edocs®

What's New in eaDirect 3.2

eaDirect is a member of the eaSuite™ product line

V3.2
Document ID: DIGN-07-3.2-01
Data Published: 10.11.02

edocs, Inc., Two Apple Hill, 598 Worcester Road, Natick, MA 01760

Printed in the USA.

Table of Contents

New Features in eaDirect 3.2

1

Summary

eaDirect 3.2 enhances its B2B capabilities with the addition of the Hierarchy Console and sub-account indexing for improved access and reporting over organizational account structures. For the business user, these enhanced capabilities make account data easier to analyze and act on, consequently making account information more meaningful and more valuable. eaDirect 3.2 also lets you cut, copy, and paste DDF elements for additional efficiency in DefTool.

Note that the information in this document supplements the currently published guides for eaDirect and will be merged with these documents in the next full release of eaDirect.

Sub Account Indexing Feature

- Increased flexible optimization controls specifically for massive B2B data volumes

- Sub-account indexing speeds access through large data volumes directly to specific account sub-structures

- The Development Environment and Command Center updated for new configuration support

Benefits

- Increased control for configuring data sources in multiple application environments; configure applications for B2B or B2C data sources or mix configurations for consolidated applications

- Delivers additional reporting options for cross-invoice reporting, summary reports, and trending at the summary level

Hierarchy Console

- New Web console interface for creating and managing organizations
- Manage structures of accounts and users via a Web Console, create groupings of sites, departments or regions
- Pin-point users and accounts with search and return lists
- Includes user navigational access control in one console for easy management
- Supports upload and export of data for updates and reuse

Benefits

- Allows operations staff, customer service representatives or customer organizations themselves to create and manage their organizational hierarchy
- Visual and intuitive Web console supports a broad spectrum of skills, using a flexible and open format that can be customized
- Provides one Web console for a view of the organizational structure of accounts and users
- Supports custom mappings and attributes with full create, update and maintenance of mappings, useful for extensions like Device IDs and tracking

New Custom Post-Conversion Builder

- Development Environment updated with new formatting specification builder
- Default list of conversions still provided, but now its easy to add your own
- Create your new conversion routines just using the “conversion builder”
- Add, save and re-use conversion routines

Benefits

- Frees application developers to create their own format conversions, as needed
- Especially useful for internationalization of applications where presentation requirements vary widely

Sub-Account Indexing 2

About Sub-Account Indexing

Sub-account indexing feature enables you to:

- Quickly access sub-account data in a data input file for presentment in a view. (DefTool is the eaDirect GUI application you use to define the data extraction rules for your input file.)
- Include sub-account information in a history list, or “hit list.”

For example, you could define the cell phone detail sections of a large B2B telecommunications statement as sub-accounts, speeding Web access to line detail on request.

To enable the sub-account indexing feature, you must define the sub-account data area of an input file as a group, then define a group field on a piece of data that uniquely identifies the sub-account (or sub-document), called a subkey. Just as a primary key identifies a document, the subkey identifies each sub-document group.

You can also define additional group fields for indexing and presentment. eaDirect indexes the group fields that belong to the sub-account group as part of the regular Indexer production job, storing exact offsets and other information about the start and end of each sub-account in the database. This index data enables you to include sub-accounts in a history list and to click a link to quickly access sub-account data in a view (without having to scan the entire statement to find the group information).

You must create a separate view (DDF/ALF pair) for presenting sub-account information to users. You can also define more than one sub-account for an application; create one indexing DDF to use for all the sub-accounts and a separate ALF for each sub-account view.

The larger document that a sub-document belongs to is called the root, or parent, document. One parent document can consist of many instances of one or more sub-documents (sometimes referred to as the “children” of the parent document).

The following section describes the general process you must follow to set up sub-document indexing in your application.

New sample application to show sub-document indexing

eaDirect 3.2 includes a new sample application view to demonstrate sub-document indexing. The files are:

- *EDCSbd\samples\NW_Subdocument\NW_Subdocument.ddf*
- *EDCSbd\samples\NW_Subdocument\NW_Subdocument.alf*

Use the *NationalWireless.txt* data file with this application.

Setting Up Sub-Account Indexing

This section describes the general process required to implement sub-document indexing in a new or existing eaDirect application; each step is described in more detail in separate sections in this chapter. All of these instructions are supplemental to the *eaDirect User’s Guide*, which contains complete instructions on using DefTool and Composer to define an application.

You must tailor these procedures to the particular data in your input file and the views you need to present. Include these steps in the overall plan, design, and implementation of your eaDirect application.

Note that the procedure to implement sub-document indexing in an *existing* eaDirect application is different from creating the feature in a new application. For an existing eaDirect application, you can use your existing DDF, however you must create a separate ALF for presenting the sub-document view.

To implement sub-account indexing in a *new* eaDirect application:

1. Using DefTool, create a DDF specifically to index and extract data for the sub-account view. See “Creating a DDF for Indexing and Sub-Account Presentment“ below along with the *eaDirect User’s Guide* for details about using DefTool to create a new DDF and define the extraction rules described in Steps 1-5.

2. Define the doc style and the **primary key** in the indexing DDF, along with any other **pure fields you need to index** for your application. (You must index fields to make them available for inclusion in the history list.)

3. In this DDF, define the sub-document data as a **group**. (This includes creating group markers, creating the group, and assigning the group markers to a group.)

4. Identify a piece of data that uniquely identifies each sub-account in the data input file, such as sub-account number, and define this as the **subkey** group field for the sub-account. The subkey must be located within the sub-account group.

   eaDirect uses the subkey to locate the requested sub-document during dynamic retrieval. When the Indexer job indexes the subkey, it stores the exact offsets to each sub-document. By storing offsets for each sub-document, eaDirect has the information it needs to directly access any sub-document in the input file, eliminating the need to search the entire data file to identify the sub-document boundary.

5. Define any **group fields** you want to index, such as the name or other information associated with the sub-account number. Define any group fields that need to be indexed within the sub-account parent group.

6. In Composer, create a new Application Logic File (ALF), selecting the “Use sub-document” option to identify it as a sub-document view. Map the tables and fields to the ALF’s default or parent template, which is essentially the group template. (Composer is the eaDirect GUI application you use to define the presentation rules for displaying statements on the Web.) See “Creating an ALF for the Sub-Account View in Composer” in this chapter for details, along with the comprehensive instructions on using Composer in the *eaDirect User’s Guide*.

7. When you configure the Indexer job using the eaDirect Command Center, publish the indexing DDF with the Indexer job. eaDirect automatically indexes all group fields defined in the parent group of a sub-account in the DDF. See the *eaDirect Production Guide* for details about fields you need to index in your application and other Indexer job configuration details.

To implement sub-account indexing in an *existing* eaDirect application:

1. Make sure that you have defined the sub-account data as a **group** in your existing DDF.

2. Identify a piece of data that uniquely identifies the sub-account, such as a sub-account number. Right-click on the group name in the Definition Tree and select Edit from the right-click menu. On the Group Properties dialog, click Create. Define the field you want to use as a **subkey**.

3. On the Group Properties dialog, click Create again to define **any additional group fields** you want to index and/or include in the view, such as the name associated with the sub-account number. Define any group fields that need to be indexed within the sub-account parent group.

4. In Composer, you **must** define a new ALF for presenting the view of sub-account data, selecting the "Use sub-document" option to identify it as a sub-document view. Drag and drop (map) the tables and fields to the ALF's default or parent template, which is essentially the group template. See "Creating an ALF for the Sub-Account View in Composer" for details.

5. In the Command Center, publish the updated DDF.

6. Run the Indexer job. eaDirect automatically indexes all group fields defined in the parent group of the sub-account in the DDF.

Creating a DDF for Indexing and Sub-Account Presentment

Follow the procedures in this section if you are creating a *new* eaDirect application and need to implement sub-account indexing. In the DDF you create for sub-account indexing, you must define all the fields that you need to index for the application.

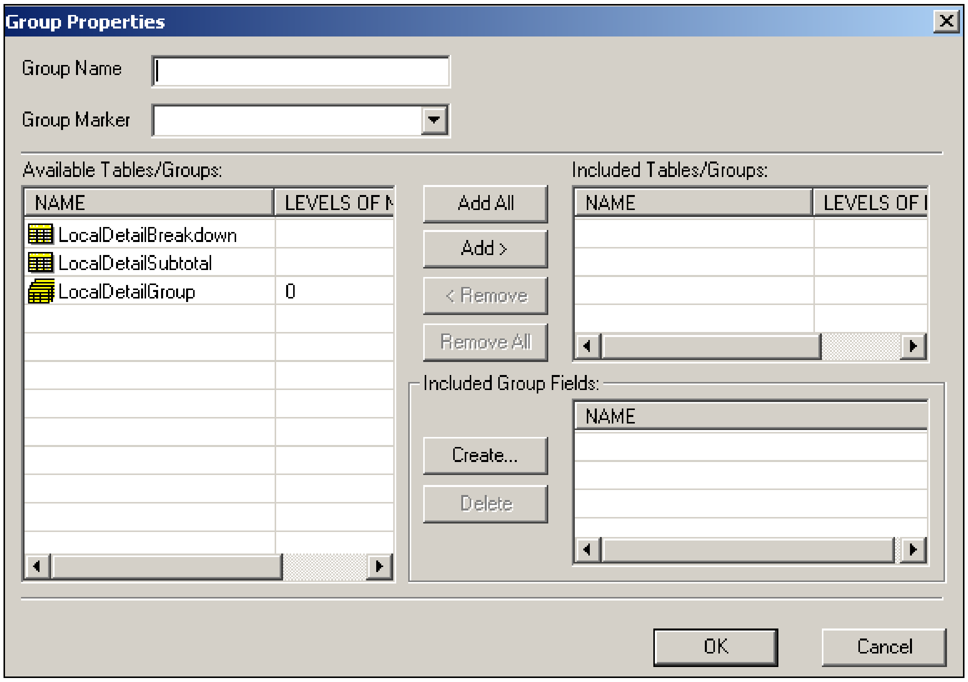
At each step using DefTool or Composer, refer to the *eaDirect User's Guide* for complete instructions; these are too detailed to include here.

To create a new DDF for indexing and sub-account presentment in DefTool:

1. In DefTool, select `New` from the File menu. DefTool begins the Data Configuration Wizard. Complete the wizard and follow the procedures described in the *eaDirect User's Guide* for creating a document style, primary key, and any additional page styles.

2. Click the New Marker icon and define the start and end markers for the sub-account group. You define the start and end markers for the group indicating the group offsets – the exact location in the data input file where eaDirect should begin searching for the group data. Once you define a group, it appears in the Definition Tree on the left of the DefTool window for the DDF.

3. Click the Group icon to display the Group Properties dialog. Define the properties of the sub-account group. Specify the group name, associated markers, and select the tables in the group.

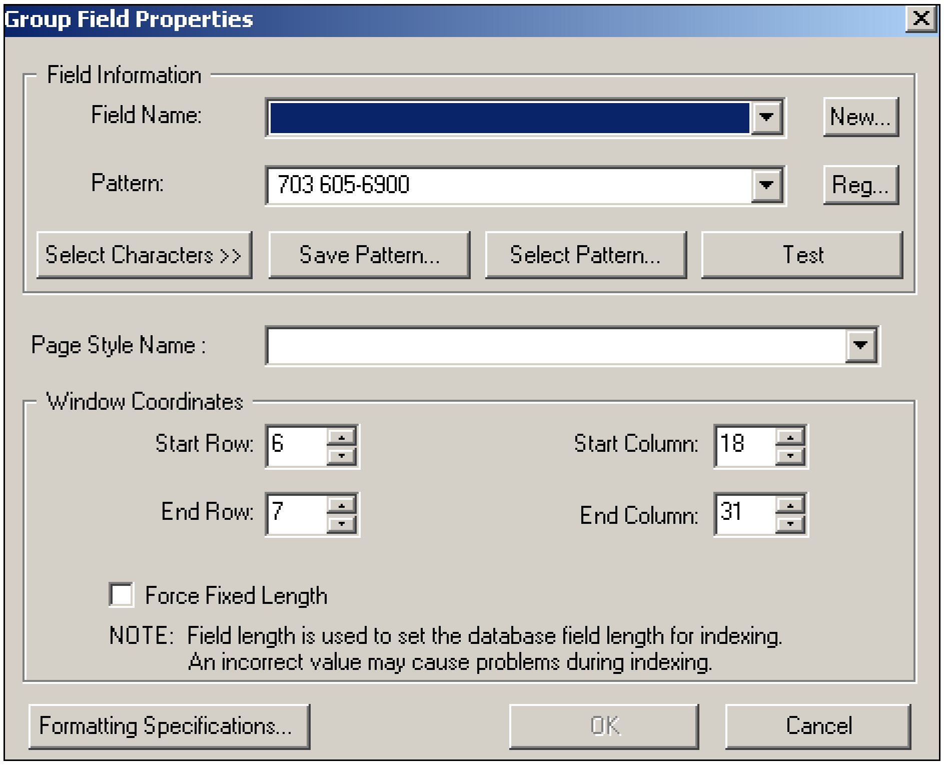


Available Tables/Groups:

NAME	LEVELS OF N
LocalDetailBreakdown	
LocalDetailSubtotal	
LocalDetailGroup	0

4. Create a subkey for the sub-documents: identify a piece of data in the data input file that uniquely identifies each sub-account. The subkey and any anchor you specify must be between the group's start and end markers.

5. On the Group Properties dialog, click **`Create`**. Use the left mouse button to drag the crosshair selector box around the data you want to use for the subkey group field. At the Select Your Choice dialog, select **`Field`** and click **`OK`**. DefTool displays the Group Field Properties dialog:



Group Field Properties

Field Information

Field Name:

Pattern: 703 605-6900

Page Style Name :

Window Coordinates

Start Row: 6

Start Column: 18

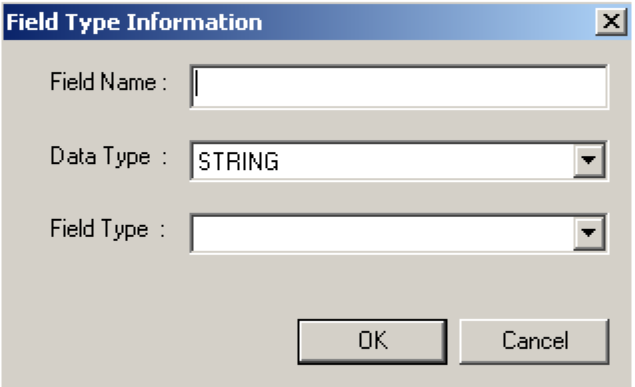
End Row: 7

End Column: 31

☐ Force Fixed Length

NOTE: Field length is used to set the database field length for indexing. An incorrect value may cause problems during indexing.

6. Click **New**. DefTool displays the Field Type Information dialog. Specify a name for the subkey field, select the data type, and select **Sub key** from the Field Type drop-down list.



Field Type Information

Field Name :

Data Type : STRING

Field Type :

7. Click **OK**. Continue to define the subkey group field extraction rules (properties) including the page style to associate with the group field, and click **OK** when you're done. DefTool displays the new subkey group field under the group and its associated markers in the Definition Tree; note that the subkey group field icon looks different from the regular group field icon.

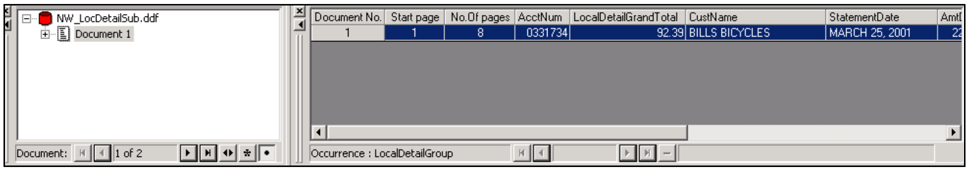
8. On the Group Properties dialog, define any additional group fields you want to index within the sub-account parent group. You can also define group fields in nested groups. You can define more than one group field for a group, but you must define only one as the subkey. You can use a group field in multiple locations.

9. Use the DefTool Simulator, described below, to test extraction of subkeys and group table data from your data input file. To edit data extraction rules, right-click a group or on a marker in the Definition Tree and select **Edit** from the right-click menu. To edit a subkey or other group field extraction rules, see "Modifying a group field or subkey " in this section.

10. Save the DDF.

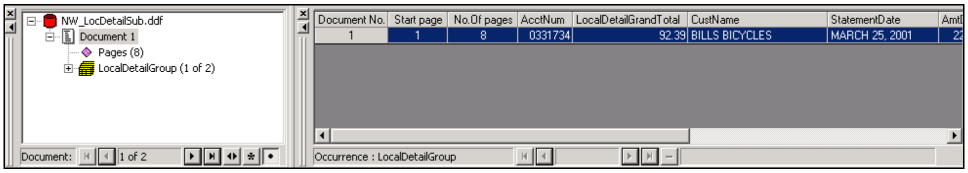
To simulate extraction of sub-account group data and subkeys in DefTool:

1. Click the Extractor Simulation icon in the Simulation toolbar or select **Extractor Simulation** from the Simulator menu. The Simulation pane appears at the bottom of the DefTool main window. Nodes shown in the tree on the left represent each document detected in the input file. On the right, Simulator displays the document boundaries and extracted fields.



Document No.	Start page	No.Of pages	AcctNum	LocalDetailGrandTotal	CustName	StatementDate	Amt[
1	1	8	0331734	92.39	BILLS BICYCLES	MARCH 25, 2001	22

2. Click the document to expand the list of pages, groups, and fields.



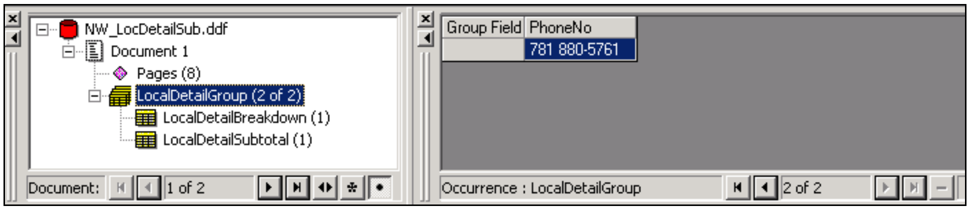
NW_LocDetailSub.ddf
Document 1
Pages (8)
LocalDetailGroup (1 of 2)

Document No.	Start page	No.Of pages	AcctNum	LocalDetailGrandTotal	CustName	StatementDate	Amt
1	1	8	0331734	92.39	BILLS BICYCLES	MARCH 25, 2001	22

Document: 1 of 2

Occurrence : LocalDetailGroup

3. Click the sub-document group to display the subkey or group fields for the first occurrence.



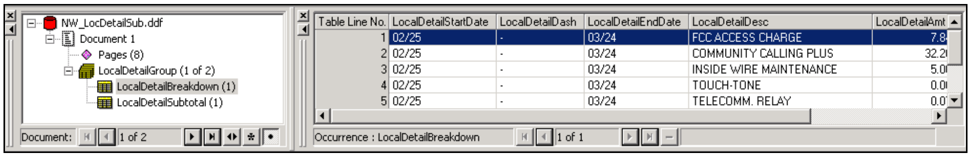
NW_LocDetailSub.ddf
Document 1
Pages (8)
LocalDetailGroup (2 of 2)
LocalDetailBreakdown (1)
LocalDetailSubtotal (1)

Group Field	PhoneNo
	781 880-5761

Document: 1 of 2

Occurrence : LocalDetailGroup 2 of 2

4. To display group data extraction, click the table.

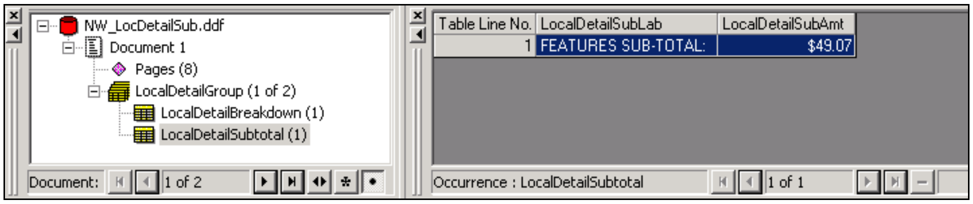


NW_LocDetailSub.ddf
Document 1
Pages (8)
LocalDetailGroup (1 of 2)
LocalDetailBreakdown (1)
LocalDetailSubtotal (1)

Table Line No.	LocalDetailStartDate	LocalDetailDash	LocalDetailEndDate	LocalDetailDesc	LocalDetailAmt
1	02/25	-	03/24	FCC ACCESS CHARGE	7.8
2	02/25	-	03/24	COMMUNITY CALLING PLUS	32.2
3	02/25	-	03/24	INSIDE WIRE MAINTENANCE	5.0
4	02/25	-	03/24	TOUCH-TONE	0.0
5	02/25	-	03/24	TELECOMM. RELAY	0.0

Document: 1 of 2

Occurrence : LocalDetailBreakdown 1 of 1



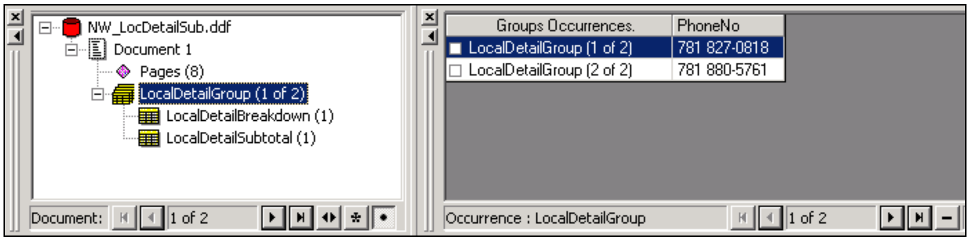
NW_LocDetailSub.ddf
Document 1
Pages (8)
LocalDetailGroup (1 of 2)
LocalDetailBreakdown (1)
LocalDetailSubtotal (1)

Table Line No.	LocalDetailSubLab	LocalDetailSubAmt
1	FEATURES SUB-TOTAL:	$49.07

Document: 1 of 2

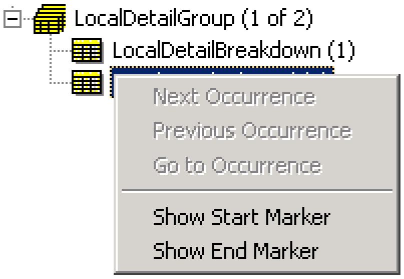
Occurrence : LocalDetailSubtotal 1 of 1

5. To test the extraction of all subkeys, click the Show All Tables icon, click the Document to expand the list of fields, tables, and groups, then click the sub-document data group. DefTool displays the extracted subkey and group fields for each group occurrence in the document.



Groups Occurrences.	PhoneNo
□ LocalDetailGroup (1 of 2)	781 827-0818
□ LocalDetailGroup (2 of 2)	781 880-5761

6. You can use all the existing features of the DefTool Simulator to test the groups, tables, and markers. For example, you can right-click on a group or table to display a menu of options:



7. As with any groups, tables, or fields, you can choose to selectively simulate a sub-document group (using the Simulator Settings icon or the `Simulator Settings` option from the Simulator menu, Group tab). By default, DefTool Simulates all tables, group fields, and subkeys associated with the group.

Modifying a group field or subkey properties

You can edit the properties of a group field, including a subkey, after you have created it. For example, the simulation process may reveal a problem that requires you to edit the group field extraction rules, such as the field pattern or window coordinates.

To edit the properties of a group field:

1. With the DDF open in DefTool, expand the Definition Tree to expose the group field.

2. Right-click on the group field (can be a subkey), and select **`Edit`** from the right-click menu. The Group Field Properties Dialog appears.

3. Make your edits using the Group Field Properties Dialog and click **`OK`**.

Deleting a group field or subkey

You can delete a group field, including subkeys, from a DDF. If a group field or subkey has multiple locations, you can select and delete by location (similar to how fields work).

To delete a group field or subkey:

1. With the DDF open in DefTool, expand the Definition Tree to expose the group field.

2. Right-click on the group field (can be a subkey), and select Delete from the right-click menu. DefTool asks if you're sure you want to delete the field.

3. Click **`Yes`**. DefTool deletes the group field.

To delete multiple group fields:

1. With the DDF open in DefTool, expand the Definition Tree to expose the group.

2. Right-click the group and select **`Edit`** from the right-click menu. DefTool displays the Group Properties dialog.

3. Under the Included Group Fields, select a field to delete and click **`Delete`**. DefTool asks if you're sure you want to delete the field.

4. Click **`Yes`** to delete the field. DefTool deletes the group field.

5. Continue deleting group fields as necessary.

6. Click **`OK`**.

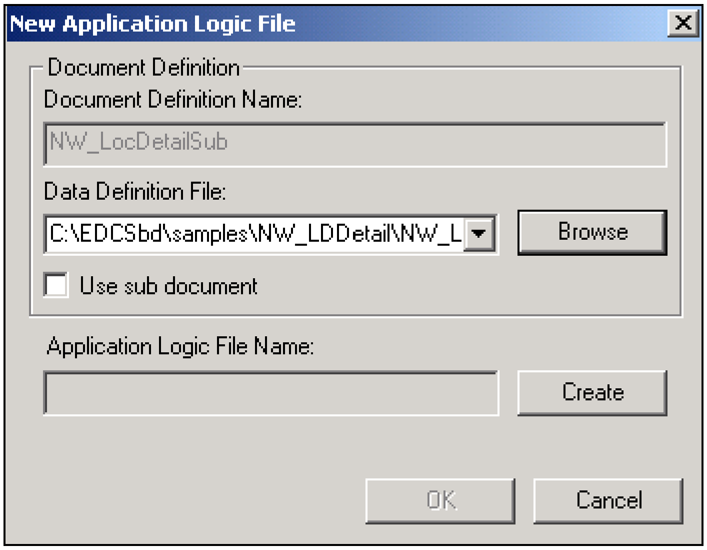
Creating an ALF for the Sub-Account View in Composer

To use the sub-document indexing feature you **must** create and use a separate Application Logic File (ALF) for presenting the sub-document view. This is true for both new eaDirect applications and for adding the feature to an existing application.

You must 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 New ALF from the File menu. Composer displays the New Application Logic File dialog. Select the application DDF you created for this view.



New Application Logic File

Document Definition

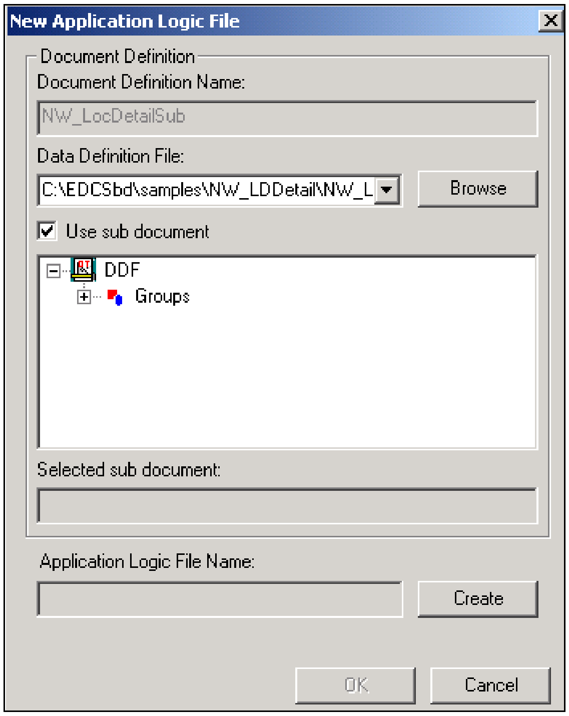
Document Definition Name: NW_LocDetailSub

Data Definition File: C:\EDCSbd\samples\NW_LDDetail\NW_L

☐ Use sub document

Application Logic File Name:

2. Select the **`Use sub-document`** option. Composer displays a tree showing the sub-document groups in the DDF.



Document Definition Name: NW_LocDetailSub

Data Definition File: C:\EDCSbd\samples\NW_LDDetail\NW_L

☑ Use sub document

Selected sub document:

Application Logic File Name:

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.

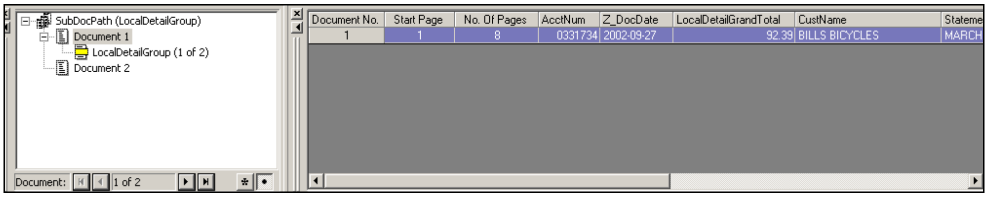
4. Click **`Create`** and specify a name for the ALF, then click **`Save`**. Click **`OK`**. Note that in the Definition Tree, Composer converts the sub-document group into pure tables, fields into pure fields, and the subkey to the primary key. Composer treats the sub-document as a normal document, not a group (and there is no group template for the group).

5. In the Application Tree, right-click the **`Home`** node and select **`Add Default Template`** from the right-click menu. The New Default Template dialog appears. Select the template you to use as the default, and click **`OK`**. Composer opens the selected template.

6. Drag-and-drop the tables and groups to the template as you want them to appear in the view.

7. Click 💾 to save the ALF.

8. You can use the Composer Simulator to verify that the composed document appears as you intend. Expand the document node in the left pane to display the sub-documents.



Document No.	Start Page	No. Of Pages	AcctNum	Z_DocDate	LocalDetailGrandTotal	CustName	Stateme
1	1	8	0331734	2002-09-27	92.39	BILLS BICYCLES	MARCH

To delete a group field or subkey from a template:

- With the ALF open in Composer (WYSIWYG view), select the group field or subkey E tag, such as [E]PhoneNo,F[/E], and delete it. (This is how you currently delete tables and fields.)

Applying sorting, filtering, and other features within sub-accounts

You can apply the following features within sub-documents in much the same way you apply them to regular (parent) documents. You can apply:

- **Sorting, filtering, arithmetic subtotaling, combine, and paging elements** to tables or groups within a sub-document, but not on a group field.

- **Dynamic patterns** to pass patterns to any child group markers within the sub-document group, but not to a group field.

- **Links** to a group field, including a subkey.

- **Conditional logic** on tables and groups within the sub-document, but not on a group field.

Purging Sub-Account Index Data

eaDirect purges sub-document index data from the database when purging the data for its parent, or root document.

Use the Purge Apps Command Center job to purge data. If you have sub-document data, the PurgeIndexSummary task of the Purge Apps job automatically purges the related sub-document data. See the *eaDirect Production Guide* for more information about purging data.

APIs for Sub-Account Indexing

eaDirect 3.2 includes new signatures of APIs in the Content Access and Verify SDK packages that enable you to add sub-account indexing to an eaDirect application that has customized content access and/or uses the custom index volume verification feature.

The new APIs use one of the following parameters (instead of a string):

```
Name[] accounts
```

```
Name account
```

`Name` represents a generic name, in this case the hierarchical sub-document account path; `account` or `accounts` represent the sub-account number.

See the Javadoc included with 3.2 eaDirect for details on using these APIs to include sub-document indexing with this custom functionality.

com.edocs.app.user

The com.edocs.app.user SDK lets you customize content access and history lists in your eaDirect application.

The `User` class is the content access interface to the eaDirect core. Its methods retrieve and send statement data for a given user account, as well as sorting, subtotaling, and updating optional fields at presentment.

Use the new APIs listed below to modify your application to add access to sub-documents and include sub-account data in a history list. These new APIs use the new parameter shown for sub-documents:

SendDocument

```
Name[] accounts
```

GetSummary

```
Name account
```

UpdateSummaryInfo

```
Name account
```

getDocumentReader

```
Name[] accounts
```

GetDocumentInputStream

```
Name[] accounts
```

com.edocs.app.verify

The `com.edocs.app.verify` SDK lets you implement the custom Audit to Verify feature in your eaDirect application. This feature let users audit indexed data volumes before releasing them for presentment.

This `Verify` class is the interface to eaDirect core for managing index files (or volumes). Use the new Verify APIs listed below to modify your application to include sub-document indexing with this feature. These APIs use the new parameter shown for sub-documents:

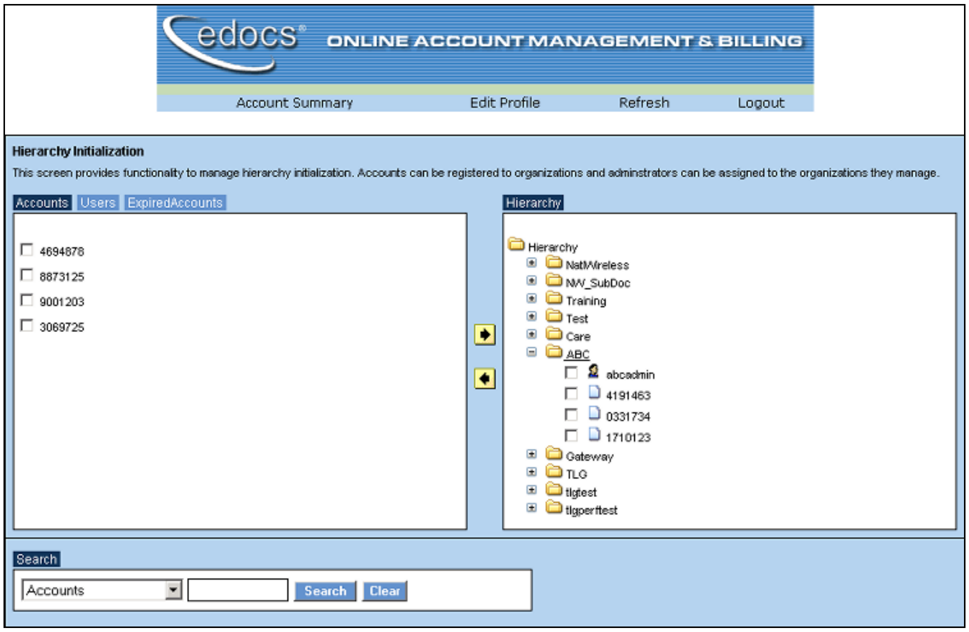
getHitList

```
Name account
```

Managing B2B Hierarchies 3

About the Hierarchy Console and B2B Hierarchies

The Hierarchy Console provides an intuitive user interface for administrators to organize and edit account information. Its flexible format is open for customization and import/export aware through edocs Common Directory Access (CDA). Use the Hierarchy Console to map primary and secondary keys into account groups that can control access privileges, statement viewing, editing user profiles, and other online account management.



Hierarchy Initialization

This screen provides functionalty to manage hierarchy initialization. Accounts can be registered to organizations and adminstrators can be assigned to the organizations they manage.

The Hierarchy Console web application (EAR file) extends the eaDirect user management framework (UMF) to support a flexible B2B hierarchy. The Hierarchy web application and its APIs support:

- Large and complex account statements
- B2B ways of organizing account information (detail records, departments, cost centers, documents and subdocuments) not contained in the data source
- Changes to organization structures, cost centers and divisions that are charged back through internal A/R processes for billed services. These changes need not be present in the data source, for example, when changes result from acquisition, restructuring, or consolidation.

Direct billing to customers (B2C) usually involves simple relationships between statements and users. One customer has one account number where they log in to view one statement each month. Business-to-Business companies (B2B) generally have much more complex relationships with their customers.

B2B customers may have one or many users looking at one or many statements or pieces of statements. These users will also have different privileges for what content in the statements they can view, and what actions they may take on that content, for example to approve or pay it.

The **Hierarchy Console** lets you define and describe these users, privileges, and actions through a simple web console interface. For example, you can use the hierarchy Console to design and apply a **B2B hierarchy** of cost centers. The eaSuite can then organize, extract, and display a company phone bill by cost center, so that each manager could view, approve, and pay only their department's charges. This process of organizing and displaying subdocuments is sometimes called “bursting” an invoice or statement.

Tip

The Hierarchy Console does not perform any calculations directly. Rather, it passes data and metadata to other components that calculate statement data for presentment.

The Hierarchy Console itself does not control access to statements. An end user would not view an eaDirect online statement through the console. Rather, an administrator can assign a user or account to a folder in the hierarchy, so that anyone with access privileges to that folder may use eaDirect to view statements for users in that folder.

Setting Up the Hierarchy Console

This procedure assumes you have installed eaDirect 3.2 and the web application archive **`ear-hierarchySample.ear`** .

To set up the Hierarchy Console:

1. Create the database schema with the script create_hierarchy_schema. This creates the root node o=Hierarchy.
2. Deploy the sample EAR file ear-hierarchySample.ear.
3. Index the HierarchySample data by creating an eaDirect application and running the Indexer job on the sample data file.
4. Navigate to the Hierarchy Console at your application server and port, for example: http://plato:7001/hierarchySample/user/jsp/index.jsp

Caution The *o=Hierarchy* name is the *context root* of the hierarchySample web application. You cannot change this name without making a deployment descriptor change to ejb-enrollment-hm.jar. In META-INF/ejb-jar.xml, change the contextRoot value to your new context root. For more information on customizing application context, see the SDK **User Management Framework.**

5. Enroll an administrative user as a SuperAdmin with root level privileges.

For details on creating folders and enrolling users, see Working with Hierarchies.

Creating The Database Schema

In this step, you set your classpath and run the CDA client to create the database schema for the Hierarchy Console. This procedure assumes you have created an Oracle database with **`edx_admin.sh`** as part of your eaDirect for Solaris installation.

You need classpath and password information from your database installation for this procedure. Write down your information in this table, and contact your system administrator if you need help.

Database Value	Default/Example	Your Info
Login user	oracle	
Login password	edocs	
EDX_HOME	/opt/EDCSbd/lib/edx_common.jar	
WL_HOME	opt/BEA/wlserver6.1/lib/weblogic.jar	
JAVA_HOME	/export/home/bea/jdk131	
Server name	dusky	
Database port	1521	
Database instance	edx0	
DBA name	edx_dba	
DBA password	edx	

To create the default CDA database schema for the hierarchy Console:

1. Log in as the oracle user and set your CLASSPATH to include your edocs, application server, and Java home directories. For example:

   ```
   CLASSPATH=<EDX_HOME>/lib/edx_common.jar:/<WL_HOME>/lib/w
   eblogic.jar:/$WL_HOME /jdk131
   ```

Tip — If your JDK is set elsewhere in your environment, you may not need to include it in your classpath. If you receive errors while running the script, set your classpath to include your JDK.

2. Export CLASSPATH.

3. Navigate to the sample directory for `create_hierarchy_schema` at `<EDX_HOME>/samples/hierarchySample/schema`

4. To start the CDA client and create the default database schema for the hierarchy, edit this command for your own environment with the values for your database server name and port and oracle username, password, and database instance.

    ```
    java -Djdbc.drivers=oracle.jdbc.driver.OracleDriver
    com.edocs.jndi.cda.cli.Main
    jdbc:oracle:thin:@dusky:1521:edx0 edx_dba edx <
    create_hierarchy_schema
    ```

5. Run your edited command. If the script executes successfully, you will see a series of caret prompts, like this: >>>>>>>>>>

6. Stop and start your application server to pick up the changes to the CDA schema.

7. If you run the **`create_hierarchy_schema`** script again, or if you run another database script such as **`create_training_schema`**, you may need to remove or comment out any schema binding conflicts. For example, both these scripts create an **`sb role`** attribute, and the second script will throw a **`SchemaBind`** exception.

Caution The application server caches schema information, so that you must stop and start your application server to pick up the changes to the database. If you receive errors enrolling a SuperAdmin, be sure to stop and start the app server before enrolling users.

By default, the **`create_hierarchy_schema`** script prepopulates the Hierarchy Console with a root level organization “Hierarchy.” You then use the Hierarchy Console to create your own hierarchy subfolders under this root level. You can then map account and user information from subdocuments into your hierarchy to group and organize accounts.

You must first create a root-level hierarchy folder (or run **`create_hierarchy_schema`**) and enroll at least one SuperAdmin user with root level privileges before you can assign users and accounts to folders. For more information, see Creating Hierarchy folders.

Deploying The Hierarchy Web Application

Start your application server if it is not already running, and deploy the sample EAR file

```
/opt/EDCSbd/samples/hierarchySample/J2EEApps/weblogic/ea
r-hierarchySample.ear
```

Indexing Hierarchy Sample Data

Your eaDirect installation places a sample DDF, ALF, and dataset for the Hierarchy Console in

```
<EDX_HOME>/samples/hierarchySample/NW_Subdoc
```

Use the files in this folder to create an eaDirect application and run the Indexer job.

The Indexer job indexes all **subdocuments** defined in your DDF. For more information, see **Subdocument Indexing**.

Working with Hierarchies

SuperAdmin and Admin Roles and Privileges

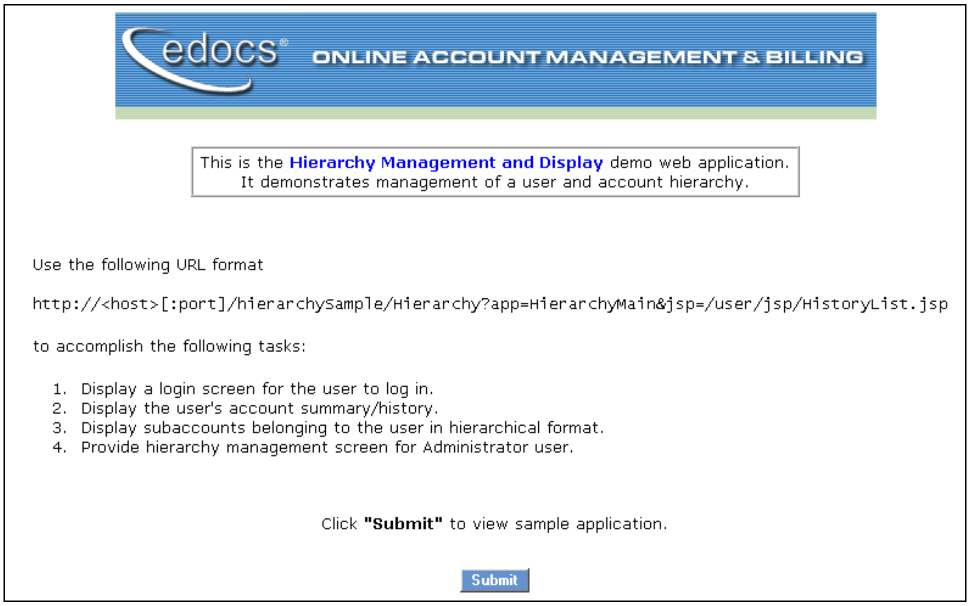
A **SuperAdmin** *registers* **primary accounts** with an **organization**. When a SuperAdmin logs in, the left pane of the Hierarchy Console presents a list of *unregistered* accounts containing all primary accounts for the DDN, for example IBM. These primary accounts can then be associated with an organization within the hierarchy, for example Northeast or Central.

An Admin *assigns* **subaccounts** into one or more organizations within the hierarchy. When an Admin logs in, the left pane of the Hierarchy Console presents a list of *unassigned* subaccounts for the Admin's organization(s), from all primary accounts that the SuperAdmin has registered with that organization. For example, these subaccounts could be the telephone numbers of IBM employees in the Northeast, which the Admin could then assign to the Boston or New York organizations.

Logging In to the Hierarchy Console

To bring up the Hierarchy login screen, use the following URL format for your own host and port:

`http://<host>[:port]/hierarchySample/user/jsp/index.jsp`



This is the **Hierarchy Management and Display** demo web application.
It demonstrates management of a user and account hierarchy.

Use the following URL format

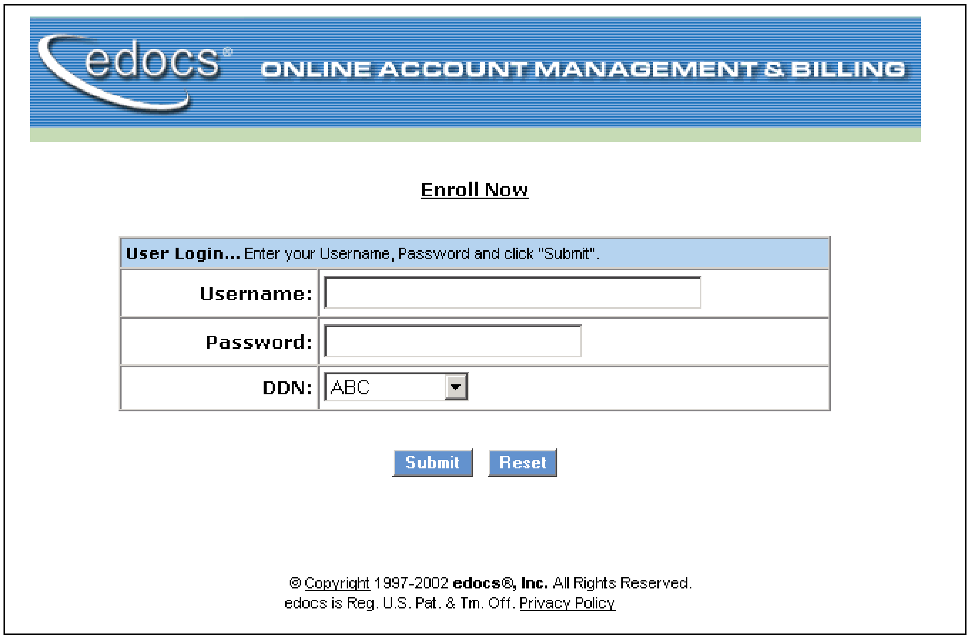
```
http://<host>[:port]/hierarchySample/Hierarchy?app=HierarchyMain&jsp=/user/jsp/HistoryList.jsp
```

to accomplish the following tasks:

1. Display a login screen for the user to log in.
2. Display the user's account summary/history.
3. Display subaccounts belonging to the user in hierarchical format.
4. Provide hierarchy management screen for Administrator user.

Click **"Submit"** to view sample application.

Clicking the Submit button brings you to the login screen.



User Login... Enter your Username, Password and click "Submit".

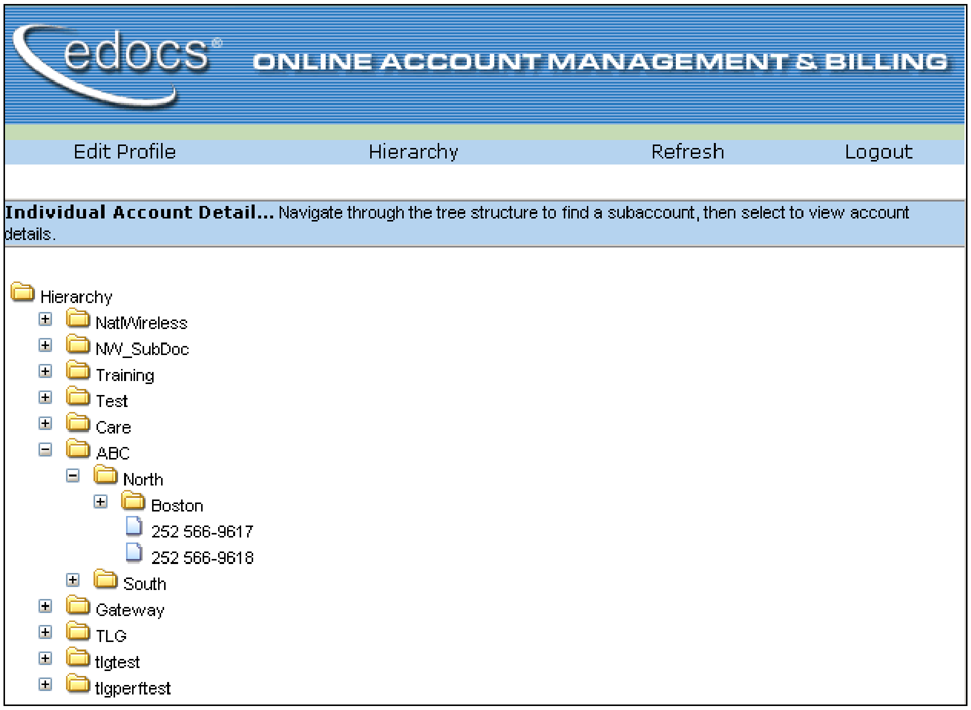
Username:	
Password:	
DDN:	ABC

© Copyright 1997-2002 **edocs®, Inc.** All Rights Reserved.
edocs is Reg. U.S. Pat. & Tm. Off. Privacy Policy

From this screen, you can log in as an existing user or enroll a new user. For enrollment, see Enrolling Users in the Hierarchy.

When a user logs in, she sees her assigned level in the hierarchy and may navigate down through any subtrees to view any accounts for which she has privileges.

For example, a **SuperAdmin** user has privileges to view all levels in the hierarchy. In this example, the SuperAdmin user **su** has logged in to the Hierarchy Console and can see the root level folder **Hierarchy** and all folders below it.

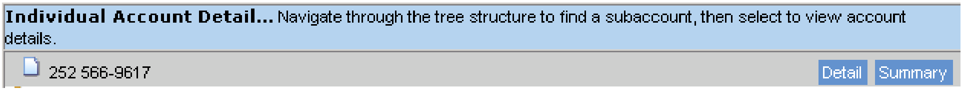


Individual Account Detail... Navigate through the tree structure to find a subaccount, then select to view account details.

Hierarchy
- NatWireless
- NW_SubDoc
- Training
- Test
- Care
- ABC
  - North
    - Boston
    - 252 566-9617
    - 252 566-9618
  - South
- Gateway
- TLG
- tlgtest
- tlgperftest

You must enroll at least one SuperAdmin user before enrolling users at lower levels in the hierarchy. For more information, see Enrolling Users in the Hierarchy.

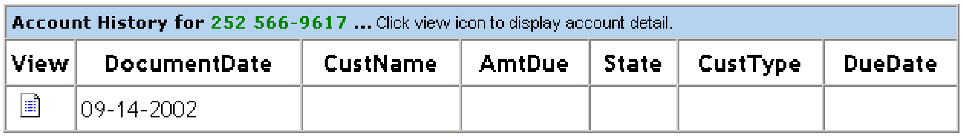
Select an account from the tree. It appears as a highlighted bar at the top of the Console pane.



Individual Account Detail... Navigate through the tree structure to find a subaccount, then select to view account details.

252 566-9617

Click the **Summary** button to view account data for this account.



Account History for 252 566-9617 ... Click view icon to display account detail.

View	DocumentDate	CustName	AmtDue	State	CustType	DueDate
🗎	09-14-2002					

From the login screen, click **Hierarchy** in the navigation bar to go to the Hierarchy Console itself.

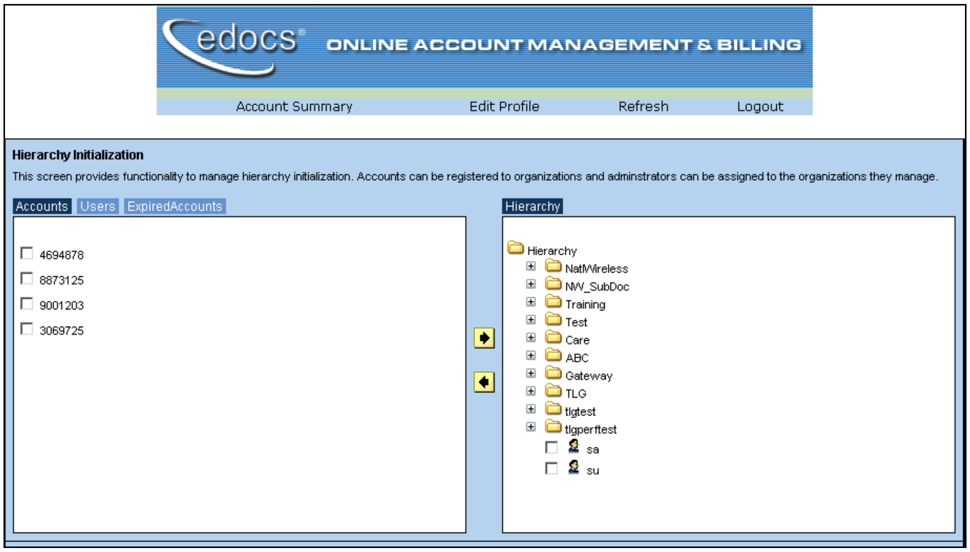
Navigating the Hierarchy Console

The Hierarchy Console has two panes, each containing a directory tree of nested folders. The left pane has three tabs: **Accounts, Users**, and **ExpiredAccounts**.

Caution

The **ExpiredAccounts** tab is visible only to Admin users and not SuperAdmins or Users. *Only unassigned accounts may be expired.*

Click the **Accounts** tab to show all accounts not assigned to a folder in the hierarchy. To assign accounts, see Assigning accounts into hierarchy folders.



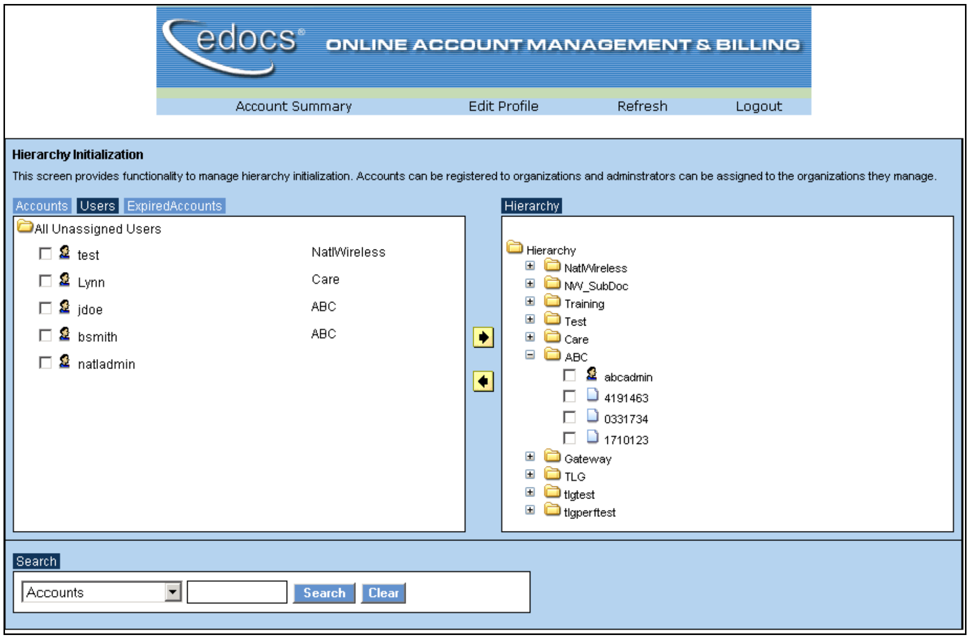
Hierarchy Initialization

This screen provides functionality to manage hierarchy initialization. Accounts can be registered to organizations and adminstrators can be assigned to the organizations they manage.

☐ 4694878
☐ 8873125
☐ 9001203
☐ 3069725

Hierarchy
- NatlWireless
- NW_SubDoc
- Training
- Test
- Care
- ABC
- Gateway
- TLG
- tlgtest
- tlgperftest
- ☐ sa
- ☐ su

Click the **Users** tab to show all users (including any Admins and SuperAdmins) not assigned to a folder in the hierarchy. To assign users, see Assigning users into hierarchy folders.



Hierarchy Initialization

This screen provides functionality to manage hierarchy initialization. Accounts can be registered to organizations and adminstrators can be assigned to the organizations they manage.

In the right pane, the **Hierarchy** tree presents a directory of **nodes** in the hierarchy as folders. Clicking a folder expands it to show all accounts and users assigned to that node in the hierarchy.

Creating Hierarchy Folders

An Admin or SuperAdmin may create folders at any level in the hierarchy for which they have privileges. A SuperAdmin must create at least a top-level folder for each organizational hierarchy before anyone can assign users or accounts.

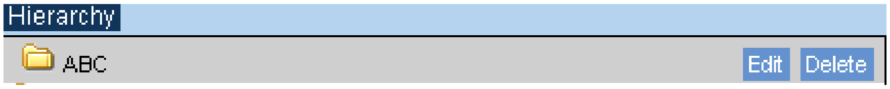
To create a folder in the Hierarchy Console:

1. Bring up the hierarchy console and log in with your username, password, and DDN to go to the login page.

2. Create a root organization as a top-level folder. Only a SuperAdmin may create or access this root folder.

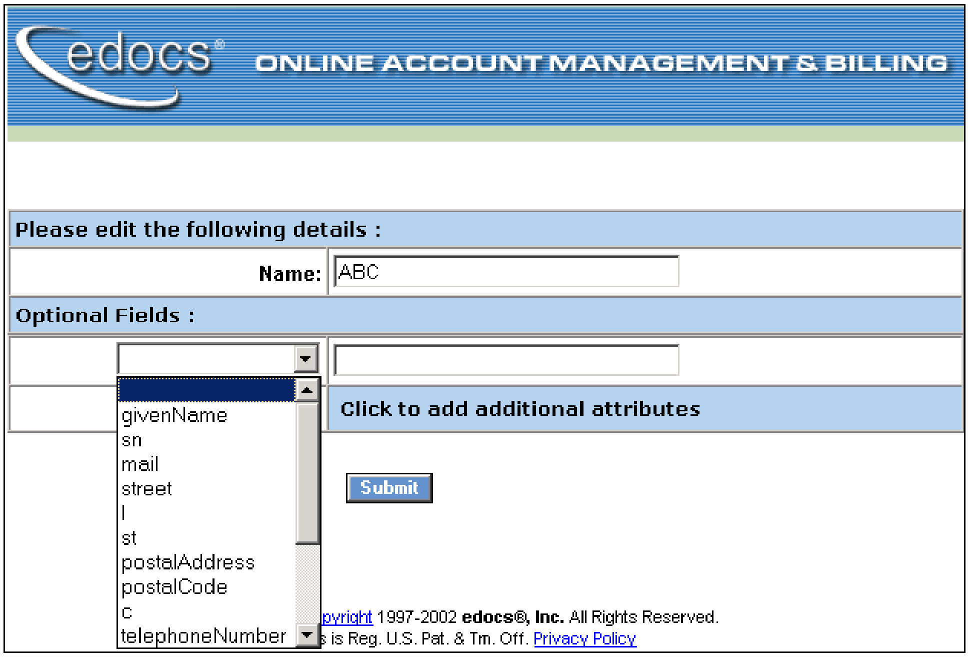
3. Within that folder, you may then create subfolders for your Organizational Units, (ex. ABC, Gateway).

Clicking a folder highlights it at the top of the page with **Edit** and **Delete** buttons.



Editing Hierarchy Folders

Highlight the folder and click **Edit**. You can add additional attributes to a folder through this screen.



Please edit the following details :

Name: ABC

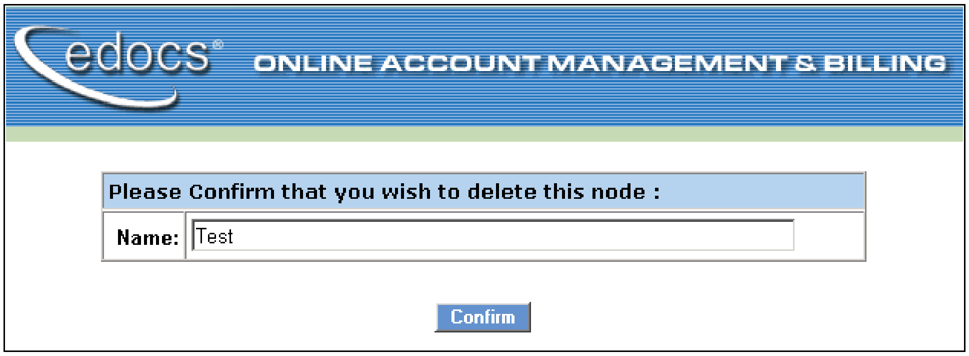
Optional Fields :

pyright 1997-2002 **edocs®, Inc.** All Rights Reserved.
s is Reg. U.S. Pat. & Tm. Off. Privacy Policy

Deleting Hierarchy Folders

An Admin or SuperAdmin may delete folders at any level in the hierarchy for which they have privileges. You may not delete a folder containing accounts or subfolders (children or leaf nodes). Before you can delete a folder, you must unassign all accounts and delete any subfolders.

Highlight the folder and click **Delete**. A confirmation screen appears. Click **Confirm** to delete the folder.

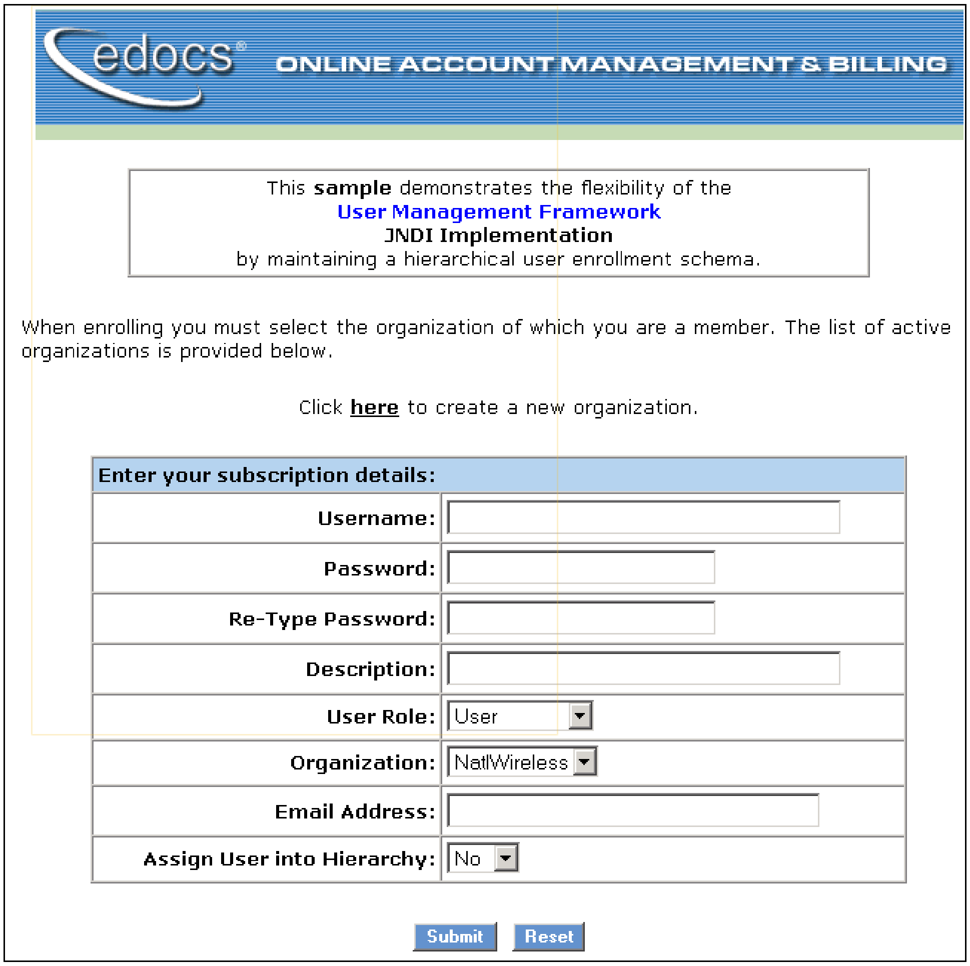


Please Confirm that you wish to delete this node :

Name: Test

Enrolling Users in the Hierarchy

From the login screen, click **Enroll Now** to enroll a new user. The enrollment page appears.



This **sample** demonstrates the flexibility of the
User Management Framework
JNDI Implementation
by maintaining a hierarchical user enrollment schema.

When enrolling you must select the organization of which you are a member. The list of active organizations is provided below.

Click **here** to create a new organization.

Enter your subscription details:	
Username:	
Password:	
Re-Type Password:	
Description:	
User Role:	User
Organization:	NatlWireless
Email Address:	
Assign User into Hierarchy:	No

This table shows the allowable values for each enrollment field.

Field Name	**Allowable Values**	**Status**	**Notes**
Username	User-defined	Required	Enter the username you wish to create for this user. This is the primary account key.

Field Name	Allowable Values	Status	Notes
Password	User-defined	Required	Enter the password you wish to create for this user. Retype the password for security.
Description	User-defined	Optional	Enter an optional description of this user, for example their title or name.
User Role	SuperAdmin Admin User *(Names are for this sample and may be customized in a custom schema)*	Required	Select the level of the hierarchy you want this user to see. A **SuperAdmin** can see all folders from the root level. An **Admin** can see all folders at and below their assigned folder. A **User** can see only their own account information.
Organization	Hierarchy North South IBM Coke	Required	Select your DDN or use the **Click Here** link to create a new organization.
Email Address	User-defined	Optional	Enter the e-mail address for this user.

Field Name	Allowable Values	Status	Notes
Assign User into Hierarchy	Yes/No	Required	**Yes** automatically adds this user to the organization (folder) you define. This allows this user to see the Hierarchy tab in the Console at the folder level you define on enrollment. **No** leaves this user unassigned. Later, a SuperAdmin or Admin for this hierarchy can assign unassociated users to a folder.

Assigning Accounts Into Hierarchy Folders

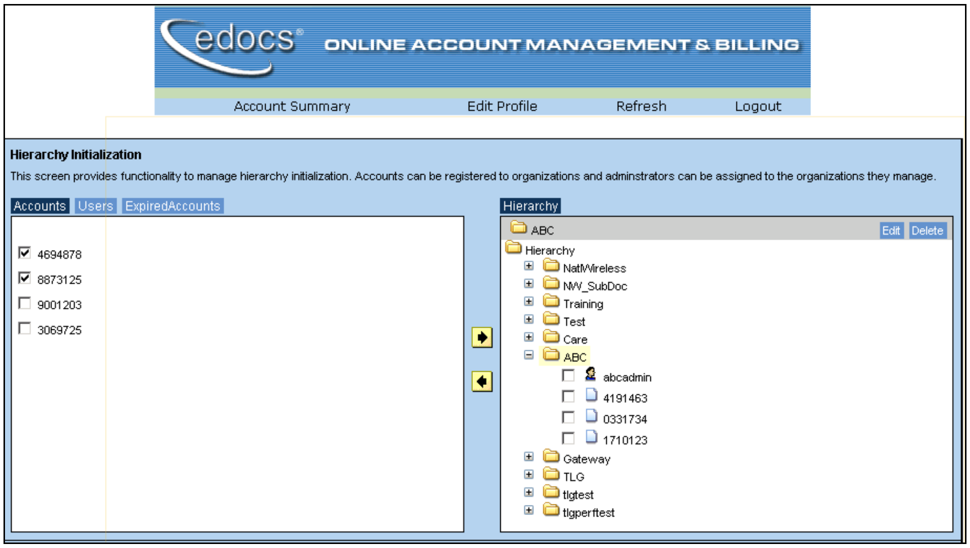
Admins and SuperAdmins may assign some or all accounts to a folder at any level of the hierarchy *except the context root.* A folder may contain multiple accounts, but an account may be assigned to only one folder.

Admins and SuperAdmins may also remove an account from a folder. This does not remove the account from the system, but simply means the account is no longer associated with that folder. The dissociated account reappears in the list of unassigned accounts.

To assign an account to a folder in the hierarchy:

1. Log in as the SuperAdmin or Admin and click the Hierarchy link. The Hierarchy Console displays all folders for which you have privileges.

2. Click the **Accounts** tab in the left pane to see a list of unassigned accounts.



Hierarchy Initialization

This screen provides functionality to manage hierarchy initialization. Accounts can be registered to organizations and adminstrators can be assigned to the organizations they manage.

☑ 4694878
☑ 8873125
□ 9001203
□ 3069725

- Hierarchy
  - NatlWireless
  - NW_SubDoc
  - Training
  - Test
  - Care
  - ABC
    - □ abcadmin
    - □ 4191463
    - □ 0331734
    - □ 1710123
  - Gateway
  - TLG
  - tlgtest
  - tlgperftest

3. In the left pane, select the checkbox for the account(s) you want to assign.

4. In the right pane, select the folder to which you are assigning them. The folder will highlight as shown.

5. Click the **right arrow**. The account(s) will appear within the assigned folder.

Assigning Users Into Hierarchy Folders

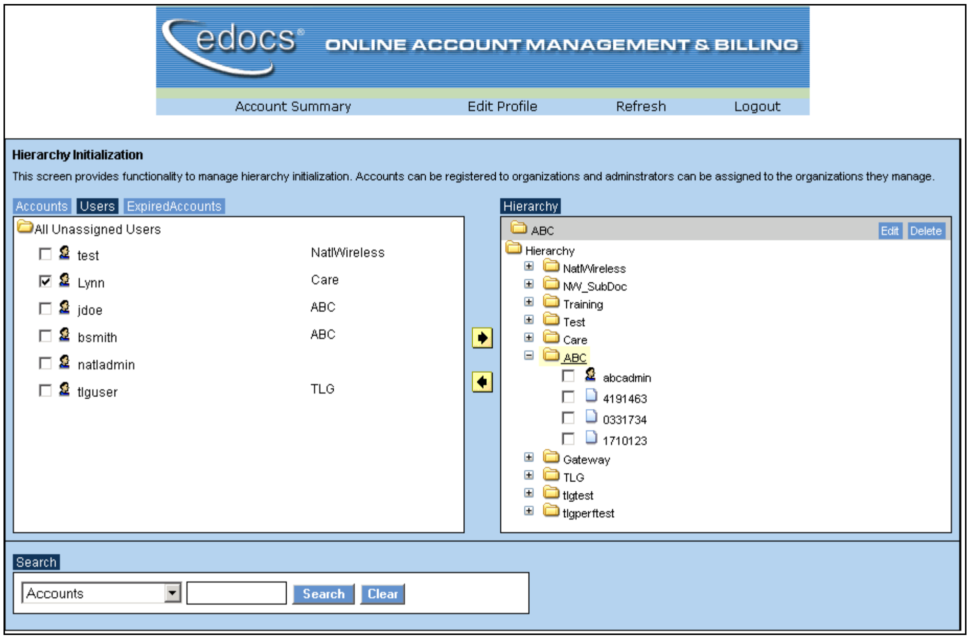
Admins and SuperAdmins may assign users to a folder at any level of the hierarchy, or assign a user directly to a single account. A folder may contain multiple users, but each user may be assigned to only one folder.

Admins and SuperAdmins may also remove a user from a folder. This does not remove the user from the system, but simply means the user is no longer associated with that folder.

To assign a user to a folder in the hierarchy:

1. Log in as the SuperAdmin or Admin and click the Hierarchy link to go to the Hierarchy Console. You will see all folders for which you have privileges.

2. Click the **Users** tab in the left pane to see a list of unassigned users.



Hierarchy Initialization

This screen provides functionality to manage hierarchy initialization. Accounts can be registered to organizations and adminstrators can be assigned to the organizations they manage.

All Unassigned Users

□ test	NatlWireless
☑ Lynn	Care
□ jdoe	ABC
□ bsmith	ABC
□ natladmin	
□ tlguser	TLG

ABC

Hierarchy
- NatlWireless
- NW_SubDoc
- Training
- Test
- Care
- ABC
  - □ abcadmin
  - □ 4191463
  - □ 0331734
  - □ 1710123
- Gateway
- TLG
- tlgtest
- tlgperftest

Search: Accounts

3. In the left pane, select the checkbox for the user(s) you want to assign.

4. In the right pane, select the folder to which you are assigning them. The folder will highlight as shown.

5. Click the **right arrow**. The user(s) will appear within the assigned folder.

Viewing Accounts For a User in the Hierarchy

To view a hitlist of accounts:

1. Enroll a User with the Hierarchy sample enrollment page (not the eaSample page). Specify a primary/secondary account and select **user** as the role.

2. Log in as that User and click the Hierarchy link to go to the Hierarchy Console.

3. The Hierarchy Console will display a hitlist of available accounts, but not the Hierarchy assignment panes.

Customizing the Hierarchy Console

The 3.2 release of the hierarchy Console **must be customized** with the `com.edocs.hierarchy` API to support advanced features, including:

- Viewing and paying statements directly through the Hierarchy Console (as this is controlled by each web application)
- Custom logging and error messaging
- Reporting on data stored in the hierarchy
- Storing hierarchy history (NOT CUSTOMIZABLE. Only the most current hierarchy is supported.)
- Customizing the Hierarchy Console user interface
- Searching the hierarchy
- Support for multiple DDNs, for example to group the accounts of a single customer with ten accounts across three DDNs.
- Linking to a user's statement from within the Hierarchy Console

For custom solutions integrating the Hierarchy Console, please consult your edocs Professional Services representative.

Application Programming Interfaces (APIs) for the Hierarchy Console

com.edocs.hierarchy

Provides an interface to define the values of constants for hierarchy fields and utility classes for creating a hierarchy as a directory information tree (DIT).

com.edocs.hierarchy.app

Provides servlet classes to extend and override the example servlets in **`com.edocs.app`** for user login and content access to account data mapped to the hierarchy.

com.edocs.hierarchy.render

Provides an interface and classes to design and display the user interface of the hierarchy console as HTML.

com.edocs.hierarchy.taglib

Provides a custom tag library of JSP tags for presenting hierarchy data.

For more information on Hierarchy APIs, please see the eaDirect 3.2 Application Programming Interface Specification (Javadoc).

For example implementations of these APIs in the Hierarchy Console web application, please see the sample JSP pages in **`ear-hierarchySample.EAR.`**

Cutting, Copying, and Pasting DDF Entities

4

About Cutting, Copying, and Pasting in a DDF

You can now use cut, copy, and paste functionality to move or copy the following DDF content within a DDF or to another DDF:

- Page styles
- Fields, including group fields
- Markers
- Tables and columns
- Groups

Using cut, copy, and paste functions saves time and reduces errors when building multiple application views.

(Note that you can only open a second DDF with a second data input file.)

If you copy and paste an entity into the same location as the original, you may be prompted to give it a new name. A “location” just refers to another place within the same DDF.

When you cut and paste an item, DefTool also moves or copies any associated child items (entities that belong to them) as child items in the tree.

The basic rules and behavior associated with copying each type of DDF entity are described here.

Page styles

When you copy and paste a page style (in the same DDF), DefTool prompts you to enter a new name. DefTool copies all the entities belonging to that page style onto the new one. Therefore, if a marker has a location in one page style, DefTool creates another marker in the target page style as well. (DefTool displays the marker under the root of the tree, not the node of the page style.)

Fields, group fields

You can cut, copy, and paste fields within the same page style or to a different one. You can also cut, copy, and paste group fields from one group to another.

If you paste a field within the same page style, you must provide a new name for the field.

If you paste a field to another page style, DefTool adds it as another location, so you do not need to rename the field.

Markers

You can move or copy a marker pair (start and end markers) to the same or another page style. DefTool moves or copies the associated tables and groups with the marker pair.

If you copy a marker pair to another page style, DefTool adds an additional location to the markers and displays the marker in the Definition Tree root (and not under the page styles).

If the source and target page styles are the same, DefTool adds a new location to each marker, and markers remain under their page node.

You can also move or copy *individual* start and end markers to the same or to another start or end marker.

Note that if you rename a marker location, it is equivalent to renaming the marker.

Tables, columns

To copy a table, you must paste it to the marker pair you want to associate it with.

When you copy and paste a table, DefTool prompts you for a new name and uses this name as a prefix to the columns of the new table. DefTool displays the renamed table under the marker node in the Definition Tree.

When you paste a table, DefTool copies or moves the associated anchor and columns.

You can also cut or copy a column from one table to another table. DefTool prompts you for a new name. If you cut a column that is an internal anchor, the anchor becomes external.

Groups

You can copy and paste a group to the same set of markers or to a new pair you want to associate the group with. When you copy and paste a group you must provide a new group name (only the top-most group is a copy; the table and column names remain the same).

When you paste a group, DefTool moves all child groups and tables belonging to the group.

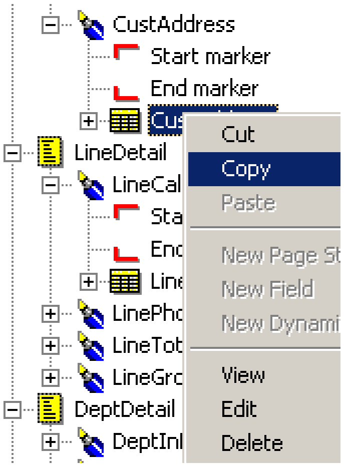
You can also copy a group and paste it as a child group under another parent group (in this case the group name stays the same).

How to Move or Copy DDF Entities

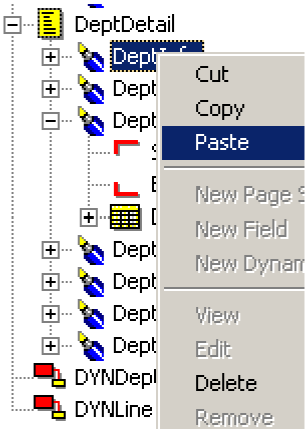
To move or copy a DDF entity:

1. With the DDF open in DefTool, right-click the item in the tree you want to cut or copy and paste.

2. Choose `Cut` or `Copy` from the right-click menu.



3. Right-click the target location or item in the open DDF (or switch to another open DDF and right-click the target item there) and select **`Paste`** from the right-click menu.



4. If DefTool prompts you for a new name, enter a unique name for the item and click OK.

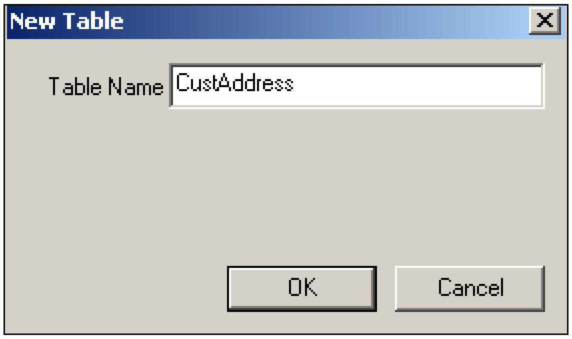
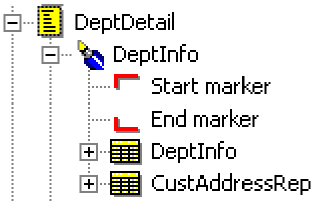
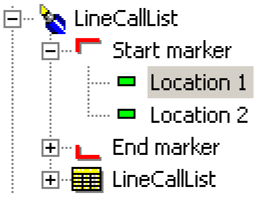


Table Name CustAddress

5. DefTool redisplays the Definition Tree to reflect the result of your edits.



6. Note that if you copy and paste a marker pair to a different page style, DefTool displays the markers under the root, with multiple location nodes beneath it. To see which page style a location node represents, right-click on the node, such as “Location 1” and select `Edit` from the right-click menu. The Marker Properties dialog shows the page style name.



7. Click 💾 to save the DDF.

Custom Post-Conversions 5

About Creating Custom Post-Conversions

eaDirect 3.2 lets you create custom post-conversions to use with the data extraction rules you define using DefTool. When creating a new eaDirect application in DefTool, you define rules for extracting each element from your data input file for online presentment. The rules tell eaDirect where to find the data in the file (general location), the type of information it contains (content), and how to format the extracted data.

DefTool lets you apply post-conversion formatting specifications to a field or column value to further format the data for dynamic display online or for data storage purposes using the Command Center Indexer job. You can currently apply standard positional-based alphanumeric and date post-extraction conversions to extracted data, such as “Delete all trailing zeros or spaces™” and “Translate Aug 21, 2002 to 08/21/2002™.” eaDirect 3.2 lets you create *customized* post-conversions to apply along with any sequence of existing (standard) post-conversions. You can apply multiple standard or custom post-conversions to a data field.

You can create custom post-conversions using:

- Regular expressions using the third-party PCRE 3.9 RegEx library for non-date and non-time data
- Date and time string formats

New regular expression library

When creating a regular expression for custom post-conversions in eaDirect, you must use the syntax of the third-party PCRE 3.9 RegEx library. For information about this library and the use of regular expressions, see “Mastering Regular Expressions” by Jeffrey E.F. Friedl, available at http://www.oreilly.com/catalog/regex/toc.html

It is important to note that is not the same library you use to create regular expressions for extracting data using the DDF.

Date and time formats

See “Appendix A – Data Definitions” in the *eaDirect User’s Guide* for a list of date and time formats you can use to build post-conversions for date and time data in eaDirect.

Reusing custom post-conversions

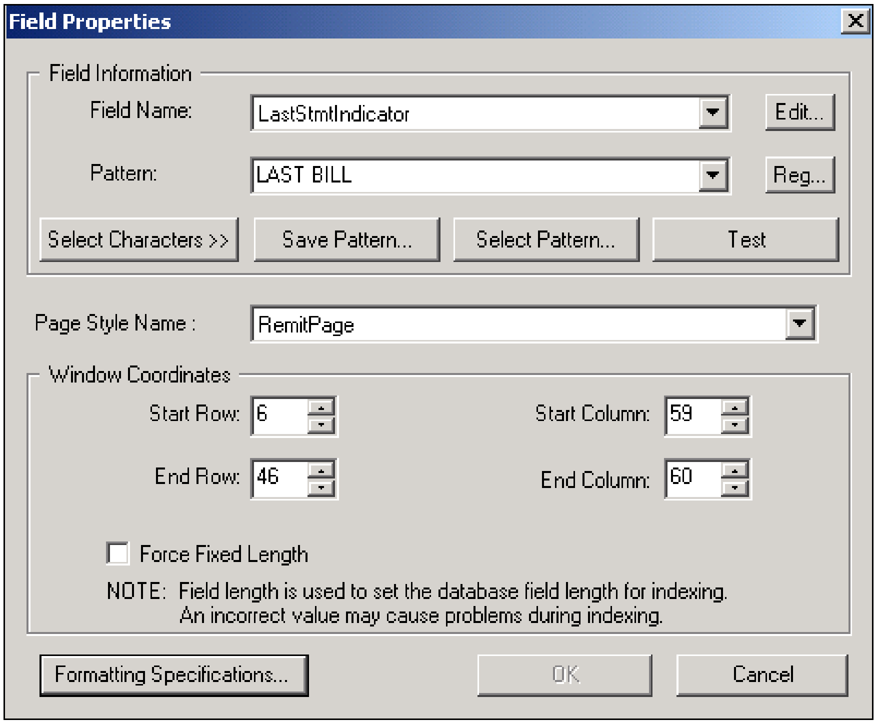
Each custom post-conversion you create becomes available for reuse with other data fields in the same eaDirect application (in the same DDF).

In addition, DefTool lets you save an application’s custom post-conversions to an xml file. This enables you to individually copy and paste the expressions into custom post-conversions you create in other another DDF.

Creating Custom Post-Conversions

To create a custom post-conversion for a field or column:

1. When defining the extraction rules for a field, DefTool displays the Field Properties dialog. (When defining a column, DefTool uses the similar Column Properties dialog.)



Field Properties

Field Information

Field Name: LastStmtIndicator

Pattern: LAST BILL

Page Style Name : RemitPage

Window Coordinates

Start Row: 6

Start Column: 59

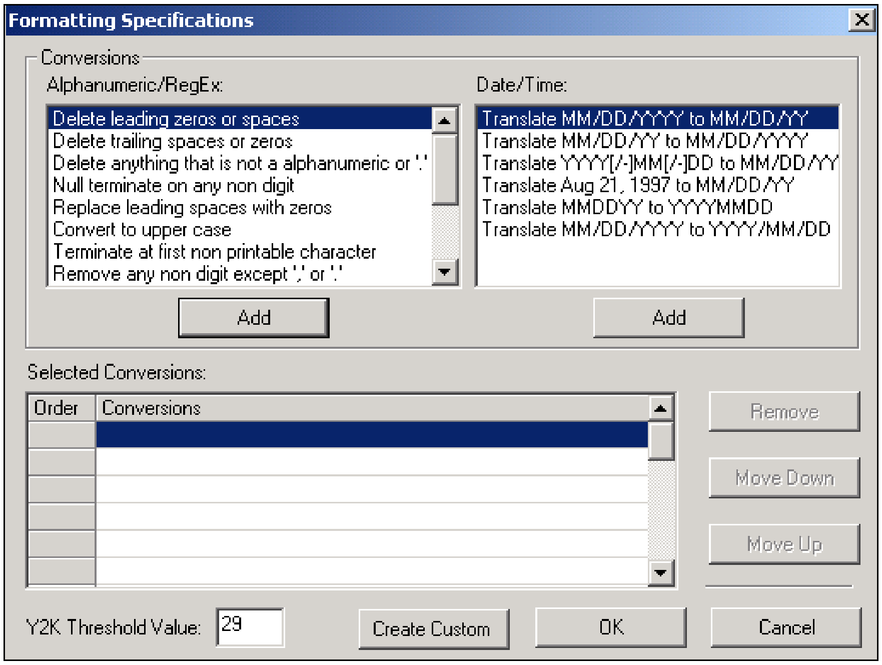
End Row: 46

End Column: 60

□ Force Fixed Length

NOTE: Field length is used to set the database field length for indexing.
An incorrect value may cause problems during indexing.

2. Click **`Formatting Specifications`**. DefTool displays the Formatting Specifications dialog. Use this dialog to choose any combination of standard or custom post-conversions and their application sequence to apply to the field or column.



Formatting Specifications

Conversions

Alphanumeric/RegEx:

- Delete leading zeros or spaces
- Delete trailing spaces or zeros
- Delete anything that is not a alphanumeric or '.'
- Null terminate on any non digit
- Replace leading spaces with zeros
- Convert to upper case
- Terminate at first non printable character
- Remove any non digit except ',' or '.'

Date/Time:

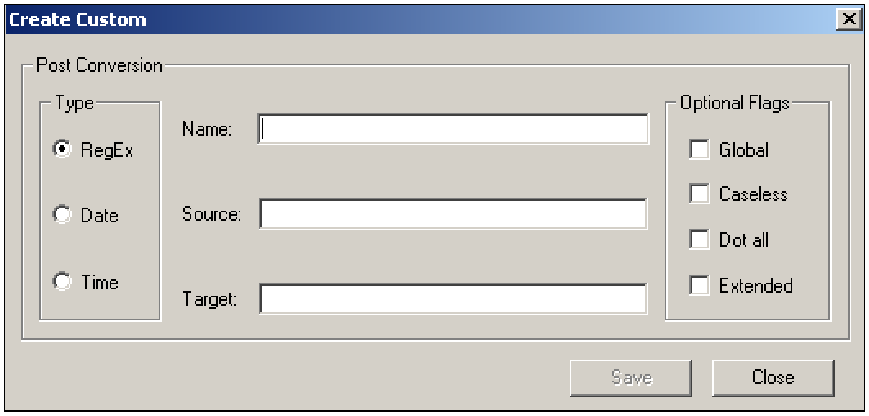
- Translate MM/DD/YYYY to MM/DD/YY
- Translate MM/DD/YY to MM/DD/YYYY
- Translate YYYY[/-]MM[/-]DD to MM/DD/YY
- Translate Aug 21, 1997 to MM/DD/YY
- Translate MMDDYY to YYYYMMDD
- Translate MM/DD/YYYY to YYYY/MM/DD

Selected Conversions:

Order	Conversions

Y2K Threshold Value: 29

3. Click **`Create Custom`**. DefTool displays the Create Custom dialog.



Create Custom

Post Conversion

Type

● RegEx

○ Date

○ Time

Name:

Source:

Target:

Optional Flags

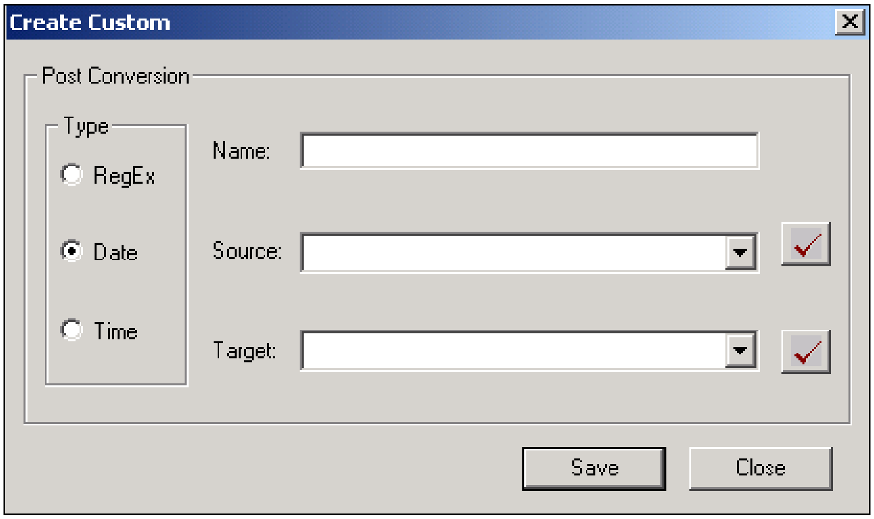
☐ Global

☐ Caseless

☐ Dot all

☐ Extended

4. Specify the type of post-conversion you want to create: Date or Time to convert most date or time data, or RegEx to specify a regular expression for all other data types. If you choose a date or time, the Create Custom dialog redisplays with different options.



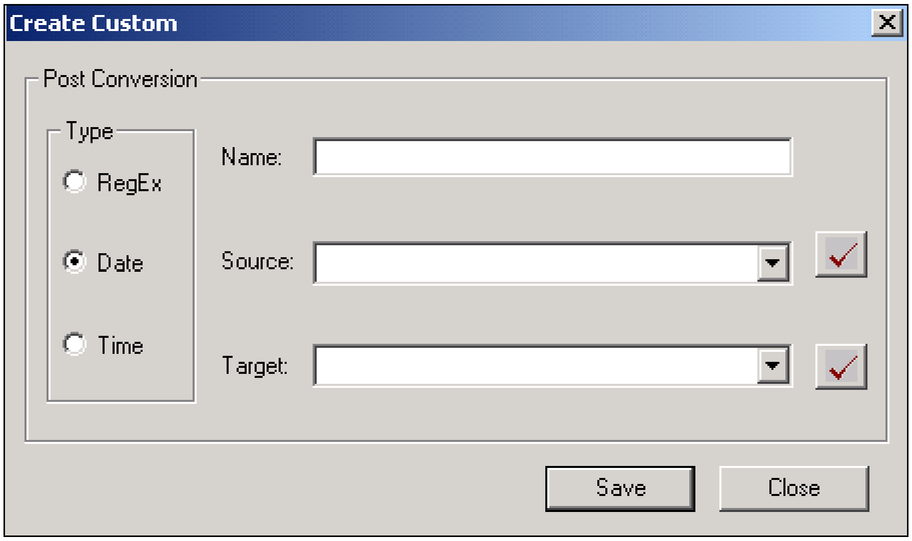
5. Specify a meaningful name to identify the custom post-conversion. (This name appears in the “Alphanumeric/RegEx” or “Date/Time” conversion list on the Formatting Specifications dialog after you save, making it available to select with other fields or columns.)

6. In the Source field, enter the regular expression or date/time format functions you want to apply to the extracted data (this is on top of the regular expression defined on the Field Properties dialog that eaDirect first applies to the extracted data). For a regular expression, you can choose any of the following matching option letters; DefTool adds them to the expression between “(?” and “)”.

   - g - Global
   - i - Caseless
   - s - Dotall
   - x - Extended

7. In the Target field, enter the expression for converting the extracted source data for presentment.

8. When creating a date or time post-conversion, click ✓ after specifying the source and target functions to verify that the format is valid.



9. When you're through specifying the post-conversion functions, click **`Save`**. Continue defining any additional post-conversions you need, then click **`Close`**. DefTool displays the Formatting Specifications dialog with the new custom post-conversion in the appropriate conversion list.

10. To add the custom post-conversion to the field, highlight the conversion and click **`Add`**. Click **`OK`** when you are done defining formatting specifications for this field.

11. You can continue the process of defining extraction rules for this field. See the *eaDirect User's Guide* for complete instructions.

Editing a post-conversion

To edit a custom post-conversion:

1. With the DDF open in DefTool, from the View menu, select **`Custom List`**. The Custom Post Conversion List dialog appears.

2. Highlight the custom post-conversion you want and click **`Edit`**. The Edit Custom dialog appears.

3. Enter your edits. Click `Save`.
4. Click `Close`.

Removing a custom post-conversion from a field or column

To delete a post-conversion (custom or standard) from the formatting specification for a particular field:

1. With the DDF open in DefTool, right-click the field or column in the tree.
2. Select `Edit` from the right-click menu.
3. Click `Formatting Specifications`.
4. Highlight the custom post-conversion in the Selected Conversions list and click `Remove`.
5. Click `OK`.
6. Click `OK` on the Field (or Column) Properties dialog.

Deleting a custom post-conversion from an application

To delete a custom post-conversion from the application:

1. With the DDF open in DefTool, from the View menu, select `Custom List`. The Custom Post-Conversion List dialog appears.
2. Highlight the custom post-conversion you want and click `Delete`.
3. If the selected custom post-conversion is not in use, DefTool asks if you are sure you want to delete it. If it is in use, DefTool tells you which fields currently use the custom post-conversion and warns you that deleting it also removes it from the field formats. Click `Yes` to delete the custom post-conversion (or `No` to cancel). DefTool removes the custom post-conversion from the application and from any fields where it was in use.

4. Click **Close**.

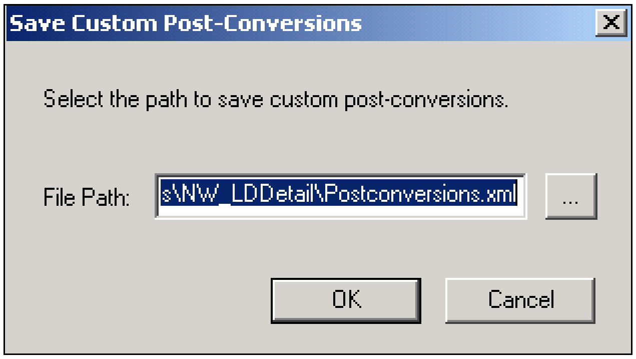
Sharing custom post-conversions with other applications

When you create and save a custom post-conversion, it becomes available for use in that application DDF only. To make those custom post-conversions available for use in another DDF, DefTool lets you save the custom post-conversions to a text file (in xml format). You must then recreate the custom post-conversions in the other DDF, copying and pasting the individual expressions from the file.

You can also add custom post-conversions to an existing file.

To save custom post-conversions to an external text file:

1. From the File menu, select **Save custom post-conversions**. The Save Custom Post-Conversions dialog appears.

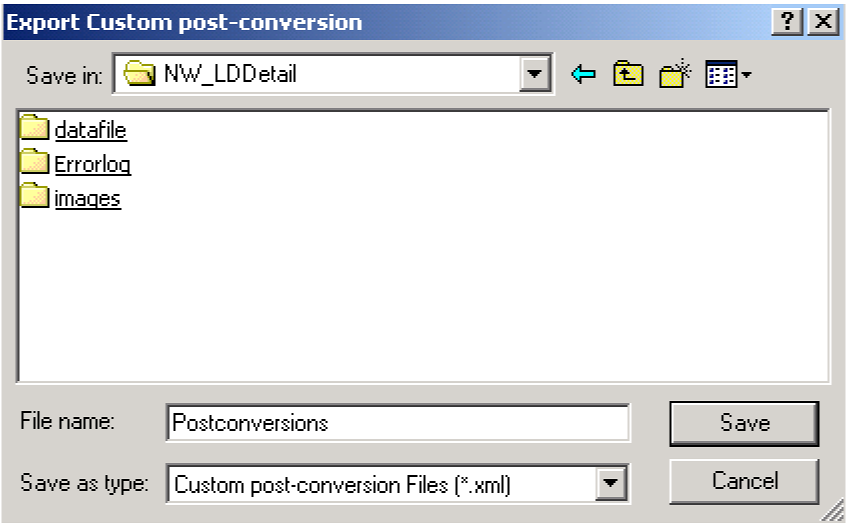


Save Custom Post-Conversions

Select the path to save custom post-conversions.

File Path: s\NW_LDDetail\Postconversions.xml

2. Click ... to browse for a directory and/or edit the file name. DefTool displays the Export Custom Post-Conversion dialog. Specify the path and/or file name and click **Save**.



Export Custom post-conversion

Save in: NW_LDDetail

datafile
Errorlog
images

File name: Postconversions

Save as type: Custom post-conversion Files (*.xml)

3. Click **OK**.

To copy saved custom post-conversions to another DDF:

1. In DefTool, open the DDF you want to add the saved custom post-conversions to.

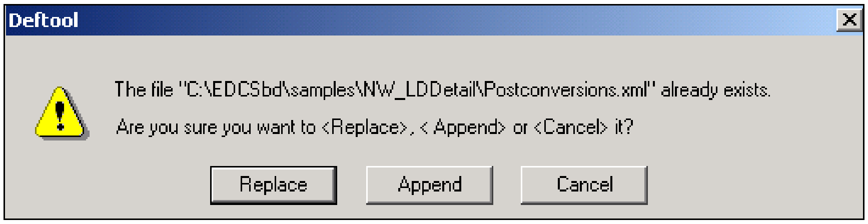
2. In a text editor, open the xml file containing the saved custom post-conversions.

3. Follow the instructions in this chapter to create custom post-conversions, copying and pasting the individual expressions from the xml file to recreate the custom post-conversions in this DDF.

To append or replace an existing custom-post-conversions file:

1. From the File menu, select `Save Custom Post-Conversions`. The Save Custom Post-Conversions dialog appears.

2. Click ... to browse for the xml file you want to append or replace. DefTool displays the Export Custom Post-Conversion dialog:



Deftool

The file "C:\EDCSbd\samples\NW_LDDetail\Postconversions.xml" already exists.
Are you sure you want to <Replace>, < Append> or <Cancel> it?

3. Click **`Append`** or **`Replace`**. DefTool adds any new custom post-conversions to the file or replaces the file contents entirely.

Date and Time Format Constraints

See “Appendix A: Data Definitions” in the *eaDirect User’s Guide* for details about the basic date and time formats available.

You must observe the following restrictions when using date and time formats to create custom post-conversions in DefTool:

- When converting a time duration format to another duration format, use only the following string components: %s, %Q, %K and %G
- You cannot convert a time format to a time duration or vice-versa. For example, you cannot convert the time duration “%K:%Q” to time format “%H:%M:%S” or the reverse. (It is OK to convert a time format to another time format, or to convert a time duration to a time duration, however.)
- Do not include both time duration formats and time formats in either the source or target. For example, you cannot have a source format of “%H:%Q:%S” and make the target “%T” because “%Q” represents a duration and “%H” is a time format.
- You cannot convert minutes to hours. Since the clock is in 1000 minutes, you can specify only %Q for the source. This only stores minutes in the tm structure and will contain values up to 999 and does not populate the hour field.

Examples of Custom Post-Conversions

The following examples show the regular expressions and formats you would use in the Source and Target fields on the Create Custom dialog to create various custom post-conversions.

<table>
<tr><th>Post-Conversion</th><th>Source</th><th>Target</th><th>Flag (Mode)</th></tr>
<tr><td><b>1,234,567.89</b> to <b>1.234.567,89</b>
Convert dot(.) to comma(,) and comma(,) to dot(.)</td><td><code>(\.){1}|(\,)</code></td><td><code>(?1,)(?2.)</code></td><td><code>/g</code> (Global)</td></tr>
<tr><td><b>1234567.89</b> to <b>1.234.567,89</b>
Insert thousand separator as dot(.) and decimal separator as comma(,)</td><td><code>(\.){1}|\G(\d{</code></td><td><code>(?1,)(?2$&.)</code></td><td><code>/g</code> (Global)</td></tr>
<tr><td><b>00001234567.89</b> to <b>1.234.567,89</b>
Remove leading zeros; insert thousand separator as dot(.); change decimal separator to comma(,)</td><td><code>(\.){1}|^0+|\G(\d{1,3})(?=(?:\d\d\d)+(?!\d))</code></td><td><code>(?1,)(?2)(?3$&.)</code></td><td><code>/g</code> (Global)</td></tr>
<tr><td><b>-1234567.89</b> to <b>-1.234.567,89</b>
Maintain negative sign; insert thousand separator as dot(.); change decimal separator to comma(,)</td><td><code>(\.){1}|\G\-?(\d{1,3})(?=(?:\d\d\d)+(?!\d))</code></td><td><code>(?1,)(?2$&.)</code></td><td><code>/g</code> (Global)</td></tr>
</table>

<table>
<tr><th>Post-Conversion</th><th>Source (Regular Expression)</th><th>Target (Format)</th><th>Flag (Mode)</th></tr>
<tr><td><b>-1,234,567.89</b> to <b>-1.234.567,89</b>
Maintain negative sign; change thousand separator to dot(.) and decimal separator to comma(,)</td><td><code>(\.){1}|(,)</code></td><td><code>(?1,)(?2.)</code></td><td><code>/g</code> (Global)</td></tr>
<tr><td><b>€1,234,567.89</b> to <b>€1.234.567,89</b>
Maintain the currency sign; change decimal separator to comma(,) and thousand separator to dot(.)</td><td><code>(\.){1}|(,)</code></td><td><code>(?1,)(?2.)</code></td><td><code>/g</code> (Global)</td></tr>
<tr><td><b>€-78912354354.65767</b> to <b>€-78.912.354.354,65767</b>
Maintain currency sign; change decimal separator to comma(,); insert thousand separator as dot(.)</td><td><code>(\.){1}|\G(€?|\-?|€?\-?)(\d{1,3})(?=(?:\d\d\d)+(?!\d))</code></td><td><code>(?1,)(?2$&.)</code></td><td><code>/g</code> (Global)</td></tr>
<tr><td><b>€-000058736545656.74647467</b> to <b>€-58.736.545.656,74647467</b>
Maintain currency sign; change decimal separator to comma(,); insert thousand separator as dot(.); trim off leading zeros</td><td><code>€?(0+)|€\-(0+)|\-?(0+)|(\.){1}|\G(€?|\-?|€?\-?)(\d{1,3})(?=(?:\d\d\d)+(?!\d))</code></td><td><code>(?1€)(?2€-)(?3-)(?4,)(?5$&.)</code></td><td><code>/g</code> (Global)</td></tr>
</table>

<table>
<tr><th>Post-Conversion</th><th>Source
(Regular Expression)</th><th>Target
(Format)</th><th>Flag (Mode)</th></tr>
<tr><td><b>00000000.234</b> to <b>0,234</b>
Trim off leading zeros; change decimal separator to comma(,)</td><td><code>(0){1,}(\.)</code></td><td><code>0,</code></td><td><code>/g</code> (Global)</td></tr>
<tr><td><b>000000001.234</b> to <b>1,234</b>
Trim off leading zeros; change decimal separator to comma(,)</td><td><code>(\.)|0+(\.){1}|^(0)+</code></td><td><code>(?1,)(?20,)(?3)</code></td><td><code>/g</code> (Global)</td></tr>
<tr><td><b>9999999999</b> to <b>(999)999-9999</b>
This requires you create and apply two separate, sequential post-conversions.</td><td>Post-conversion 1:
<code>\G^(\d{1})|\G(\d{5})</code>
Post-conversion 2:
<code>\G(\()+(\d{3})</code></td><td>Post-conversion 1:
<code>(?1\($&)(?2$&-)</code>
Post-conversion 2:
<code>(?1$&\))</code></td><td></td></tr>
<tr><td><b>10/14/2002</b> to <b>October 14, 2002</b></td><td>%m/%d/%Y</td><td>%B %d, %Y</td><td></td></tr>
<tr><td><b>18 julio 2002</b> or <b>20 agosto 2002</b> or <b>05 mayo 2001</b> to
<b>18-07-2002</b> or <b>20-08-2002</b>
Use these conversions for dates in languages other than English and French.
This requires you create and apply two separate, sequential post-conversions.</td><td>Post-conversion 1 (Regular expression):
<code>(?:enero)|(?:febrero)|(?:marzo)|(?:abril)|(?:mayo)|(?:junio)|(?:julio)|(?:agosto)|(?:septiembre)|(?:octubre)|(?:noviembre)|(?:diciembre)</code>
Post-conversion 2 (Date conversion):
%d %m %Y</td><td>Post-conversion 1 (Regular expression):
<code>(?101)|(?202)|(?303)|(?404)|(?505)|(?606)|(?707)|(?808)|(?909)|(?1010)|(?1111)|(?1212)</code>
Post-conversion 2 (Date conversion):
%d-%m-%Y</td><td><code>/g</code> (Global)</td></tr>
</table>

Explanation of first example

The first example in the table above shows that to convert 1,234,567.89 to 1.234.567,89, the post-conversion you specify must change the dot(.) to a comma(,) and the comma(,) to a dot(.).

Source – The regular expression looks for exactly one occurrence of dot(.) or a comma(,) in the source data:

```
(\\.){1}|(\\,)
```

Target – The target format specifies that if condition 1 is satisfied, replace the dot with a comma; else if condition 2 is satisfied, replace the comma with a dot:

```
(?1,)(?2.)
```

Flag (mode) – The `/g` or Global flag applies the formatting information globally (throughout the input string).

Explanation of second example

The second example in the table above shows that to convert 1234567.89 to 1.234.567,89, the post-conversion you specify must search for and insert the thousand separators as dot(.) and the decimal separator as comma(,).

Source – The regular expression looks for exactly one occurrence of a dot(.) or sets the anchor to the beginning, if a digit between exactly one to three occurrences, look ahead three digits followed by not a digit:

```
(\.){1}|\G(\d{1,3})(?=(?:\d\d\d)+(?!\d))
```

Target – The target format specifies that if condition 1 is satisfied, then replace the dot with a comma, else if condition 2 is satisfied, then insert a dot(.):

```
(?1,)(?2$&.)
```

Flag (mode) – The `/g` or Global flag applies the formatting information globally (throughout the input string).