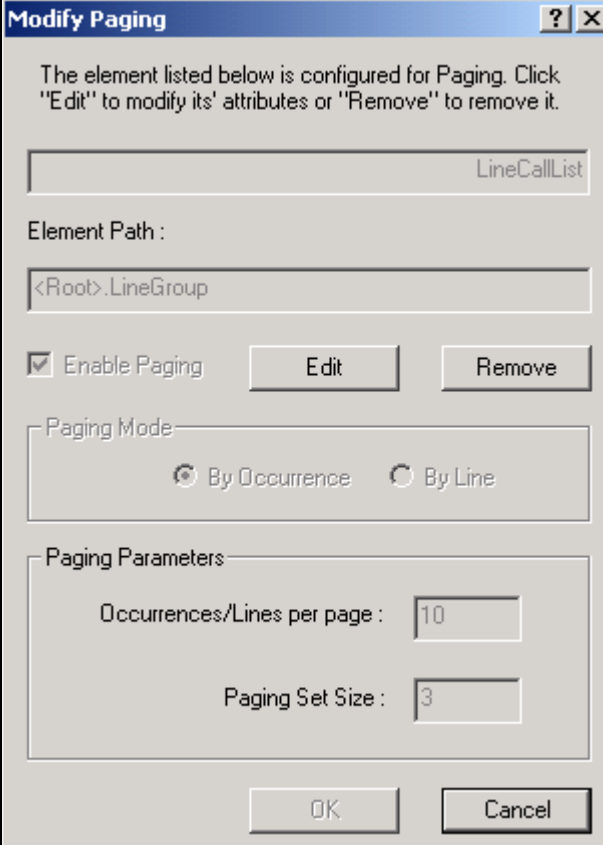1. With the ALF open in Composer, double-click the Page Element node in the Application Tree. Right-click on the page element and select **Modify Page Element** from the right-click menu. eaDirect displays the Modify Paging dialog.

2. Click **Edit** and modify the page element.

3. Click **OK**.

Removing a Page Element

To remove a page element:

1. Either:

   - Right-click on the page element in the Application Tree and select `Modify Page Element`. Click `Remove` on the Modify Paging dialog.

   - Right-click anywhere in the template and select `Paging and Remove Paging Element` from the right-click menu.

2. Click `OK` when prompted to remove the existing page element.

Filtering Data

You can let Web users view a narrow extraction of data based on filtering criteria you define in a filter element. You must create a filter on a particular table column then add a link to the statement template, which users click to compose the filter. You can filter table or group data.

You can create two types of filter elements:

- General content-based – To apply a filter to extracted data, producing a narrowed view of the data set. You could create a filter to display only those line items past 60 days due, over $1,000, etc.

- Top/bottom – To create a counted subset of filtered data based on a configurable number of items, from either the top or bottom of the filtered data set. For example, you could create a top/bottom filter element to display the top 10 most expensive phone calls in a telecomm statement.

For each filter element (except top/bottom) you must define an expression that compares the value in a particular column to either a hard value (date, integer, time, currency amount, decimal value, or string), another table column, or a field. For example, a simple filter expression for string data consists of a column name followed by an operator (==, <, >, <=, >=, etc.) and the comparator, for example:

```
AreaCode == "MA"
```

You can create a compound filter expression using AND, OR, and NOT operators. For details and examples of data filtering expressions, see Appendix A.

The Simulator lets you validate the format of sort elements; see "Simulating Sorting and Filtering Elements" on Page 143.

Table Filtering

You can enable Web users to view a filtered subset of table rows using either a general content-based filter or a top/bottom counted subset filter. To filter a table, you must create a filter element (in table mode) that defines the table and column to filter along with filter extraction criteria, then create a filter element link and add it to the statement template.

When a Web user clicks the link, the statement page redisplays with only the table rows that satisfy the specifications of the filter element expression.

You can create a filter on an individual table or on each occurrence of a table in a group (or across group occurrences). To enable filtering across occurrences of a table, create a combine element on the table in addition to the filter element. To create a combine element, see "Using Combine Elements to Page, Sort, & Filter Across Table or Group Occurrences" on Page 129.

Group Filtering

You can enable Web users to view only the **occurrences** of a group that satisfy a general content-based filter or a top/bottom counted subset filter. To filter a group, you create a filter element (in group mode) identifying the group to filter, along with the filter extraction criteria, then create a filter element link and add it to the template. When a Web user clicks the link, the statement page redisplays with only the group occurrences that meet the specifications of the filter element expression.

eaDirect filters group occurrences based on the filter column value in the **first row of the first occurrence of the table** in the group. The first occurrence of the table on which you define the filter element acts as a "lead table," and the first row of this table occurrence acts as a "lead line" for filtering the group.

To enable filtering across occurrences of a table and/or group, create a combine element on the table and/or group in addition to the filter element. To create a combine element, see "Using Combine Elements to Page, Sort, & Filter Across Table or Group Occurrences" on Page 129.

Creating a General Filter Element

To create a general filter element:

1. With the ALF open in Composer, right-click the ![Filter Elements icon] Filter Elements node in the Application Tree. (Open NW_LocSummary.alf.)

2. Select **Create Filter Element** from the right-click menu. The Configure Filter dialog appears.
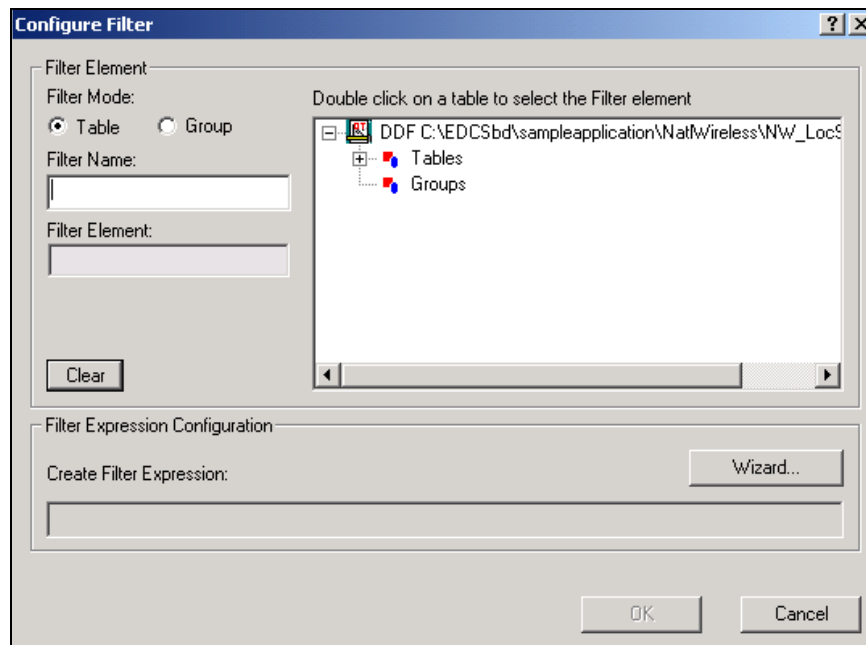
3. Select the filter mode to indicate whether the filter is on a table or group. (Keep Table node selected for National Wireless.)

4. Enter a descriptive name for the filter. The name must not contain spaces or special characters, must start with an alpha character, and cannot exceed 30 characters in length. (Enter **LocalOverFiftyDol**.)

5. Double-click the table (or group, if in group mode) you want to filter. If you make a mistake, click **Clear** and reselect the table. (Select the **LocalLineSummary** table.)

6. Click [Wizard...] to build a filter expression. The Define Expression dialog appears.
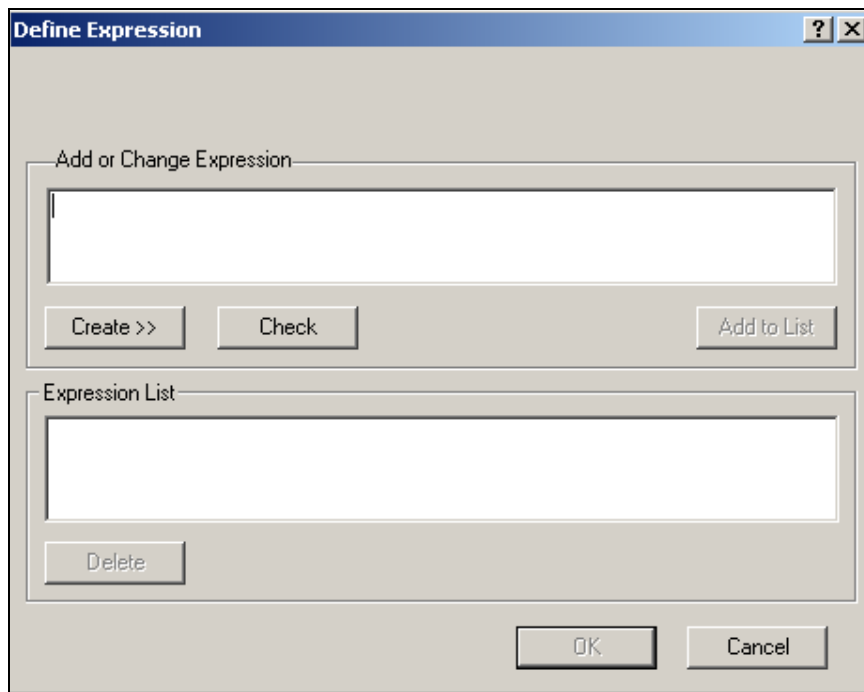
7. Click [Create >>] to display the buttons for selecting fields, operators, dates, etc. to build an expression in the **Add or Change Expression** box. Select or enter items in the sequence you want them to appear in the expression. (For detailed information on Data Filtering Expressions, see Appendix A.) A drop-down menu with button selections appears.
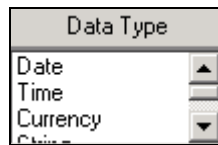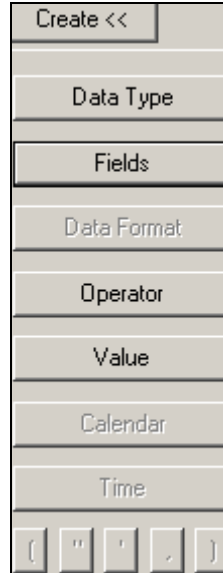
8. Click **Data Type** **and select** the type of data in the
column or field you want to compare. You do not need to select a data type
for a string. (Select **Currency**.) Composer adds the data type and an open
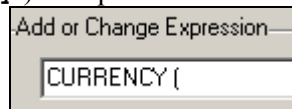
parenthesis to the expression: .
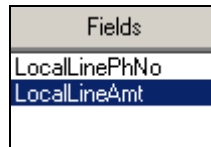
9. Click **Fields** **and select** the column or field you
want to compare. (Select **LocalLineAmt**.) Composer adds the column name
(and a comma, if data type is not string) to the expression.
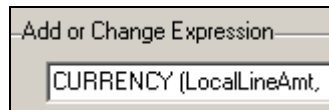
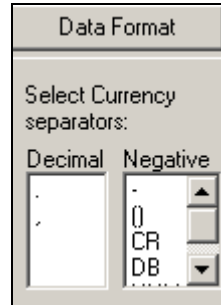10. If you are filtering on a date, time, decimal, or currency value, click **Data Format.**

11. Select the format of the data as stored in the input file. (For additional information about valid format strings you can enter, see Appendix A.) Composer adds the format and a close parenthesis to the expression. Manually add any required separators, as in **DATE (DueDate, "%m/%d/%Y").** (For NatlWireless, select a period "**.**"for the decimal separator, and a minus sign "**-**" for the negative separator.)
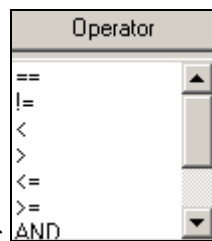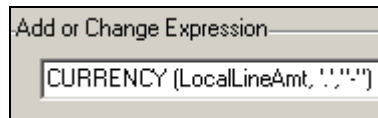
12. Click **Operator** and select the appropriate operator for the filter expression. (Select "**>=**".) Composer adds the operator to the expression:
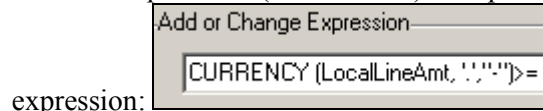
13. Click **Data Type** and select the same data type you selected in the first part of the filter expression. (Select data type Currency.) Composer adds the data type and an open parenthesis to the second part of the expression.

14. Specify the field, column, date, time, or value for the filter comparator:

- **To specify a date for comparison**, click **Calendar** and choose a specific month, year, and day. Composer adds the date in quotation marks, followed by a comma, to the expression.

- **To specify a time for comparison**, click **Time.** Click the hour, minutes, seconds, or AM/PM to highlight, and use the up and down arrows to specify the value: . Composer adds the time in quotation marks, followed by a comma, to the expression.
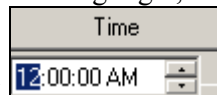
- **To specify a currency, decimal, or integer value**, enter the value between quotation marks followed by a comma. Add a close parenthesis after an integer value. (For NatlWireless, enter "50" followed by a comma, then select "**.**"and "**-**" separators.)
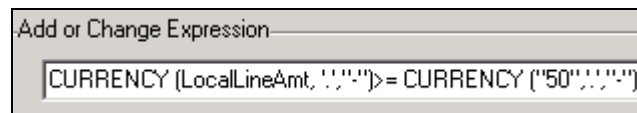
- **To specify a string value**, enter the comparator string in quotation marks, as in "Massachusetts". Type quotation marks directly or click .

- **To compare a string column to a NULL or EMPTY**, click **Value** and select **NULL** or **EMPTY**. Composer adds the word NULL or EMPTY to the expression.

- **To compare the value to a column or field**, click `Fields` and choose the column or field. Note that you can compare a column to a column, pure field, or dynamic field. In group mode, you can compare a group field to a group, pure, or dynamic field, but not to a column. Composer adds the column or field name in quotation marks, followed by a comma, to the expression.

15. If the data type is a date, time, currency, or decimal, click `Data Format` and select the same format you selected in the first part of the expression. Composer adds the format to the expression followed by a close parenthesis. Manually add any separators to the second part of the expression, as in: `DATE ("10/15/2002","%d/%m/%Y").`

16. When the filter element expression is complete, click `Check` to verify the integrity of the expression.

17. When the expression is correct, click `OK`.

18. Click `Add to List`. The expression appears in the Expression List.
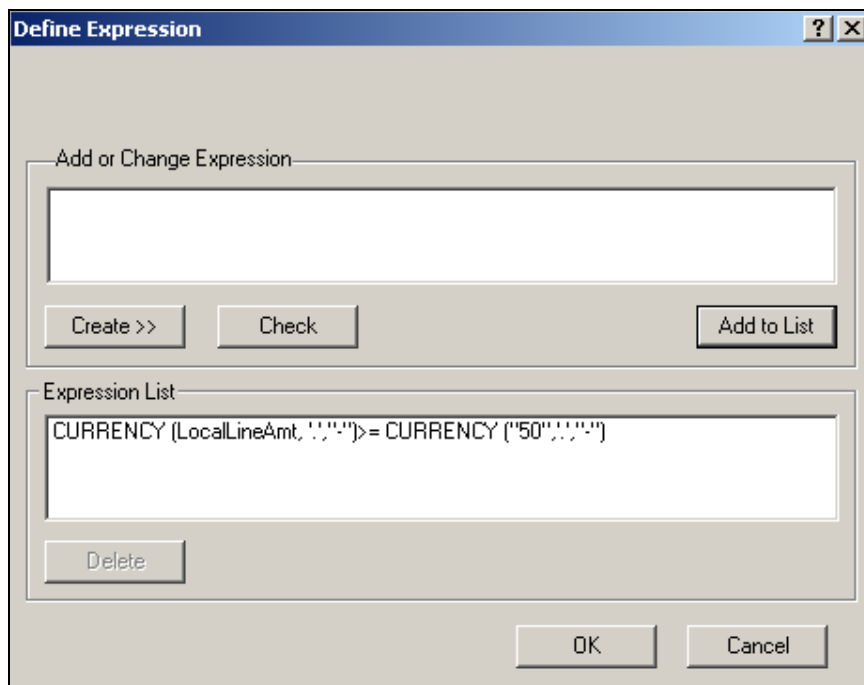
19. To create a compound expression, click **Operator** and select AND, OR, or NOT, and creating another expression.

20. Click **OK**.

21. Click **OK**.on the Configure Filter screen. Composer displays the filter name in the Application Tree. The "T" next to the filter name indicates it is on a table.
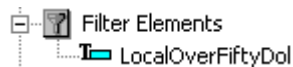
22. Click to save the ALF.

Creating a Top/Bottom Filter

Top/bottom filters let you create a counted subset of filtered data from either the top or bottom of the filtered data set. For example, you could create a top/bottom filter element to display the top ten most expensive phone calls in a telecomm statement.

To create a top or bottom filter:

1. With the ALF open in Composer, right-click the Top/Bottom Elements node. (Open NW_LocSummary.)

2. Select **Create Top/Bottom Element** from the right-click menu. The Configure Top/Bottom Filter dialog appears.
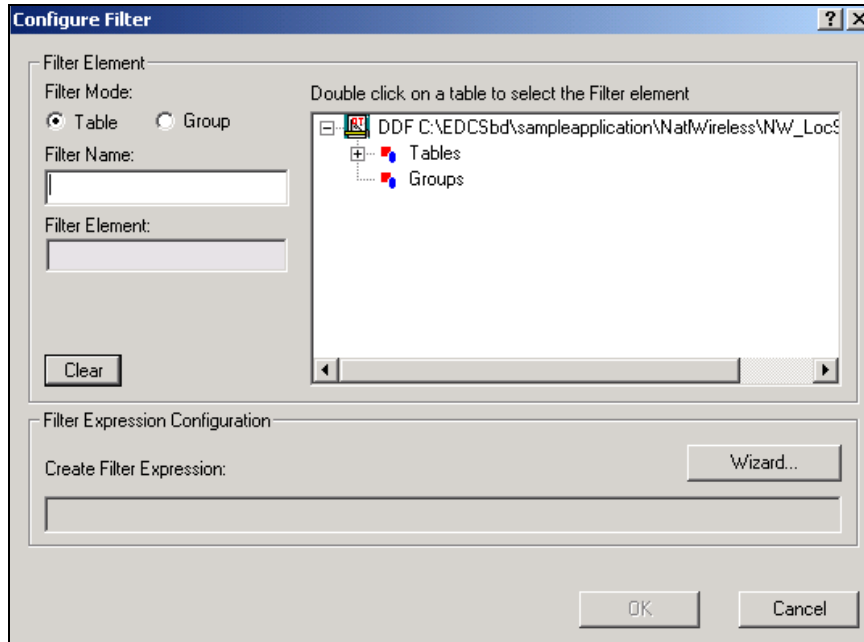
3. Select the filter mode to indicate whether the top/bottom filter is on a table or group.

4. Enter a descriptive name for the top/bottom filter. The name must not contain spaces or special characters, must start with an alpha character, and cannot exceed 30 characters in length. (Enter **LocalTop10Filter**.)

5. Select the group or table you want to filter. If you make a mistake, click **Clear** and reselect the table. (Open the **LocalChargeSummary** table.)

6. Select the column you want to filter. If creating a filter on a group, you can select a group field. (Select **LocalChargeAmount**.)

7. Select a data type. (Select **Currency**.)

8. If the data type is a date or time, then select the appropriate format in the Data Format field. See Appendix A for more information about data definitions.

9. If the data type is date or time, click the Validate button to verify that the format in the Data Format field complies with the data type. (See Appendix A for more information about data definitions.)

10. Select a value for the number of data entries you wish to present. (Select **10**.)

11. Select the direction of the filter; top or bottom of the data. (Select **Top**.)

12. If the data type is currency or decimal, select a decimal separator. (Select a decimal point ".")

13. If the data type is currency, select a negative separator. (Select the minus sign.)
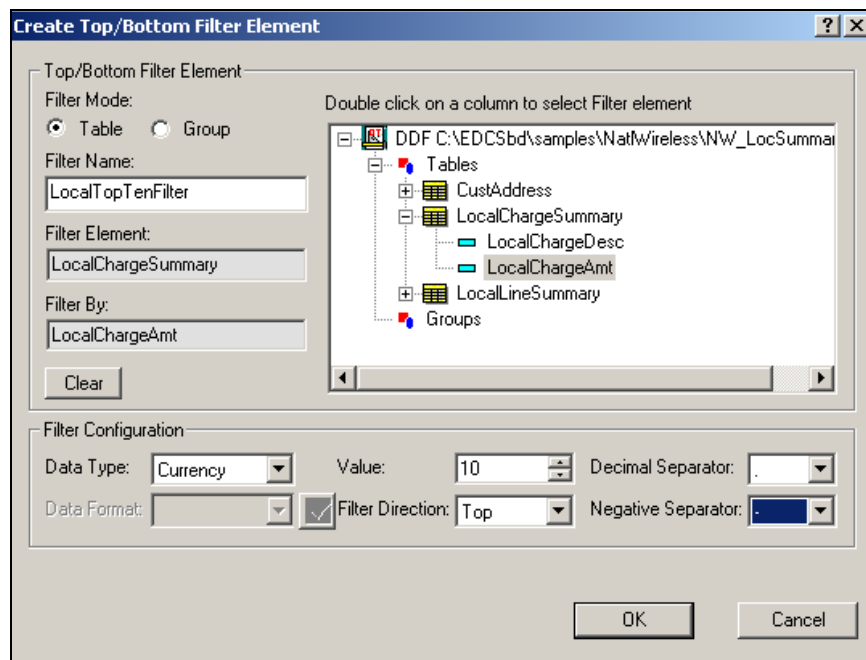
14. Click **OK**. The top/bottom filter appears under Top/Bottom Elements of the Application Tree. The "T" next to the filter name indicates it is on a table.
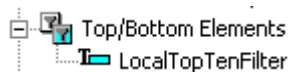
Linking a Filter Element to a Statement

After creating a filter element, you must add a link to the statement for users to view the filtered data. (You can optionally create a filter or sort element while creating a link.) You can create a link on static text, an image, or on an **[E]** tag.

You can have a combination of link to view, sort, and filter links in a statement.

You can optionally choose to reset the URL as part of the link. This enables you to clear any previous URL actions the Web user took, such as sorting or filtering data, before applying the filter. If you do not choose this option, the filter would be cumulative on any previous URLs the user chose in the session.

To create a link to a filter element:

1. With the ALF open, select the item you want to use as a link (be sure to completely highlight the item).

2. Right-click and select **Create Link** from the right-click menu. The Create Link dialog appears.

3. Click **Parameters**. The Add Parameters dialog appears.

4. If you want to display the filtered data in a different view, enter the name of the **Target View**. (You can also add a filter element, sorting element, or a name/value pair from dynamic pattern matching to the link.)

5. Click the **Filter Elements** or **Top/Botton Elements** drop-down list and select the filter element you want to link.

Tip   You can optionally create a sort or filter element from this dialog by clicking
Configure and Add . See the appropriate procedure in this guide for creating the particular element.

---

6. Click Add To Link . Composer adds the filter element to the Added Elements list.

7. If you are using more than one sort or filtering function, click **Reset URL** if you want to add the Reset URL function to the link (this clears the history of the previous URL before applying this filter).

8. Click **OK**. The Create Link dialog appears showing the view name and filter element.
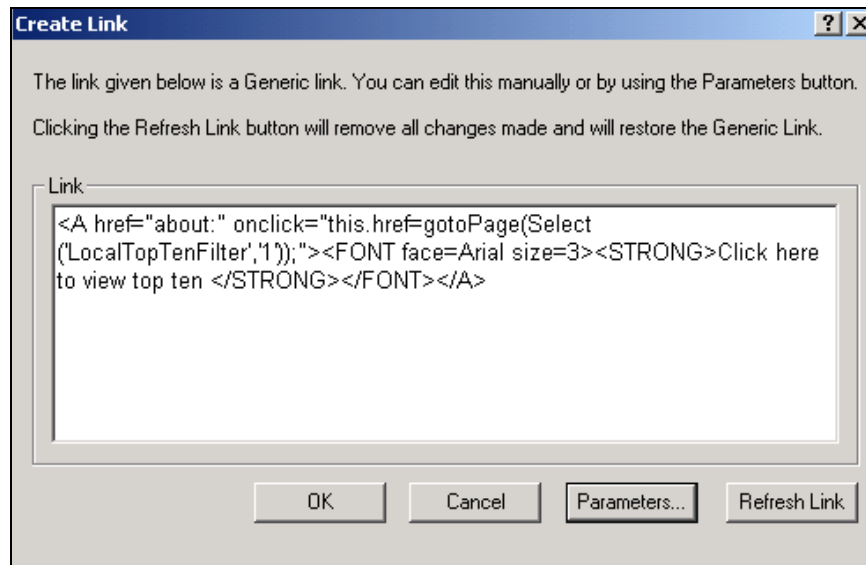
9. Click **OK** to save the link to the HTML Template. Composer adds a blue hyperlink to the statement.

10. Save the template by clicking .

11. Click to save the ALF if you added a link to a table column.

Modifying a Filter Element

To change the properties of a filter element:

1. With the ALF open in Composer, right-click the filter element (general or top/bottom) and select **Modify Filter Element** (or **Modify Top/Bottom Element)** from the right-click menu. Composer displays the Configure Filter (or Configure Top/Bottom Filter) dialog.

2. Edit a top/bottom filter as needed, or click `Wizard` to edit a general filter. To delete one or more expressions from a compound filter element, select the expression and click `Delete Expression`.

3. Click `OK`. (Click `OK` again if editing a general filter.)

Deleting a Filter Element

To delete a filter element:

*   With the ALF open in Composer, right-click the filter element (general or top/bottom) and select `Delete Filter Element` (or `Delete Top/Bottom Element)` from the right-click menu.

Using Combine Elements to Page, Sort, & Filter Across Table or Group Occurrences

Creating a combine element on a group or table lets you to "combine" occurrences of groups or tables when paging, sorting, or filtering a statement.

Defining a table or group as a combine element eliminates breaks across occurrences of the group or table for *all* page, sort, and filter (general or top/bottom) elements defined for the group or table.

Creating a combine element lets the Web user:

*   **Click through fewer pages to view a statement.** By default, paging elements start a new Web page for each new table or group occurrence, regardless of how many rows are in each table, often not filling the entire page. Creating a combine element on a table or group eliminates page breaks at the end of the group or table occurrence, displaying them contiguously, requiring fewer pages. Paging lets you display a consistent number of HTML lines per page. (You must create a page element on the table or group to configure the paging feature; see "Paging Statements" on Page 108.)

- **Sort or filter (general or top/bottom) statement data across all table or group occurrences.** By default, eaDirect sorts or filters each occurrence of a table. Creating a combine element on a table or group lets users sort or filter data across all occurrences of the table or group. (You must create a sort or filter element on the table or group to configure these features; see "Sorting Tables" on Page 97 or "Filtering Data" on Page 113.)

Create a combine element wherever it makes sense to remove the boundaries of table or group occurrences and let users page, sort, or filter across this statement data.

Unlike other element types, you do not create a link for a combine element.

You define combine elements in the default ALF only, however eaDirect applies the combined property to any elements that appear in any switch ALFs.

> **Caution** If you create a combine element on a group that has group fields at that level, eaDirect displays the value in the first occurrence of the group only. The data associated with the group field in subsequent occurrences of the group does not appear, and is effectively lost to the viewer. (Group fields below that level still appear with each occurrences.)

Creating a New Combine Element

To create a new combine element:

1. Creating With the default ALF open in Composer, right-click the Combine Elements node in the Application Tree.

2. Select `Configure Combine Elements` from the right-click menu. Composer displays the Configure Combine Elements dialog.
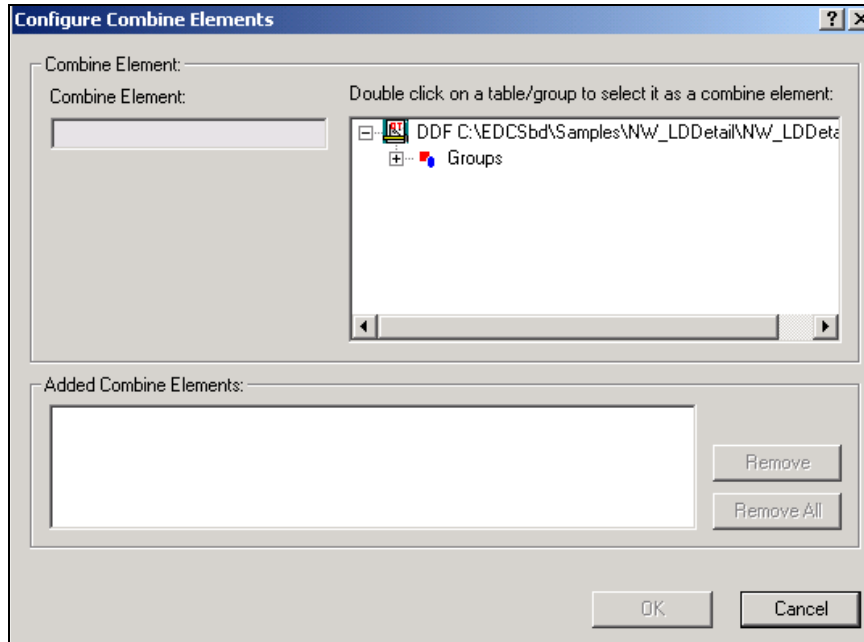
3. Select the group or group table on which you want to combine occurrences.

4. Select any additional groups or group tables to create additional combine elements (if any).

5. Click **OK**.

Viewing and Deleting Combine Elements

To view a list of or delete combine elements:

1. With the default ALF open in Composer, right-click .

2. Select `Configure Combine Elements` from the right-click menu. Composer displays the Configure Combine Elements dialog.

3. You can see the complete list of combine elements for the view listed under Added Combine Elements.

To delete a combine element, select the element from the list and click **Remove**. (To remove all combine elements, click **Remove All**.)

4. Click **OK**.

Totaling with Arithmetic Elements

You can create an arithmetic element to calculate and display the sum of all values in a table column or the total number of rows in a table (line count). Arithmetic elements let you create subtotals or totals not provided in your application's data input file. You can also calculate totals of sorted or filtered data. You can create an arithmetic element on a group field and place it anywhere on a template, on any hierarchy.

You can use arithmetic elements with paging, however, *if you do not also sort or filter the data, you can only display a total at the end.* (Paged data extracts progressively, and is therefore not available for summing until the last set displays. If you sort or filter the data when paging, however, data extraction is not progressive, and is extracted and available for subtotaling individual tables in a group for display on each page.)

You can configure an arithmetic element to create a total for:

- A single occurrence of a table (one total for one occurrence of a table, also called a "pure table")

- Each occurrence of a table (a total for each table occurrence in a group)

- Each occurrence of a group (one total for all tables in a group occurrence)

- All occurrences of a table across all occurrences of a group (one total for the statement)

Once you configure the arithmetic element for a table, you must drag and drop the arithmetic element to the appropriate place in the default HTML template or in the group template. The value eaDirect calculates depends on where you place the arithmetic element in the statement:

- **In the default HTML template** - To total a column in a single occurrence of a table or to generate one grand total of column data across all occurrences of a table, add the arithmetic element directly to the default HTML template.

- **In the group HTML** - To calculate a total on each occurrence of a group, place the arithmetic element in the group template. To calculate a total for each table occurrence in a group, add an HTML table to the group template and place the element inside the table. (This behaves like a footer to the table section in the ALF and calculates a total for each group occurrence.)

You can total the values in a column of data that contains a time, currency, integer, or decimal value only. You can calculate a line count on any type of column.

For time, currency, and decimal values, you must specify how the data is stored in the input file (time requires a strptime data format). You can also specify the output format, as well as any text or characters to display with the output, such as "Total:" and a currency symbol ($, £, etc.).

Specifying output format with the arithmetic element (vs. directly on the template) is necessary when calculating a subtotal for each table in a group or when you apply paging without also sorting or filtering. eaDirect can't compose a sum until all data has been extracted with the final page, and you wouldn't want the words "Total:" or other output formatting to appear on every page without the sum or count.

Caution	Be sure to specify the correct data format in the arithmetic element. If some rows of data match the specified data format and others do not, eaDirect includes only matching rows in the total. When you validate elements, Composer reports the non-matching rows.

Creating a New Arithmetic Element

To create an arithmetic element and add it to your statement:

1. With the ALF open in Composer, right-click the Arithmetic Elements node in the Application Tree, and select **Create Arithmetic Element**. (For NatlWireless, open NW_LocSummary.alf.) Composer displays the Configure Arithmetic Element dialog:
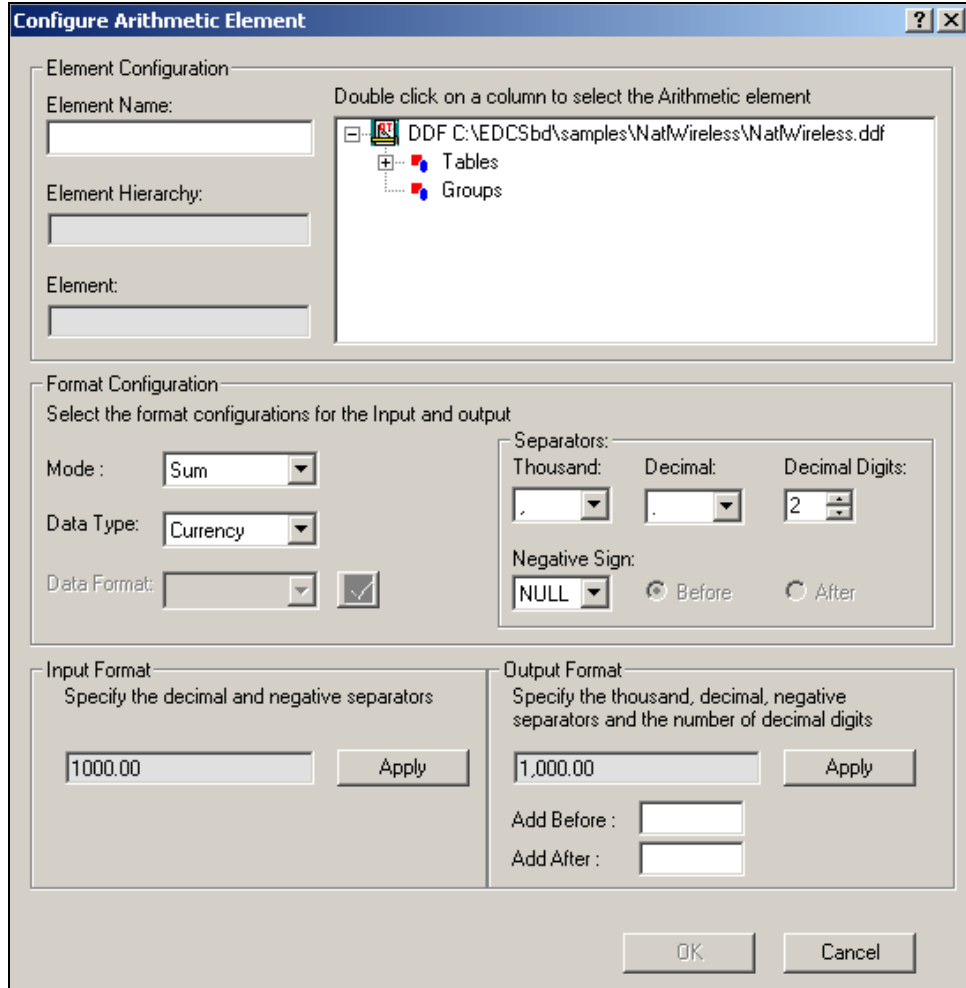
2. Enter a meaningful name for the arithmetic element. (Enter
    PhoneLineCount.)

3. Expand the **Tables** or **Groups** node and select the table you want to create
    an element on (element hierarchy). (In NW_LocSummary.alf, expand the
    LocalLineSummary table.)

4. Double-click on the column or field you want to total or count. The element
    can be a group field. (Select LocalLinePhNo.)

5. Select the type of arithmetic element you want to create, either sum or count, from the Mode drop-down list under Format Configuration. (Select Count.)

6. If the arithmetic element is to sum values, select the type of data in the column: time, currency, integer, or decimal, then specify the following data input and output information.

    Time: Select or enter the strptime format used in the input file. Click the ✓ button to verify its validity. Click the **Apply** button under Input Format, then select or enter the format you want to use to display the output, and click the **Apply** button under Output Format.

    Currency: Specify the decimal and negative separators used in the input file, and click the **Apply** button under Input Format. Specify the thousand, decimal, and negative separators and the number of decimal dig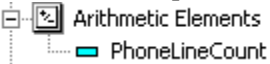its to use to format the output, and whether you want it to appear before or after the number, and inside or outside of brackets () (if you selected brackets as the negative sign). Click the **Apply** button under Output Format

    Decimals: Specify the decimal separator used in the input file, and click the **Apply** button under Input Format. Then specify the thousand and decimal separators and number of decimal digits to use for the output format, and click the **Apply** button under Output Format.

Caution   Be sure to choose a data output format with enough digits to display the potential sum value (otherwise digits could be cut off).

---

7. Click **OK**. Composer displays the arithmetic element in the Application Tree:

8. Add the arithmetic element to the statement:

    *   **To total a column in a single occurrence of a table or to generate one grand total of column data across all occurrences of a table**, click and drag the arithmetic element from the Application Tree to the default HTML template.

- **To calculate a total on each occurrence of a group**, place the element in the group HTML template; select the table, right-click and select `Configure Table` from the right-click menu. Click-and-drag the element from the Application Tree to the group HTML template.

- **To calculate a total on each occurrence of a table**, place the element in the group HTML template; select the table, right-click and select `Configure Table` from the right-click menu. Add an HTML table to the group template and drag-and-drop the arithmetic element inside the table. (This behaves like a footer to the table section in the ALF and calculates a total for each table in the group.)

9. Click to save the ALF.

Modifying an Arithmetic Element

To modify an arithmetic element:

1. With the ALF open in Composer, double-click the Arithmetic Elements node to display a list of arithmetic elements in the Application Tree. Right-click the arithmetic element you want to edit.

2. Select `Modify Arithmetic Element` from the right-click menu. The Configure Arithmetic Element dialog appears.

3. Edit the properties of the element as you did when adding the new element, and click `OK`.

Deleting an Arithmetic Element

To delete an arithmetic element:

1. With the ALF open in Composer, double-click the Arithmetic Elements node to display a list of arithmetic elements in the Application Tree. Right-click the arithmetic element you want to delete.

2. Select **Delete Arithmetic Element** from the right-click menu.
    Composer asks if you're sure you want to delete the element:
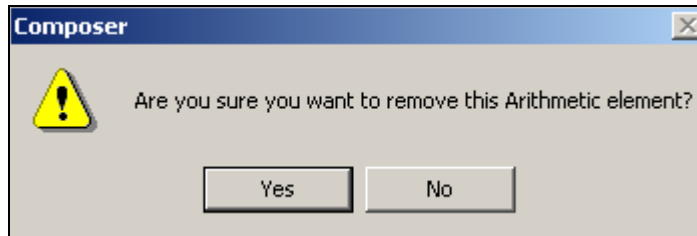
3. Click **Yes**. Composer removes the arithmetic element from the ALF.

4. Manually remove the [E]*ElementName*A[\E] tag from the correct HTML
    template.

Using the Simulator

Running a Simulation

Use Composer's Simulator feature to view the customized presentment of statement data as it would appear live in a browser. Simulator uses the DDF, ALF, and HTML templates to deliver a composed HTML version of the statements using a sample data input file. Simulator prevents the "blind design" technique often associated with application development. You can run the Simulator at any time during the development process.

You can enter values and check individual dynamic fields. Simulator also lets you select sort and filter elements to validate. You must select and validate each sort element define on the same table one at a time. By default, Simulator verifies all arithmetic elements used in the templates, reporting the results in the Data Type Report. (You can turn this option off.)

When you select elements to validate, Simulator generates the Data Type Report with the simulation results. You can view the report online, print, and/or save the log to a file.

To run a simulation:

1. Select **View>Simulate** or click ⬛ on the Simulator toolbar. The Simulate dialog appears.
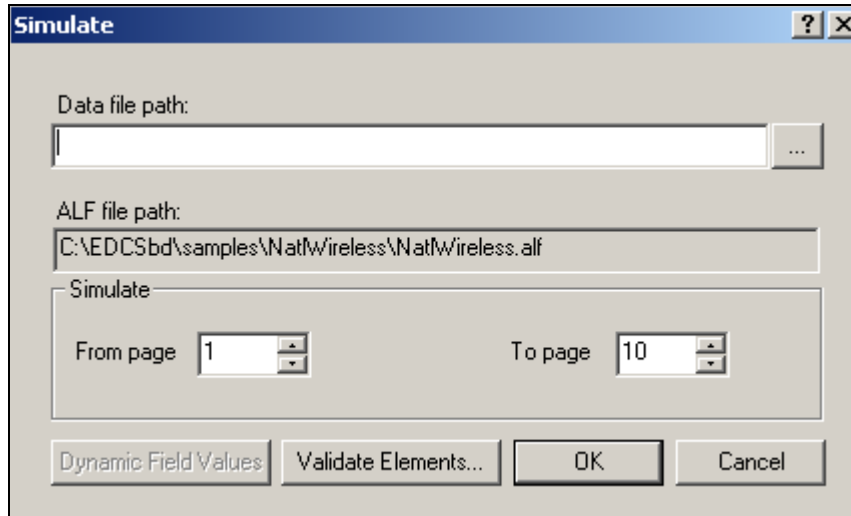
2. Select a data file. (For example, select NatlWireless.txt. for the NatlWireless.alf and the NatlWireless.htm template.)

3. Specify the page range to include in the simulation. (Enter `1-110` pages.)

4. If you are simulating using dynamic field values, see Simulating Dynamic Field Values.

5. If you are simulating sort or filter elements, see Simulating Sorting and Filtering Elements.

6. Click `OK`. The Simulate Bar appears displaying a composed HTML view of each statement appears in the right-pane and a simulated listing of the data.

7. To view the statements, click the navigation icons   on the Simulator toolbar or click on the pages in the simulated document.

Tip — You can drag and drop columns to change their position in the "hit list," which is useful for comparing columns.

---

8. To return to the uncomposed HTML statement work area, click the toggle icon .

9. After editing, click the refresh icon to refresh the simulation without navigating to a sample data source.

Simulating Dynamic Field Values

To simulate dynamic field values:

1. While simulating the statement, from the Simulate dialog, click **Dynamic Field Values**. The Apply Dynamic Field Values dialog appears.
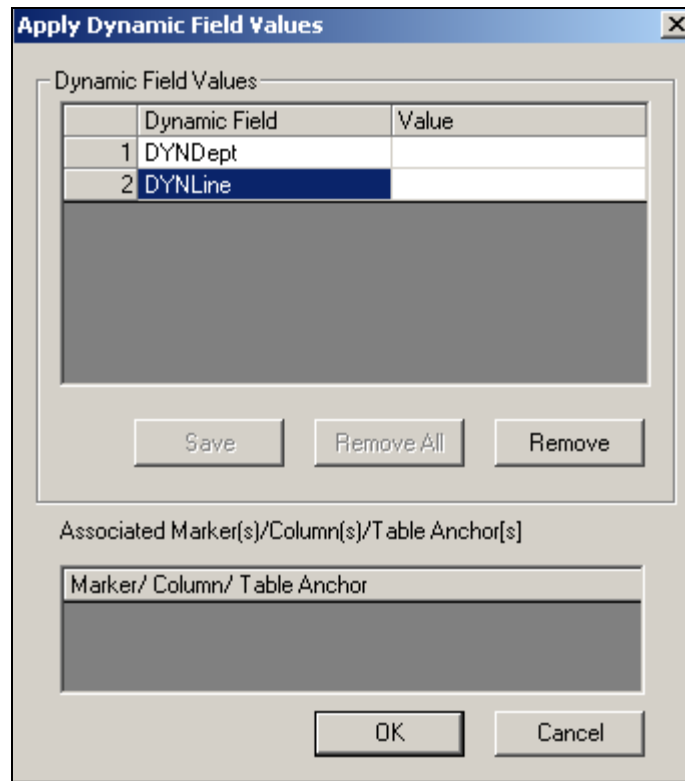
2. Click to select the dynamic field for simulation. Selecting a dynamic field will list the associated names where the dynamic field is used in the DDF data elements in the Associated Marker(s)/Columns(s)/Table Anchors(s)" List Box. (In this example, the **DYNLocalLinePhone** field is selected.)

3. Double-click in the Value field to enter the dynamic value. (Enter **202**.)

    Click **Save** to save the field values to the registry.

Click **Remove All** to remove all dynamic field values from the registry.

Click **Remove** to remove the selected dynamic field value from the registry.

4. Click **OK**. The Simulate dialog appears.

5. Click **OK**. The Simulate Bar appears with a composed HTML view of each statement, with the dynamic field values in the right-pane.

Caution Simulator continues to apply the dynamic field value for subsequent simulations. To remove the value, double-click in the Value field on the Apply Dynamic Field Values dialog and remove the value.

Simulating Sorting and Filtering Elements

To simulate sorting and filtering:

1. While simulating the statement, from the Simulate dialog, click **Dynamic Field Values**. The Validate Elements dialog appears.
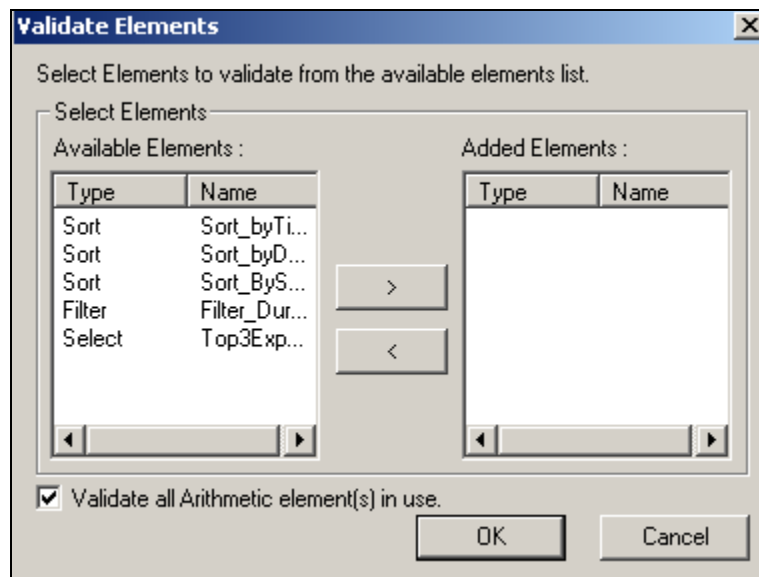
2. Select the filters and/or sort elements you want to sort from the Available Elements list and click [>] to place it in the Added Elements list. (You can also double-click on an element to move it from one window to the other.) You can only select and validate one sort element defined on table at a time.

3. If you want Simulator to validate all arithmetic elements placed in the templates, check the "Validate all Arithmetic element(s) in use" option.
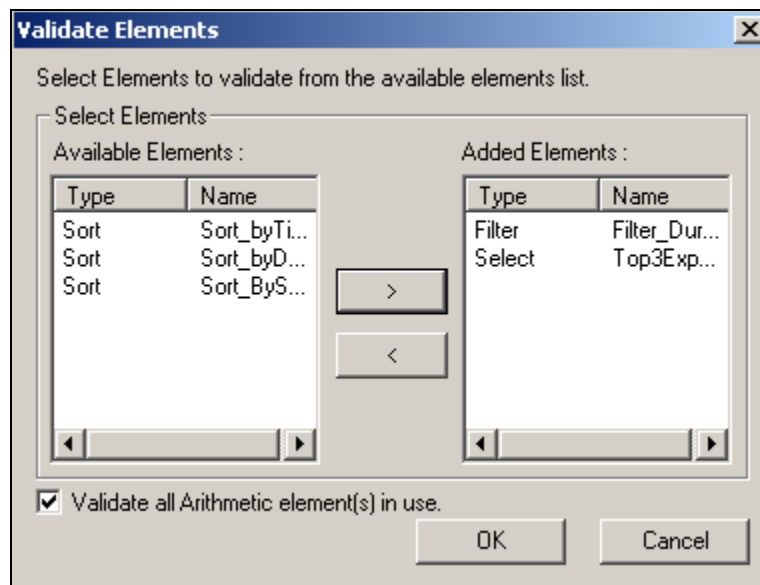
4. Click **OK**. The Simulate dialog appears.

5. Click **OK**. The Simulate Bar appears with a composed HTML view of each statement with the sorted and filtered data only appearing in the right-pane.

6. Click the Data Type Report icon in the Simulate tool bar. Composer displays the Data Type Report dialog. Scroll down the report to view element details.

```
Data Type Report                                    _ | □ | ×
┌─────────────────────────────────────────────────────────┐
│  🖫  🖶                                                    │
│  ┌───────────────────────────────────────────────────┐▲ │
│  │Application Details:                                │  │
│  │--------------------                                │  │
│  │DDF    C:\EDCSbd\Samples\NW_LDDetail                │  │
│  │\NW_LDDetail.ddf                                    │  │
│  │ALF    C:\EDCSbd\samples\NW_LDDetail                │  │
│  │\NW_LDDetail.alf                                    │  │
│  │Data File    C:\EDCSbd\samples                      │  │
│  │\NW_LDDetail\datafile                               │  │
│  │\NationalWireless.txt                               │  │
│  │                                                    │  │
│  │Document Details:                                   │  │
│  │----------------                                    │  │
│  │Document Number     1                               │  │
│  │Start Page    1                                     │  │
│  │End Page      8                                     │  │
│  │                                                    │  │
│  │Created Date & Time:                                │  │
│  │-------------------                                 │  │
│  │2002/5/10 11:59:01                                  │▼ │
│  └───────────────────────────────────────────────────┘  │
│                                                          │
│                      ┌──────────────┐                    │
│                      │     OK       │                    │
│                      └──────────────┘                    │
└─────────────────────────────────────────────────────────┘
```
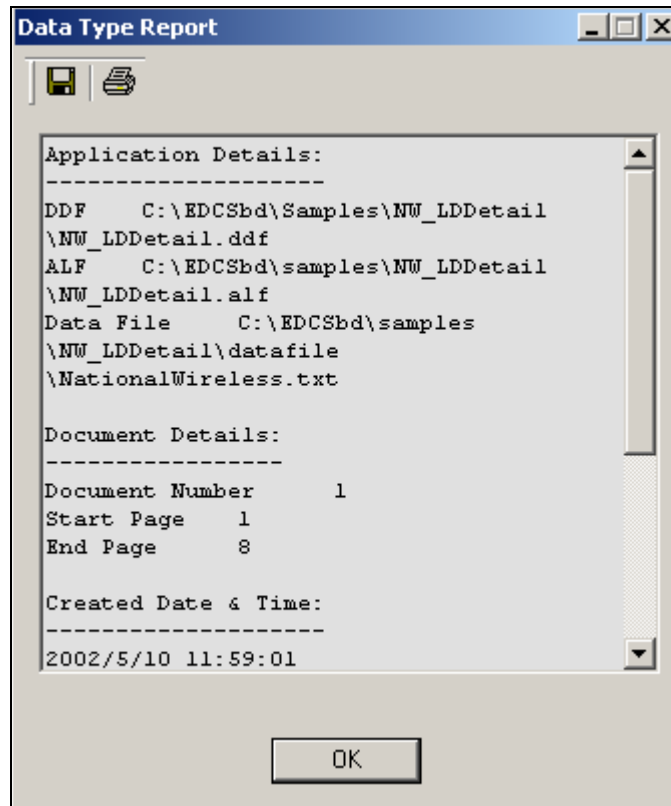
7. To save the Data Type Report information as a text document with a time stamp as the filename, click 🖫

8. To print the Data Type Report, click 🖶.

9. Click OK. Composer prompts you to save the log if you have not.

⬭ Caution   Composer continues to apply the added elements to subsequent simulations. To remove the elements, enter the Apply Dynamic Field Values dialog, select the element in the Added Elements list and click

[<] (or double-click on the element).

Mapping Tables as Charts

7

Chart Information in the ALF

You can drag and drop the tables defined in the DDF into the HTML template using the WYSIWYG or the HTML editor. You can represent a table as either a text table or as a chart in the HTML template.

> **Tip**   The charting option is NOT available for a table with only one field.

To create a chart:

1. Click the **WYSIWYG** tab or **HTML** tab.

2. Click the **Definition** tab in the Tree.

3. Click to open **Tables** in the Tree.

4. Drag and drop the table 🔳 definitions into the HTML template. The table assumes the properties of the area in which it is placed. (Drag and drop the LocalChargeSummary table to the HTML Template.)
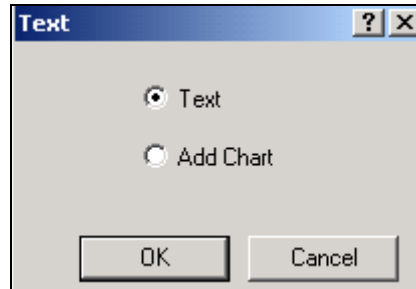
5. Select **Chart.**
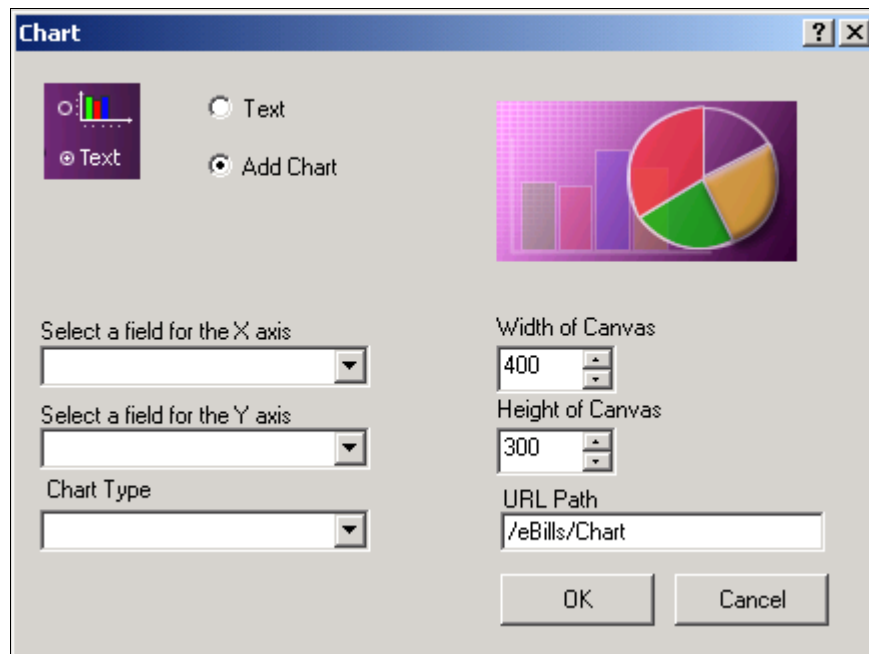
6. Select a field for the X axis of the chart. (Select **LocalChargeAmount**.)

7. Select a field for the Y axis of the chart. (Select LocalChargeDesc.)

8. Select the type of chart. (Select Pie.)

9. Select the width and height settings for the chart. (Leave at 400 and 300 respectively.)

10. Enter the URL path. (Enter the path to the Web Server chart storage area.)

11. Click **OK**. (The tag `[E]LocalChargeSummary_0,U[/E]` appears.)

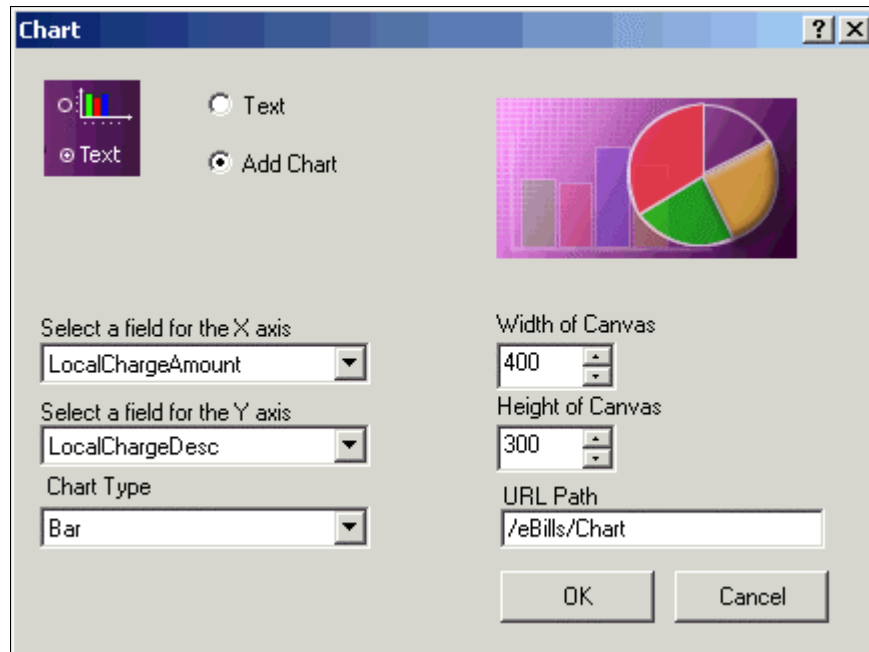   At this point a place holder for the chart is added to the template and a properties files is created. Example: If the table name is **Test** then a "*Test_0.Properties* file is created in the *AppProfiles\DDN Name* directory (Where the ALF and Templates are located). When creating a chart, make sure that there is more than one field in the table and that there are numeric values included. Even though the field type is not numeric, the values are converted into a numeric format by the Chart engine.

12. If there are placeholders in the HTML template, delete them. (Delete the "XX" placeholders.)

13. Click the **Save Template** icon.

14. Save the ALF by clicking the **Save ALF** icon.

Tip	When mapping a table to a template in the Composer, it is not necessary to encapsulate the table with HTML table row *<TR>* and table data *<TD>* (cell) tags. The table rows and table data (cells) are generated when the data is dynamically pulled from the data file and passed to the edocs WebComposer object. This object formats the table rows, cells, and font characteristics of the data based on the settings defined in the Composer.

Charting Attributes

For an eaDirect view to display charts, an ALF must be constructed and placed in the view directory. The ALF will provide the document type definition (DTD) in the ALF (usually at the beginning of the document).

The DTD will have the attributes (described here) assigned to the chart.

Attribute Name	Meaning
Height	Total height of the canvas
Width	Total width of the canvas
HidePieLegend	Toggles the display of legends for Pie charts
AddValueToLegend	Displays the percentage in the chart legend
BaseURL	Points to the location where the chart images are stored
UnixChart	Chart types in a UNIX platform

You will notice other attributes listed in the ALF. These attributes are placed in the ALF for backward compatibility with previous versions of eaDirect and have no effect on the current version of eaDirect.

Chart Information Contained Within the ALF

You can see the HTML for the chart. For example, after defining your table properties and saving your ALF, open the *NatlWireless.ALF* in Notepad and scroll down to the **[CHARTS]** section. There you will find the HTML that will be pulled into the Statement Template during assembly (when the chart tags are found). An example of chart information follows:

```
<CHARTS><Chart Name = "LocalChargeSummary_0"RecordName =
"LocalChargeSummary"TopTitle = ""BottomTitle =
""LeftTitle = ""RightTitle = ""XField =
"LocalChargeDesc"YField = "LocalChargeAmount"Key =
"1"StackedStyle = "0"ColorScheme = "0"GridLines =
"3"Full3D = "1"AngleX = "45"AngleY = "0"Atribute =
"109577474"MarkerVolume = "3"Shadow = "1"MultiShape =
"11"Dimension_3D = "1"View3DDepth = "80"Type =
"0"CGITimeSpan = ""BackgroundColor = ""ForgroundColor =
""Height = "300"Width = "400"LegendShow =
"1"LegendToolSize = "100"LegendToolStyle =
"167116800"HidePieLegend = "0"SeriesColor = ""LeftGap =
"40"RightGap = "40"ImgQuality = "75"ImgSmooth =
"0"AddValueToLegend = "0"BaseURL = ""UNIXChart =
""></Chart></CHARTS>
```

Chart Properties

The chart properties file is a list of name-value pairs that control the type of chart, location of the axes, fonts, colors, whether to use three-dimensional effects, etc. To change the display of some aspect, an alternate value for the controlling property must be provided. If there is a conflict between a chart attribute defined in the ALF and a value defined in the chart property, it will override the chart attribute.

The chart properties file is included as part of your published application view using eaDirect's Command Center. Publishing views are described in detail in the *eaDirect Data Presentation Production Guide*. Publishing chart views follows the same general process. The following summarizes the parts specific to charts:

- In the Publisher window, select the **Create** button. Set values for "ddn" (e.g., NationalWireless), job type = "HtmlDetail", and viewName (e.g. "Cell_0").

- In the next window that pops up, in the Web-Views section, which is also the upper portion of the page, select the option "Chart 0". You are next requested to specify, through your file system, the chart properties file. A file extension of *.properties* is required.

The following sections describe the types of properties included in the file.

Chart Type

The primary chart property is Type. By setting Type = Pie, one obtains a Pie chart, using "Bar" instead, results in a bar chart. The table below itemizes the various chart types currently available to eaDirect running on Unix platforms. Note that the "Type" property and its value are both case-sensitive. The remainder of the properties and their values are not so limited.

Bar	HiLoBar	HorizBar	HorizHiLoBar
IndBar	LabelLine	Line	Pie
Polar	StackBar	StackColumn	Speedo
Stick	StickBar	Regress	

The following are issues related to the chart types:

- Do not set properties that are not applicable to a chart type. For example, do not set Axis properties when requesting a Pie chart.

- Do not set Bar properties while rendering a Pie chart. For charts created using the Composer tool, the chart types: HiLobar, HorizHiLoBar, and Speedo are not available.

- If you want a Bar chart that displays elements of individual data series in different colors, select the IndBar type.

Chart General Properties

General chart properties determine whether the legend should be displayed, whether to provide three-dimensional effects, and finally whether to display the axis. Note that for Pie type charts, the convention is not to display the axis. For example:

```
Chart.LegendVisible=false

Chart.Name=MyChart

Chart.XAxisVisible=true

Chart.YAxisVisible=true

Chart.ThreeD=false
```

```
## the XOffset and YOffset determine the amount of three
## dimensional effect along the X and Y axes
respectively.

Chart.XOffset=0

Chart.YOffset=0

Chart.Quality=1
```

The `Quality` property defaults to 1 for the highest quality image possible. A value of 0 sets it to a lower quality image.

Background Properties

Titles and sub-titles are elements of the chart background. Their color, font and string value are controlled by appropriately setting the following background properties:

```
## Red, Green and Blue values, valid values 0-255
##
Background.Gc.FillColor.Red=0
Background.Gc.FillColor.Blue=0
Background.Gc.FillColor.Green=0

Background.Gc.LineColor.Red=0
Background.Gc.LineColor.Blue=0
Background.Gc.LineColor.Green=0

Background.Gc.LineWidth=1

Background.Gc.MarkerColor.Red=0
Background.Gc.MarkerColor.Blue=0
Background.Gc.MarkerColor.Green=0

Background.SubTitleColor.Red=0
Background.SubTitleColor.Blue=0
Background.SubTitleColor.Green=0

Background.SubTitleFont.name=TimesRoman
Background.SubTitleFont.style=plain
Background.SubTitleFont.size=12

Background.SubTitleString=null

Background.TitleColor.Red=0
Background.TitleColor.Blue=0
Background.TitleColor.Green=0
```

```
Background.TitleFont.name=TimesRoman
Background.TitleFont.style=plain
Background.TitleFont.size=12

Background.TitleString=null
```

Plot Area Properties

The plot area is the region bounded by the axis; where the data is actually plotted. The fill color for this region, in addition to the marker and grid line particulars, may be specified. For example:

```
Plotarea.Gc.FillColor.Red=0
Plotarea.Gc.FillColor.Blue=0
Plotarea.Gc.FillColor.Green=0

Plotarea.Gc.LineColor.Red=0
Plotarea.Gc.LineColor.Blue=0
Plotarea.Gc.LineColor.Green=0

Plotarea.Gc.LineWidth=1

Plotarea.Gc.MarkerColor.Red=0
Plotarea.Gc.MarkerColor.Blue=0
Plotarea.Gc.MarkerColor.Green=0
```

Axis Properties

The axis properties control the location of the axis on the canvas, its major and minor ticks, title, grid, and labels. Note that while only the X axis properties are presented below, an identical set of properties exist for the Y axis that starts with "Y." instead of "X.". For example:

```
## to skip labels that collide
## meaningful if the user can guess the missed labels
## generates a more legible axis without having to
reduce
## font size or increase the chart height or width.
##
X.Axis.CullingLabel=false

## Useful most times
##
X.Axis.AutoScale=true
```

```
## ending point for the axis. For AUTO_SCALE, the axis
## end will be automatically determined. For log-scaled,
it is
## a power of ten
##
X.Axis.AxisEnd=6.0

## starting point for the axis. For AUTO_SCALE,
## the axis start will be automatically
## determined. For log-scaled, it is a power of ten
##
X.Axis.AxisStart=0.0

## place bar within axis, set true for bar charts
##
X.Axis.BarScaling=true

## Display grid lines
##
X.Axis.GridVis=false

## Tilts the label by amount specified, integer value
##
X.Axis.LabelAngle=0

## Axis label color
##
X.Axis.LabelColor.Red=0
X.Axis.LabelColor.Blue=0
X.Axis.LabelColor.Green=0

## Axis Label Font
##
X.Axis.LabelFont.Name=TimesRoman
X.Axis.LabelFont.Style=plain
X.Axis.LabelFont.Size=12

## For double quantities such as currency
##
X.Axis.LabelPrecision=2

## Determines whether the label is visible
X.Axis.LabelVis=true

## Determines whether the axis line is visible
X.Axis.LineVis=true
```

```
## Determines whether the scale is log based
##
X.Axis.LogScaling=false

X.Axis.MajTickLength=5

X.Axis.MajTickVis=true

X.Axis.MinTickLength=2

X.Axis.MinTickVis=false

X.Axis.NumGrids=5

X.Axis.NumLabels=5

X.Axis.NumMajTicks=5

X.Axis.NumMinTicks=10

## reduce LlX to left shift axis position on canvas
##  increase for right shift, that is along the X
direction
##
X.Axis.Plotarea.LlX=0.2

## reduce LlY to move lower on axis on location
## increase LlY to raise axis, that is along Y direction
X.Axis.Plotarea.LlY=0.2

## similar to LlX and LlY, but controls upper right
corner
##
X.Axis.Plotarea.UrX=0.8
X.Axis.Plotarea.UrY=0.8

## places axis on left, right, top  or bottom
##
X.Axis.Side=1

X.Axis.TitleColor.Red=0
X.Axis.TitleColor.Blue=0
X.Axis.TitleColor.Green=0

X.Axis.TitleFont.Family=Times New Roman
X.Axis.TitleFont.Name=TimesRoman
X.Axis.TitleFont.Style=plain
X.Axis.TitleFont.Size=12

X.Axis.TitleString=Remember to set X axis title!
```

```
X.Axis.GridGc.FillColor.Red=0
X.Axis.GridGc.FillColor.Blue=0
X.Axis.GridGc.FillColor.Green=0

X.Axis.GridGc.LineColor.Red=0
X.Axis.GridGc.LineColor.Blue=0
X.Axis.GridGc.LineColor.Green=0

X.Axis.GridGc.LineWidth=1

X.Axis.GridGc.MarkerColor.Red=0
X.Axis.GridGc.MarkerColor.Blue=0
X.Axis.GridGc.MarkerColor.Green=0

## valid values = Left, Right, Top, Bottom
##
X.Axis.Side=Left

X.Axis.TickGc.FillColor.Red=0
X.Axis.TickGc.FillColor.Blue=0
X.Axis.TickGc.FillColor.Green=0

X.Axis.TickGc.LineColor.Red=0
X.Axis.TickGc.LineColor.Blue=0
X.Axis.TickGc.LineColor.Green=0

X.Axis.TickGc.LineWidth=1

X.Axis.TickGc.MarkerColor.Red=0
X.Axis.TickGc.MarkerColor.Blue=0
X.Axis.TickGc.MarkerColor.Green=0

X.Axis.LineGc.FillColor.Red=0
X.Axis.LineGc.FillColor.Blue=0
X.Axis.LineGc.FillColor.Green=0

X.Axis.LineGc.LineColor.Red=0
X.Axis.LineGc.LineColor.Blue=0
X.Axis.LineGc.LineColor.Green=0

X.Axis.LineGc.LineWidth=1

X.Axis.LineGc.MarkerColor.Red=0
X.Axis.LineGc.MarkerColor.Blue=0
X.Axis.LineGc.MarkerColor.Green=0
```

If your labels appear too close together on the X-axis, there is an alternative. You can place legends on the side, top, or bottom by specifying the chart property **Chart.Legend.Visible=true** and by specifying the values for height, width, and colors for the icons. Turn off the parts of the X-axis labels with the LabelVis property.

Legend Properties

The legend or a key for a chart are controlled by the following properties:

```
Legend.BackgroundVisible=true

## separation between rows of the legend
##
Legend.IconGap=0.02

## Legend icon height 0 < k < 1.0,
## where 1.0 = full height of canvas
##
Legend.IconHeight=0.05

## Legend icon width 0 < k < 1.0,
## where 1.0 = full width of canvas

Legend.IconWidth=0.07

## Red, Green and Blue values, valid values 0-255

Legend.LabelColor.Red=0
Legend.LabelColor.Blue=0
Legend.LabelColor.Green=0

## Legend Font
##
Legend.LabelFont.Name=TimesRoman
Legend.LabelFont.Style=plain
Legend.LabelFont.Size=12

## Legend lower x left corner 0 < y < 1.0,
## 1.0 = full width

Legend.LlX=0.0

## Legend lower y left corner 0 < y < 1.0, 1.0 = full
height
##
Legend.LlY=0.0
```

```
## Legend upper x right corner 0 < y < 1.0, 1.0 = full
width
##
Legend.UrX=0.0

## Upper y right corner 0 < y < 1.0, 1.0 = full height
##
Legend.UrY=0.0

## true for Legend below chart
##
Legend.VerticalLayout=false

## The following properties change a Legend's graphic
## component.
##
Legend.BackgroundGC.Gc.FillColor.Red=0
Legend.BackgroundGC.Gc.FillColor.Blue=0
Legend.BackgroundGC.Gc.FillColor.Green=0

Legend.BackgroundGC.Gc.LineColor.Red=0
Legend.BackgroundGC.Gc.LineColor.Blue=0
Legend.BackgroundGC.Gc.LineColor.Green=0

Legend.BackgroundGC.Gc.LineWidth=1

Legend.BackgroundGC.Gc.MarkerColor.Red=0

Legend.BackgroundGC.Gc.MarkerColor.Blue=0
Legend.BackgroundGC.Gc.MarkerColor.Green=0
```

Favorite Colors Properties

The properties file allows you to specify your favorite colors, to use as fill color for data slices. You can specify favorite colors to ensure a pleasing overall color scheme, in particular you may want your chart colors not to clash with the colors used in the page embedding the chart. The total number of favorite colors and their order of specification are important to bear in mind for the following reasons:

- **Number:** Favorite colors become active if there are enough of them to meet display needs. For example, if a Pie has "n" distinct slices, where n is greater than the number of favorite colors specified, the favorite colors will not be used. This is because there is no way to guess which colors would go well with those already specified.

- **Order**: The favorite colors are used in the order specified, so it is your responsibility to ensure that a color is distinct and distinguishable from its adjacent colors. Or else as in the case of a Pie chart, the resulting chart may not provide enough contrast

For example, the following defines the first and second favorite colors:

```
Favorite.1.Color.Red=201
Favorite.1.Color.Blue=92
Favorite.1.Color.Green=132

Favorite.2.Color.Red=51
Favorite.2.Color.Blue=52
Favorite.2.Color.Green=53
```

Pie Properties

The pie properties are available only when the selected chart type is Pie. These control the pie height, pie width, location of pie center on the canvas, and how the labels and legends must appear. For example, should the percentage sign be tacked on? Should the pie slice size be shown? Below is the list of various Pie properties along with their default values. Note that color and font related properties are broken into their primitive constituents.

```
Pie.LabelPosition=2

Pie.StartDegrees=0

## true displays pie slice name, e.g., "College Fund"
Pie.TextLabelsOn=false

## true displays the size of pie slice numerically,
e.g.,30.5
Pie.ValueLabelsOn=false

## true turns on the % sign after the pie slice size,
e.g., 30.5%
Pie.PercentLabelsOn=true

## Pie Label Color
##
Pie.LabelColor.Red=0
Pie.LabelColor.Blue=0
Pie.LabelColor.Green=0
```

```
## Pie Label Font,
## font names and style are the same as in java.awt.font
##
Pie.LabelFont.Name=TimesRoman
Pie.LabelFont.Style=plain
Pie.LabelFont.Size=12

## To obtain a circular Pie, scale height and width
using
## true canvas height and width to be equal in pixels.
## For width = 300 and height = 200,
## to obtain a Pie radius of 100, use:
##
Pie.Height=0.5
Pie.Width=0.33

## Locate Pie center. Take into consideration length of
labels
## in legend and whether the legend will be
## displayed horizontally or  vertically
##
Pie.XLoc=0.5
Pie.YLoc=0.5
```

If your labels appear too close together, there is an alternative. You can place legends on the side, top, or bottom by setting the chart property `Chart.Legend.Visible=true` and by specifying the values for height, width, and colors for the icons (see the *Legend Properties* section above). Turn off the label by setting the `TextLabelsOn`, `ValueLabelsOn`, and `PercentLablesOn` properties to false.

Bar Properties

The following properties control how bars are displayed in Bar Charts:

```
## Set the Bar baseline.
##
Bar.Baseline=0.0

## sets the cluster width
##
Bar.ClusterWidth=0.8
```

```
## Set to true to clip bars at Plotarea boundaries.
##
Bar.DoClip=false
```

Line Properties

Set to true to clip lines at the plot area boundary:

```
Line.Clip=false
```

Example:

The following is an example of a Pie chart and its non-default property settings:

```
Pie.Height=0.5
Pie.Width=0.33

Chart.LegendVisible=true
Legend.IconGap=0.02
Legend.IconHeight=0.05
Legend.IconWidth=0.07
Legend.LabelColor.Red=123
Legend.LabelColor.Blue=126
Legend.LabelColor.Green=129
Legend.LlX=0.7
Legend.LlY=0.4
Legend.VerticalLayout=false
Pie.PercentLabelsOn=true
Pie.TextLabelsOn=false
Pie.ValueLabelsOn=true
Pie.XLoc=0.5
Pie.YLoc=0.5
```

Changing Chart Properties

Changing chart properties is similar to the initial creation of a chart.

To display the properties of a chart:

1. In the Application Tree click to open the Charts ili. , and right-click on the chart 📊 you want to modify.
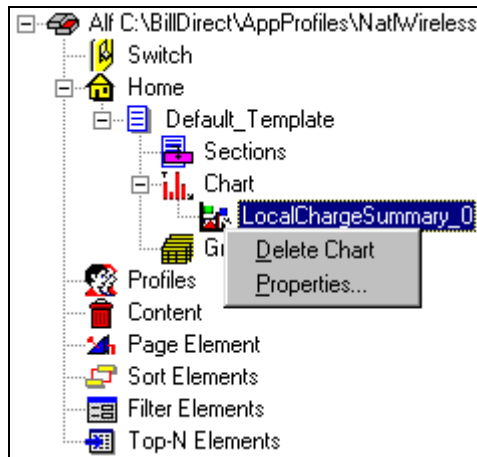
2. Select **Properties** from the right-click menu. The Chart Properties dialog box appears.
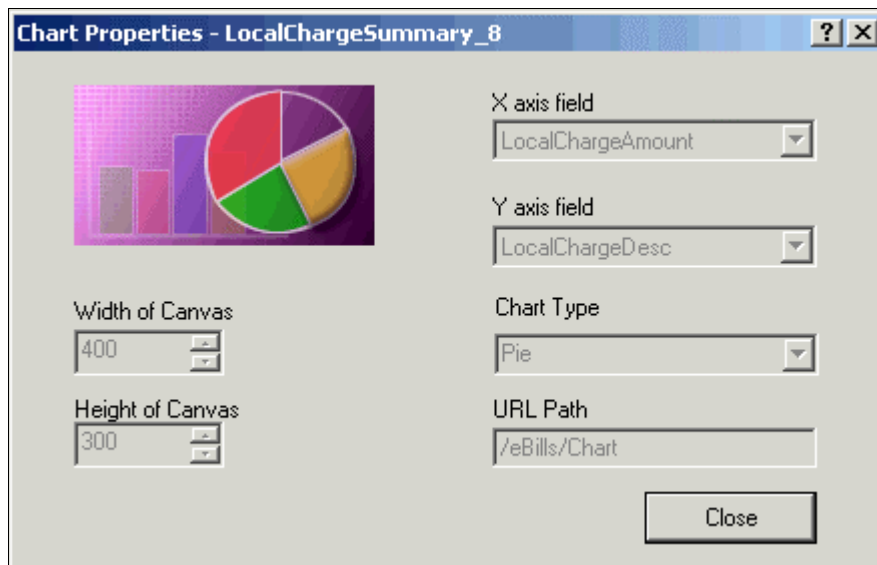
Deleting Charts

To delete a chart:

1. In the Application Tree click to open the Charts ![icon] , and right-click on the chart ![icon] you want to delete.

2. Select **Delete Chart** from the right-click menu.

3. A message appears asking you to confirm the delete operation.

4. Click **Yes**.

Appendix A: Data Definitions

EBNF Description

EBNF (Extended Backus-Naur Form) is a language described by sets of rules, syntax, tokens etc. Specify one or more syntactic groupings and give rules for constructing the language. For example, in C language, one kind of grouping is called an 'expression.' One rule for making an expression might be, "An expression can be made of a minus sign and another expression".

```
Expression ::= - Expression
```

Rules are often recursive, but there must be at least one rule that leads out of the recursion. EBNF's repetition is constructed so that an expression has only one derivation.

The most common formal system for presenting such rules that people can read is called EBNF — a context-free grammar.

Data Definitions

Data definitions supported by eaDirect for use with the EBNF expression language are:

- **String** — Delimit by double quotes.
  Example: "foo" , "00012", "508-467-9887"

- **Integer** — Format (+ | - |)(0-9)(0-9)*
  Grouping separator for thousands can be a DCHAR (. or ,)
  Example: 10, 35, 10034, -3462 etc.

- **Double/Float** — Format of (+ | - |)(0-9)(0-9)*. (0-9)(0-9)*
  This is a double precision value, i.e. a decimal value.
  Example: 0.0, -10.23, 10089.4658

- **Date** — Format where DD is the day, MM is the month, YY/YYYY is the year of format (0-9) and Mmm/ Mmmmm is the three-character month name and the full name of the month respectively.

- **Time** — HH stand for hours, MM for minute, and SS for seconds of format (0-9) [AM|PM].

- **Currency** — According to the following format:
  (+ | - | (|) (0-9)(0-9)*. (0-9)(0-9)* (- |) | CR | | DB |)
  In addition, the currency symbol can appear in the data.
  The currency format can change according to the location.

INT(*data*) — Convert the parameter passed as a string to an integer. The characters in the string must all be digits and spaces.

DOUBLE(*data, "decimal separator sign"*) — Convert the parameter passed as a string to double/float value. The characters in the string must all be digits, spaces plus the decimal separator sign.

CURRENCY(*data, "decimal separator sign", "negative format string"*) — Convert the parameter passed as a string to double/float value. The *negative format string* can be "CR", "DB" etc., and the *decimal separator sign* can be ".", "," etc. Data could also have other characters in the string, such as: "$ 1, 000, 000 . 00 CR".

Currency Example	Notes
4,123,978.999-	Trailing negative + three decimals
4,123,978.99-	Trailing negative
-4,123,978.99	Leading negative
-4,123,978.999	Leading negative + three decimals
4,123,978.99CR	Trailing CR (credit)
4,978.99 CR	Trailing CR separated by space
4,123,978.999CR	Three decimals, trailing CR (credit)
4,978.999 CR	Three decimals, trailing CR separated by space
4,123,978.99DB	Trailing DB (Debit)
4,978.99 DB	Trailing DB separated by space
4,123,978.999DB	Three decimals, trailing DB (Debit)
4,978.999 DB	Three decimals, trailing DB separated by space

DATE(*data, format_string*) — Convert the parameter passed as a string to Date value.

TIME(*data, format_string*) — Converts the parameter passed as a string to the Time value.

Date/Time Format String Components			
Format	**Description**	**English**	**French**
%%	Parse literal '%'		
%a	Day of week using the locale's weekday names (abbreviated name)	Fri	ven
%A	Day of week using the locale's weekday names (long name)	Friday	vendredi
%b	Month using the locale's month names (abbreviated name)	Jun	jun
%B	Month using the locale's month names (full name)	June	juin
%c	Locale-appropriate date and time representation (abbreviated form)	Fri Jun 08 17:53:28 2002	ven 08 jun 2002 17:50:58 EDT
%C	Locale-appropriate date and time representation (full form)		
%d	Day of month [1-31]; leading zero is permitted but not required	8 08	
%D	Date as %m/%d/%y.		
%e	Same as %d.		
%G	One digit of fractional seconds. Leading zero is permitted but not required. Fractional second value increments in six second intervals; e.g., .1=6 seconds, .7=42 seconds	.6 or 0.6 both equivalent to 0:36	
%h	Same as %b.	Jun	jun

Date/Time Format String Components			
Format	**Description**	**English**	**French**
%H	Hour (24-hour clock) [0-23]. Leading zero is permitted but not required.		
%I	Hour (12-hour clock) [1-12]. Leading zero is permitted but not required.		
%j	Day number of the year [1-366]. Leading zeros are permitted but not required.		
%K	Time duration as HHHH. Leading zeros are permitted but not required.	1324 0123	
%m	Month number [1-12]. Leading zero is permitted but not required.		
%M	Minute [0-59]. Leading zero is permitted but not required.		
%p	Locale's equivalent of either a.m. or p.m.	For U.S. locale: AM, PM, am, pm	
%Q	Time duration as MMM. Leading zeros are permitted but not required.	324 12 012	
%r	Appropriate time representation in the 12-hour clock format with %p.		
%R	Time as %H:%M.		

Date/Time Format String Components			
Format	**Description**	**English**	**French**
%S	Seconds [0-61]. Leading zero is permitted but not required. The range of values is [00-61] rather than [00-59] to allow for the occasional leap second and even more infrequent double leap second.		
%T	Time as %H:%M:%S.		
%w	Weekday as a decimal number [0-6], with 0 representing Sunday.		
%x	Locale-appropriate date representation.	06/08/02	08.06.2002
%X	Locale-appropriate time representation.	17:50:03	17:50:03
%y	The year within a century [00-99]. When a century is not otherwise specified, values in the range 69-99 refer to years in the twentieth century (1969 to 1999, inclusive). Values in the range 00-68 refer to years in the twenty-first century (2000 to 2068 inclusive). Leading zeros are permitted but not required.	02	02
%Y	Year including the century (for example 1993) [0001-9999].	2002	2002

Date String Formats		
Input Format	**Examples**	**Format String**
DDMMYYYY	31012002	`%d%m%Y`
YYYYMMDD	20021130	`%Y%m%d`
YYYY	2002	`%Y`
MMDD	1130	`%m%d`

YYYY-MM	2002-06
	2002/06
	2002.06
		`%Y-%m`
		`%Y/%m`
		`%Y.%m`

YYYY-MM-DD	2002-06-30
	2002/06/30
	2002.06.30
		`%Y-%m-%d`
		`%Y/%m/%d`
		`%Y.%m.%d`

DD.MM.YY	30.06.02	`%d.%m.%y`
DD.MM.YYYY	30.06.2002	`%d.%m.%Y`

DD-Mmm-YYYY	30-Jun-2002
	30.Jun.2002
	30 Jun 2002
		`%d-%b-%Y`
		`%d.%b.%Y`
		`%d %b %Y`

DD-Mmm-YY	30-Jun-02
	30.Jun.02
	30 Jun 02
		`%d-%b-%y`
		`%d.%b.%y`
		`%d %b %y`

DD Mmm Space between day and month	30 Jun	`%d %b`
DD/MM/YY	30/06/02	`%d/%m/%y`
DD/MM/YYYY	30/06/2002	`%d/%m/%Y`

Date String Formats		
Input Format	**Examples**	**Format String**
DD Mmmmm YYYY Includes full name of month, space between day and month, month and year	30 April 2002	`%d %B %Y`

Time String Formats		
Input Format	**Examples**	**Format String**
HHMM	2344	`%H%M`
HH:MM	23:44	`%H:%M`
HHMM PM Space before meridian indicator	1144 PM 1144 AM 1144 am	`%H%M %p`
HHMMPM	1144PM 1144AM 1144am	`%H%M%p`
HH:MM PM Space before meridian indicator	11:44 PM 11:44 AM 11:44 am	`%H:%M %p`
HH:MMPM	11:44PM 11:44AM 11:44am	`%H:%M%p`
HHMMSS	234405	`%H%M%S`
HH:MM:SS	03:44:05 22:10:36	`%H:%M:%S`
HHMMSS PM Space before meridian indicator	114405 PM 114405 pm 114405 am	`%H%M%S %p`
HHMMSSPM	114405PM 114405pm 114405am	`%H%M%S%p`
HH:MM:SS PM Space before meridian indicator	03:44:05 PM 03:44:05 pm 03:44:05 am	`%H:%M:%S %p`

Time String Formats		
Input Format	**Examples**	**Format String**
HH:MM:SSPM	03:44:05PM 03:44:05pm 03:44:05am	`%H:%M:%S%p`

Time Duration String Formats		
Input Format	**Examples**	**Format String**
HH:MM**.**D Duration of call in hours, minutes, and decimal minutes (0.1 = 6 seconds)	13:06.1	`%H:%M.%G`
MMM:SS	135:05	`%Q:%S`
HHHH:MM:SS	1233:05:17	`%K:%M:%S`

Appendix B: How Composer Assembles a Document

How Composer Assembles a Document

Composer assembles an HTML document using the following logic:

1. Identify whether there is a conditional template. If yes, determine which statement template to use for the specific customer. If no, use the default template.

2. Assemble the HTML for the selected template.

   Whenever an edocs tag `[E] or <edocsSection=NAME>` tag is encountered, the composing engine refers to the ALF for additional instructions.

   A set of tags, such as `[E]NameGroup,G[/E]`, indicates to the composing engine that a group template needs to be inserted in statement being generated. The composing engine uses the logic and formatting information in the group section of the ALF.

   When a table tag, `[E]Table,R[/E]`, is found, the ALF is referred to for formatting instructions to compose the table.

   When a field tag, `[E]Field,F[/E]`, is found, the DDF is referred to via the ALF to extract and present the correct data.

   When a chart tag, `[E]ChartTable,U[/E]`, is found, the ALF is referred to by the third party charting software for bitmap composition instructions.

   When the link tag, `[E]NotificationLink,L[/E]`, is found, the system looks at the application configurations in the Command Center to build a custom Link to a login page.

3. The group condition calls the HTML template specified to presned the group. During development, the Composer assembles that HTML in the places marked by **[E]** tags.

4. If the file includes nested groups, then the group section of the ALF is called again. Another condition calls the specified HTML template. The Composer assembles that HTML Template body tags for the inner group(s) in the places marked by **[E]** tags.

5. Assembly of the HTML document template is complete and the bill is presented to the customer.

Index

L

Line item
  creating conditions on in a table, 83
Links
  linking a filter element to a statement, 124
  mapping notification, 75
  navigational paging, inserting, 111
Logic elements
  creating, 41
  deleting, 51
  editing, 51
  syntax, 42
  viewing, 49

M

Mapping
  fields to the template, 35
  groups, 78
  notification link, 75
  tables and groups to the template, 77
  tables as charts, 147
  tables as text, 77
Menus
  Composer, 25

N

National Wireless Sample Application, 15
Navigational paging

links, inserting, 111
Non-data driven conditions on a table (pattern of rows), 94
Notification link
  mapping, 75
Null
  inserting, 88

O

Object Toolbar, 28
Occurrences of table or group, combining, 129

P

Page
  delete, 113
Page element
  creating, 109
  editing, 112
  removing, 113
Page set
  creating, 111
Paging, 108
  across table or group occurrences, 129
  links, inserting, 111
Pattern of table rows, creating conditions on, 94

R

Removing a page element, 113
Replacing a section with text or an image, 88