

Composite Rules

Acenza White Paper

Overview

Customization is the essence of Acenza. Through customization, you can configure Acenza to manage the flow of work and work objects in any organization. Composite rules are a critical part of this customization because they allow you to specify the exact information you want Acenza to retrieve and display.

A composite rule is a method that defines how Acenza assembles information from values in the database and then displays it in a particular field on the WebTop user interface. You create a composite rule in Acenza Admin by associating data fields to an Object Type Class (OTC).

Scope

This document explains how to create composite rules within Acenza Admin and validate their use on the WebTop. It is intended for a technical audience with knowledge of how to configure Acenza and use SQL Query Analyzer. In particular, anyone who uses Acenza Admin to create or modify Object Type Classes should read this document.

Important: This document does not address how to use composite rules in the context of SQL priming scripts.

Note: The examples described in this document are based on the default Insurance module that is distributed with Acenza 2.0.

Creating a Composite Rule

Creating a composite rule takes place in Admin. Verifying that the composite rule works as expected occurs on the WebTop.

Within Admin, you define OTCs for specific types of work, such as contacts and documents, and you associate data fields to the OTC. It is within the context of OTC fields that you actually define a composite rule to display data in specific fields on the WebTop. ListFields 1 through 5 are reserved specifically for that purpose. A ListField is an OTC field that holds data that Acenza dynamically assembles from the database. You define the ListField in Admin; you see the results on the WebTop.

On the WebTop, you create an object of the same type. Any data that exists in the field that you associated with ListField1 through ListField5 in Admin, displays when you search on that object type in the WebTop. The results can appear in any of several lists, such as a Search Result List or a Work Queue List.

Fields Used for Composite Rule Definitions

Composite rules can be defined in any OTC Field. However, because the majority of composite rules return data to display in a list, such as a search list, most composite rules are defined in ListField1 through ListField5. For this reason, the examples in this document focus on composite rules defined in ListFields 1 through 5.

Other OTC fields that can be used for composite rules are DisplayFields 1 through 5. A DisplayField is similar to a ListField in that it holds data that Acenza dynamically assembles from the database. DisplayFields, however, are typically used to retrieve data from the parent object at the child level. In addition, composite rules can be defined in customer fields for general data storage or for data within the same object.

Steps to Create a Composite Rule

The steps to create a composite rule can be summarized as follows:

- Select an Object Type Class to modify in Admin.
- In the Composite Rule text box that is part of the Modify OTC Field dialog box, assign a field or combination of fields to ListField1 through ListField5.
- Reset cache information in Admin or exit Admin to update the cache information automatically.
- Run Merlin, steps 4 and 7, to generate the objects and update the cache.
- Validate your result list in the WebTop by searching for the object.

The examples that follow explain each step in detail.

Composite Rule Syntax

Acenza's composite rules use a combination of custom Acenza symbols and standard SQL terms and functions. Almost any valid SQL function that returns a value can be used in a composite rule. (Example 6 shows a composite rule that uses dynamic SQL.)

The following table lists terms and symbols found in composite rules.

Term	Meaning
@	SQL symbol used with a varname.
' ' (single quotes)	SQL format used to enclose any separator character, such as a comma.
SUBSTRING	SQL server function that allows you to strip a fixed number of characters from a string.
ltrim	SQL term that means trim the left side of a string of blank spaces or carriage returns.
rtrim	SQL term that means trim the right side of a string of blank spaces or carriage returns
str	SQL function used to convert an integer to a string.
!\$~	Unique Acenza characters. The !\$ serves as a delimiter. The tilde (~) means dynamic translation of a list.
#\$	Unique Acenza characters reserved to dynamically translate the following system variables: assignedToID, statusID, objStateID, and ownerID.

Examples

This document presents six types of composite rules:

- String + string
- Trimmed string + string
- String + integer without using ~ for translation
- String + integer using ~ for translation
- Dynamic translation of system variables
- Dynamic SQL

The examples that follow—listed in order of increasing complexity—show how to create these types of rules. The intent of these examples is to show the form and function of typical composite rules. Using these examples as a guide, you can create composite rules for your customized Acenza system using any valid field name that is associated with an OTC.

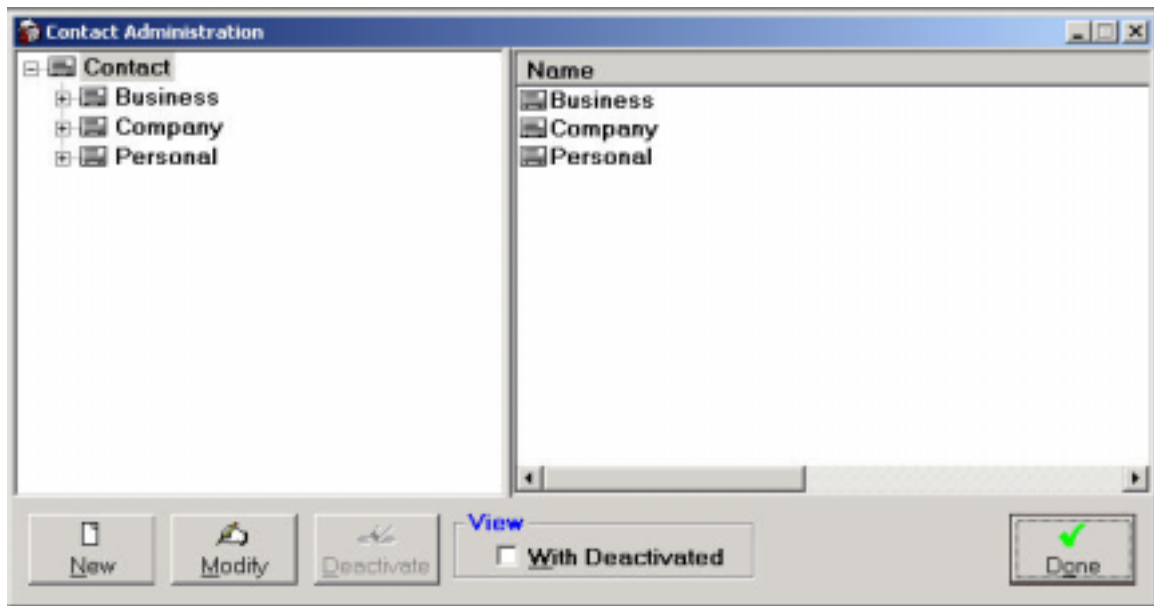
Example 1: String + String

This method concatenates two fields (strings) that belong to the same OTC, and displays the data in a WebTop list.

Scenario

To create the composite rule:

1. In Admin, select **Objects**. Then select **Contacts**. The Contact Administration dialog box appears.



2. Select an OTC, such as **Business**, and click the **Modify** button. The Modify Object Type Class (OTC) dialog box appears.
3. Click the **Fields** button to expose the OTC Fields Administration dialog box.
4. Select ListField1. Then click the **Modify** button. The Modify OTC Field dialog box appears.

Modify OTC Field

Field

Name: LIST_FIELD1

Description: List Field 1

Key Value:

Audit Null Search Search Required Required

Details

Label: List Field 1

Data Type: Miscellaneous (vc255n)

Length: 0

Var Name: listField1

Value

Options

Composite Translate

Composite Rule: @firstName + ',' + @lastName

Reset Select Ok Cancel

- Under Options, select the **Composite** check box.
- In the Composite Rule text box, enter the following composite rule:

@firstName + ',' + @lastName

Note: Field names are case sensitive and must be entered exactly as displayed in Admin's OTC Fields Administration dialog box in the Variable Name column. To ensure that you enter a field name correctly, copy and paste the Var Name into the composite rule.

- Click OK to close the dialog box.
- Update the cache by selecting **Reset Cache Information** from the File menu, or by exiting Admin.
- Run Merlin, steps 4 and 7, to generate the object and update the cache.
- Open the Acenza WebTop.

11. Click the **New** button. Then select **Contact** as the Object, **Business** as the Category, and **Agent** as the Type. Click **Continue**. The Create Contact dialog box appears.



12. Complete the required fields in the Create Contact dialog box and click **Create**.
13. Click the **Search** button. Ensure that the correct information is selected and click **Continue**. The Search Result List appears.

Acenza: Search

Acenza Object:

Business:

Category:

Type:

Filters

Subject:

Object State:

Status:

List Field 1:

City1:




State1:

Zip1:

14. Verify that the Search Result List displays Erin, Sumner.

Acenza: Search Result List

Page:

ID	Type	Primary ID	Status	Attachment	Attachment ID
 49	Agent	Erin, Sumner	Active		
 45	Agent		Active		
 44	Agent		Active		

Page:

Notes on Example 1

In the Search Result List, the name Erin, Sumner is displayed in the Primary ID column. This column is mapped to ListField1. To determine how columns are configured, or to change a configuration, open Admin and select **General, Group List Preferences**. Alternatively, open the WebTop and select **Preferences, Café, List Preferences**.

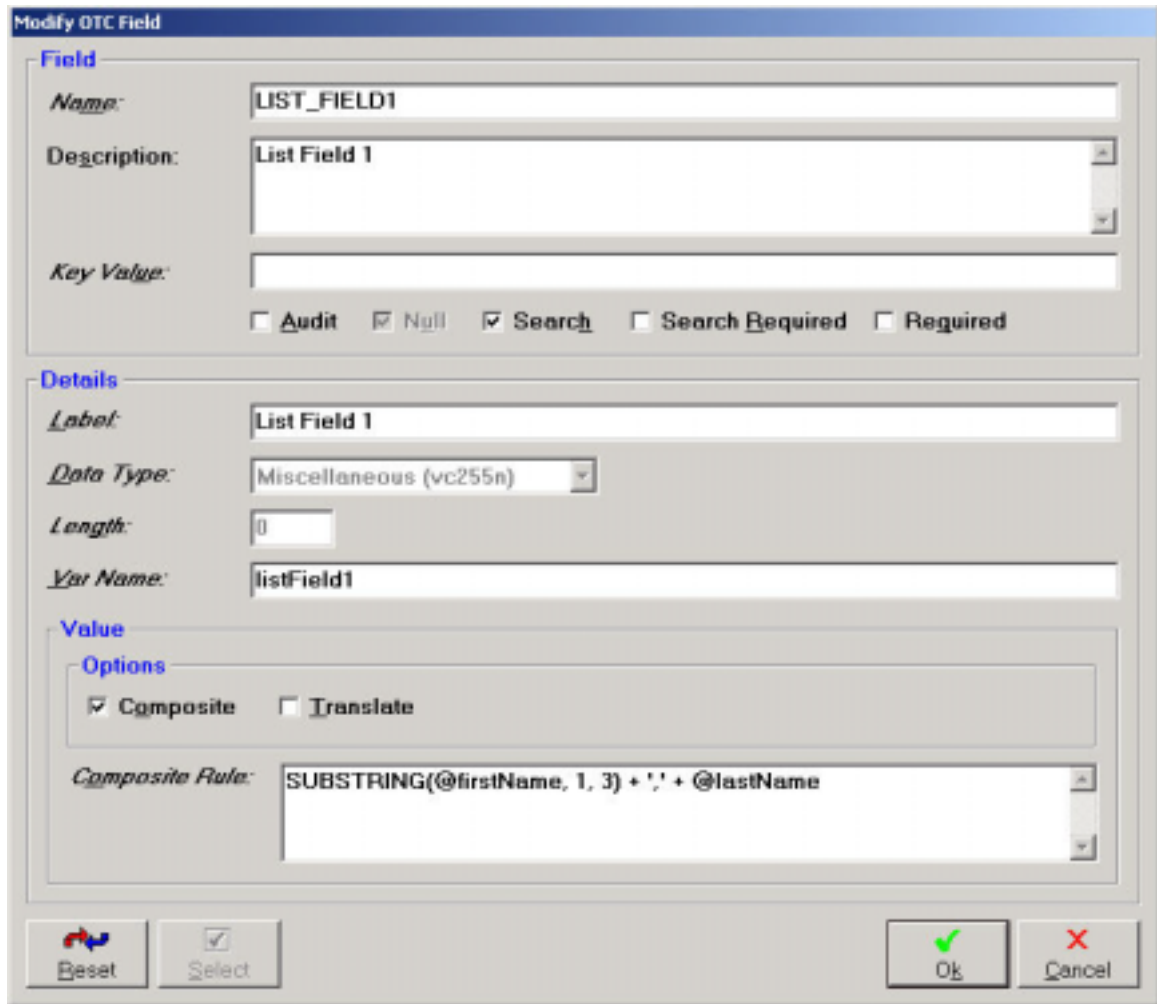
Example 2: Trimmed String + String

This rule displays only the first three characters of composite rule data.

Scenario

To create the composite rule:

1. In Admin, select **Objects**. Then select **Contacts**. The Contact Administration dialog box appears.
2. Select an OTC, such as **Business**, and click the **Modify** button. The Modify Object Type Class (OTC) dialog box appears.
3. Click the **Fields** button to expose the OTC Fields Administration dialog box.
4. Select ListField1. Then click the **Modify** button. The Modify OTC Field dialog box appears.



Modify OTC Field

Field

Name: LIST_FIELD1

Description: List Field 1

Key Value:

Audit Null Search Search Required Required

Details

Label: List Field 1

Data Type: Miscellaneous (vc255n)

Length: 0

Var Name: listField1

Value

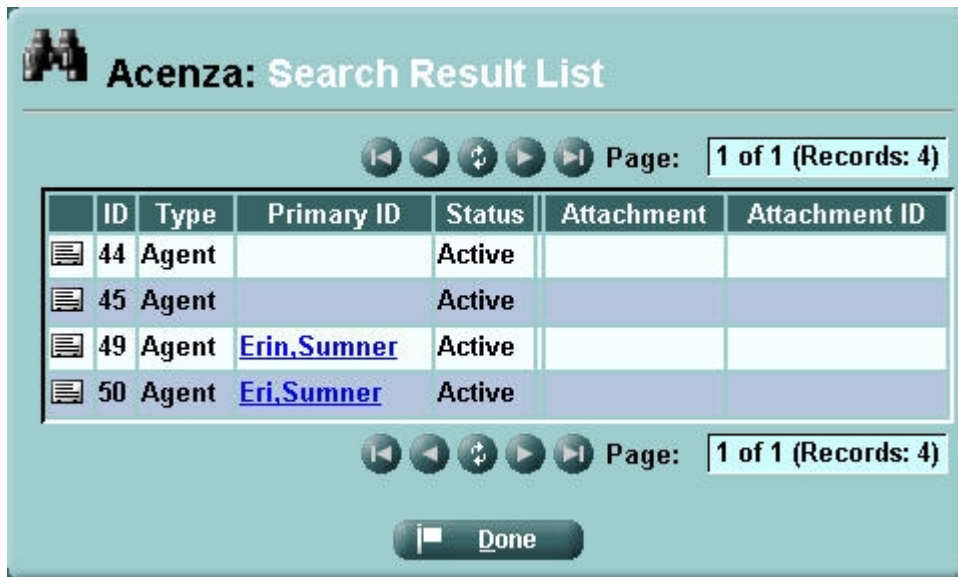
Options

Composite Translate

Composite Rule: SUBSTRING(@firstName, 1, 3) + \',\' + @lastName

Reset Select Ok Cancel

5. Under Options, select the **Composite** check box.
6. In the Composite Rule text box, enter the following composite rule:
SUBSTRING(@firstName, 1, 3) + \',\' + @lastName
7. Click OK to close the dialog box.
8. Update the cache by selecting Reset Cache Information from the File menu, or by exiting Admin.
9. Run Merlin, steps 4 and 7, to generate the object and update the cache.
10. Open the Acenza WebTop.
11. Click the **Search** button. Ensure that the correct information is selected and click **Continue**. The Search Result List appears.
12. Verify that the Search Result List displays Eri, Sumner.



ID	Type	Primary ID	Status	Attachment	Attachment ID
44	Agent		Active		
45	Agent		Active		
49	Agent	Erin,Sumner	Active		
50	Agent	Eri,Sumner	Active		

Example 3: String + Integer without Using ~ for Translation

This method concatenates a string field and an integer field from a list.

Scenario

To create the composite rule:

1. In Admin, select **Objects**. Then select **Documents**. The Document Administration dialog box appears.
2. Select **Application Docs** as the OTC and click **Modify**. The Modify Object Type Class (OTC) dialog box appears.
3. Click the **Fields** button to expose the OTC Fields Administration dialog box.
4. Select ListField2. Then click the **Modify** button. The Modify OTC Field dialog box appears.

Modify OTC Field

Field

Name: LIST_FIELD2

Description: List Field 2

Key Value:

Audit Null Search Search Required Required

Details

Label: List Field 2

Data Type: Miscellaneous (vc255n)

Length: 0

Var Name: listField2

Value

Options

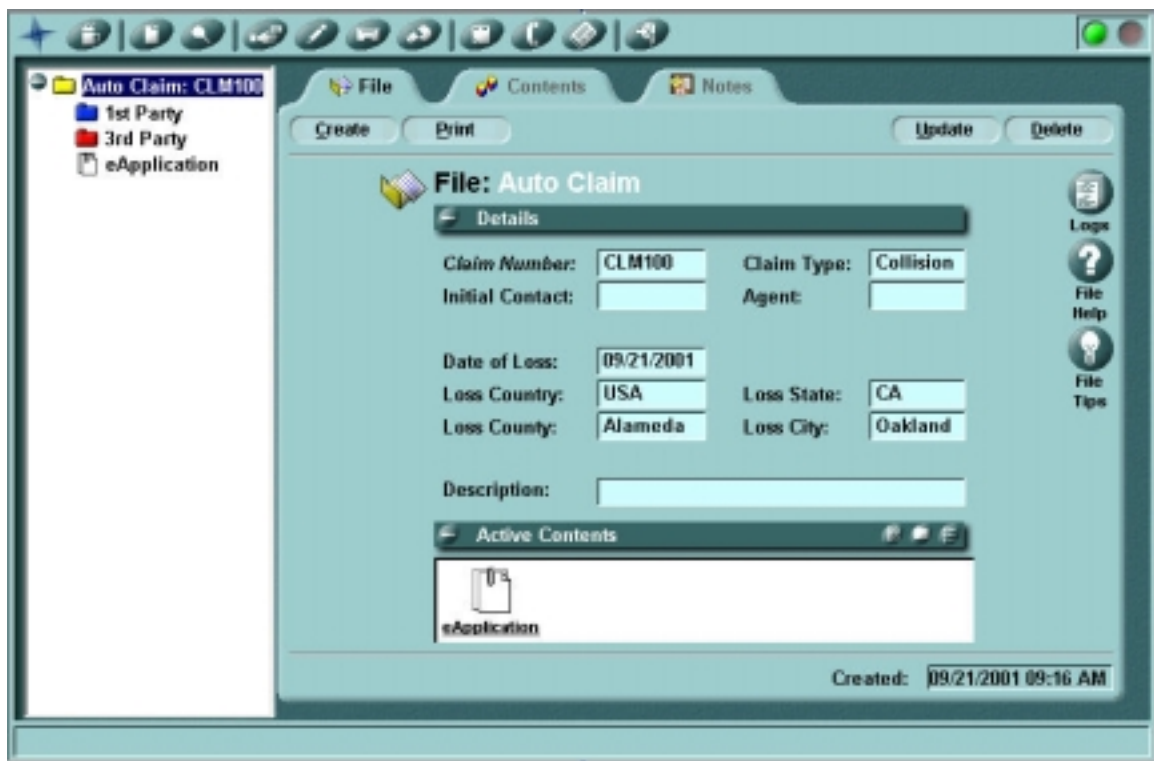
Composite Translate

Composite Rule: '\$!\$3403\$!\$' + ltrim(rtrim(str(@objTypeID))) + '\$!\$'

Reset Select Ok Cancel

5. Under Options, select the **Composite** check box.
6. In the Composite Rules text box, enter the following composite rule:
`'$!$3403$!$' + ltrim(rtrim(str(@objTypeID))) + '$!$'`
7. Click OK to close the dialog box.
8. Update the cache by selecting **Reset Cache Information** from the File menu or by exiting Admin.
9. Run Merlin, steps 4 and 7, to generate the object and update the cache.
10. Open the Acenza WebTop.
11. Click the **New** button. Then select **File** as the Object, **Policy** as the Category, and **Auto Policy** as the Type. Click **Continue**. The Create dialog box appears.

12. Complete the required fields in the Create dialog box, (such as Create Auto Policy dialog box). Click **Create**.
13. In the File dialog box, click the **Create** button to create a document to associate with the File.
14. In the New dialog box, select **Document** as the Object, **Policy Division** as the Business, **Application Docs** as the category, and **eApplication** as the Type. Click **Continue**. The Create Document dialog box appears.
15. Complete the required fields in the Create Document dialog box and Preview the document. Then click **Create**.
16. Verify that you can see eApplication (the document) in the tree at left and in the Active Contents window.



The screenshot displays the 'File: Auto Claim' form with the following data:

Claim Number:	CLM100	Claim Type:	Collision
Initial Contact:		Agent:	
Date of Loss:	09/21/2001	Loss State:	CA
Loss Country:	USA	Loss City:	Oakland
Loss County:	Alameda		
Description:			

The 'Active Contents' window shows a document icon labeled 'eApplication'. The status bar at the bottom right reads: Created: 09/21/2001 09:16 AM.

Note: To create a composite rule similar to Example 3, it is important that you define your composite rule in ListField2. It is the only field that can appear as a document in the tree.

Notes on Example 3

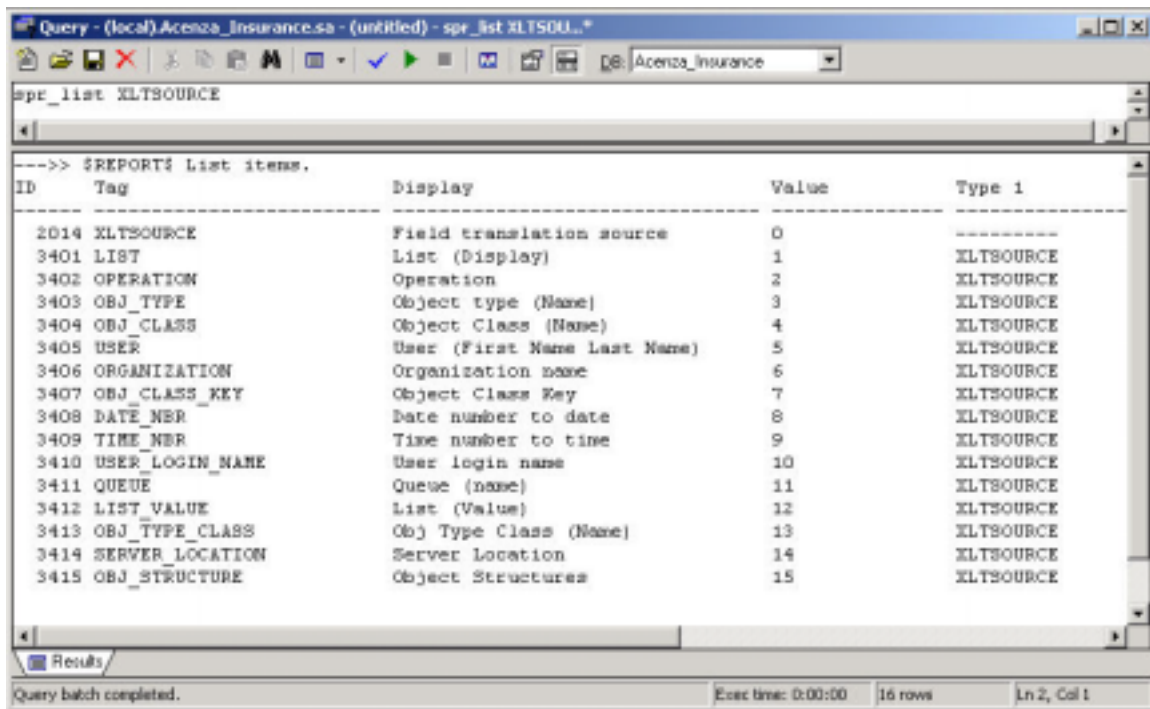
The composite rule in Example 3 translates an ID and converts an integer to a string. While most IDs require the tilde character (~) for translation, the one in Example 3 does not.

ID 2014 and the IDs listed under it (3401-3415) are special and do not require the tilde (~) character with !\$ for translation. ID 2014 (XLTSOURCE) is different from other IDs because all of the fundamental object structures in Acenza are listed under it, such as List, Object Class, Object Type, Queue, and so on.

To display the IDs listed under 2014, XLTSOURCE:

1. From the Windows Start menu, select **Programs, Microsoft SQL Server**. Then choose **Query Analyzer**. The Connect to SQL Server dialog box appears.
2. Enter the server and connection information that is appropriate for your site and click **OK**.
3. In the Query window, choose the **Acenza_Insurance** database and enter the following query:

```
spr_list XLTSOURCE
```



ID	Tag	Display	Value	Type 1
2014	XLTSOURCE	Field translation source	0	
3401	LIST	List (Display)	1	XLTSOURCE
3402	OPERATION	Operation	2	XLTSOURCE
3403	OBJ_TYPE	Object type (Name)	3	XLTSOURCE
3404	OBJ_CLASS	Object Class (Name)	4	XLTSOURCE
3405	USER	User (First Name Last Name)	5	XLTSOURCE
3406	ORGANIZATION	Organization name	6	XLTSOURCE
3407	OBJ_CLASS_KEY	Object Class Key	7	XLTSOURCE
3408	DATE_NBR	Date number to date	8	XLTSOURCE
3409	TIME_NBR	Time number to time	9	XLTSOURCE
3410	USER_LOGIN_NAME	User login name	10	XLTSOURCE
3411	QUEUE	Queue (name)	11	XLTSOURCE
3412	LIST_VALUE	List (Value)	12	XLTSOURCE
3413	OBJ_TYPE_CLASS	Obj Type Class (Name)	13	XLTSOURCE
3414	SERVER_LOCATION	Server Location	14	XLTSOURCE
3415	OBJ_STRUCTURE	Object Structures	15	XLTSOURCE

Query batch completed. Exec time: 0:00:00 16 rows 1n 2, Col 1

4. Press F5 or the Play button to execute the query.

Example 4: String + Integer Using ~ for Translation

This rule concatenates a string field and an integer field from a list.

Scenario

To create the composite rule:

1. In Admin, select **Objects**. Then select **Contacts**. The Contact Administration dialog box appears.
2. Select an OTC, such as **Company**, and click the **Modify** button. The Modify Object Type Class (OTC) dialog box appears.
3. Click the **Fields** button to expose the OTC Fields Administration dialog box.
4. Select ListField1. Then click the **Modify** button. The Modify OTC Field dialog box appears.

Modify OTC Field

Field

Name: LIST FIELD1

Description: List Field 1

Key Value:

Audit Null Search Search Required Required

Details

Label: List Field 1

Data Type: Miscellaneous (vc255n)

Length: 0

Var Name: listField1

Value

Options

Composite Translate

Composite Rule: @companyName + ', ' + '\$!\$~100004!\$' + ltrim(rtrim(str(@state1ID))) + '\$!\$'

Reset Select OK Cancel

5. Under Options, select the **Composite** check box.
6. In the Composite Rule text box, enter the following composite rule:
@companyName + ', ' + '\$!\$~100004!\$' + ltrim(rtrim(str(@state1ID))) + '\$!\$'
7. Click OK to close the dialog box.
8. Update the cache by selecting **Reset Cache Information** from the File menu or by exiting Admin.
9. Run Merlin, steps 4 and 7, to generate the object and update the cache.
10. Open the Acenza WebTop.
11. Click the **New** button. Select **Contact** as the Object, **Company** as the Category, and **Company** as the Type. Click **Continue**. The Create Contact dialog box appears.



Create Contact: Company

Personal
 Company
 Business

Company Name:
 Account Number:

Contact Name:
 Web Page:

Contact Phone:
 Web Login:

Fax Number:
 Email Address:

Company Number:

Mailing Address:

Attention:

Street:

City:

State:

Zip:

Country:

Billing Address:

Attention:

Street:

City:

State:

Zip:

Country:

12. Complete the required fields in the Create Contact dialog box and click **Create**.
13. Click the **Search** button. Ensure that the correct information is selected and click **Continue**. The Search Result List appears.
14. Verify that the Search Result List displays Eric, Inc., CA.



Acenza: Search Result List

Page:

ID	Type	Primary ID	Status	Attachment	Attachment ID
13	Company	Eric, Inc., CA	Active		

Page:

Notes on Example 4

Example 4 involves a more complex scenario than previous examples because it involves a dynamic translation and a conversion. You cannot add a string, which consists of characters, and an integer. You must convert the integer to a string.

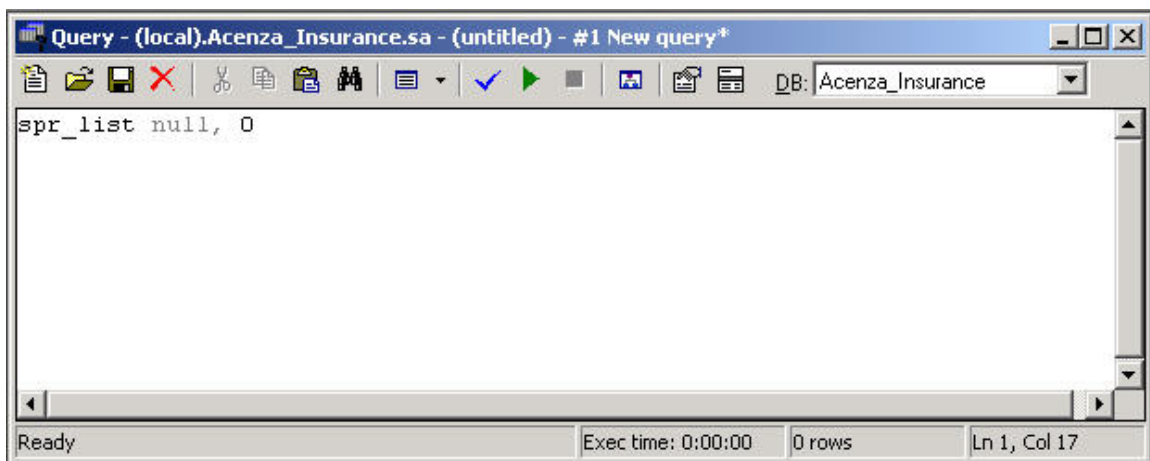
The conversion portion of Example 4 is expressed in the following portion of the rule: **ltrim(rtrim(str(@state1ID)))**. The SQL functions **ltrim** and **rtrim** remove any blank space or carriage returns from the left and right sides of the data. The **str** function converts an integer to a string.

Example 4 calls for a dynamic translation of an ID. In the composite rule, the tilde (~) character tells Acenza to dynamically translate the list associated with the ID 100004. How do we know what list is associated with the 100004?

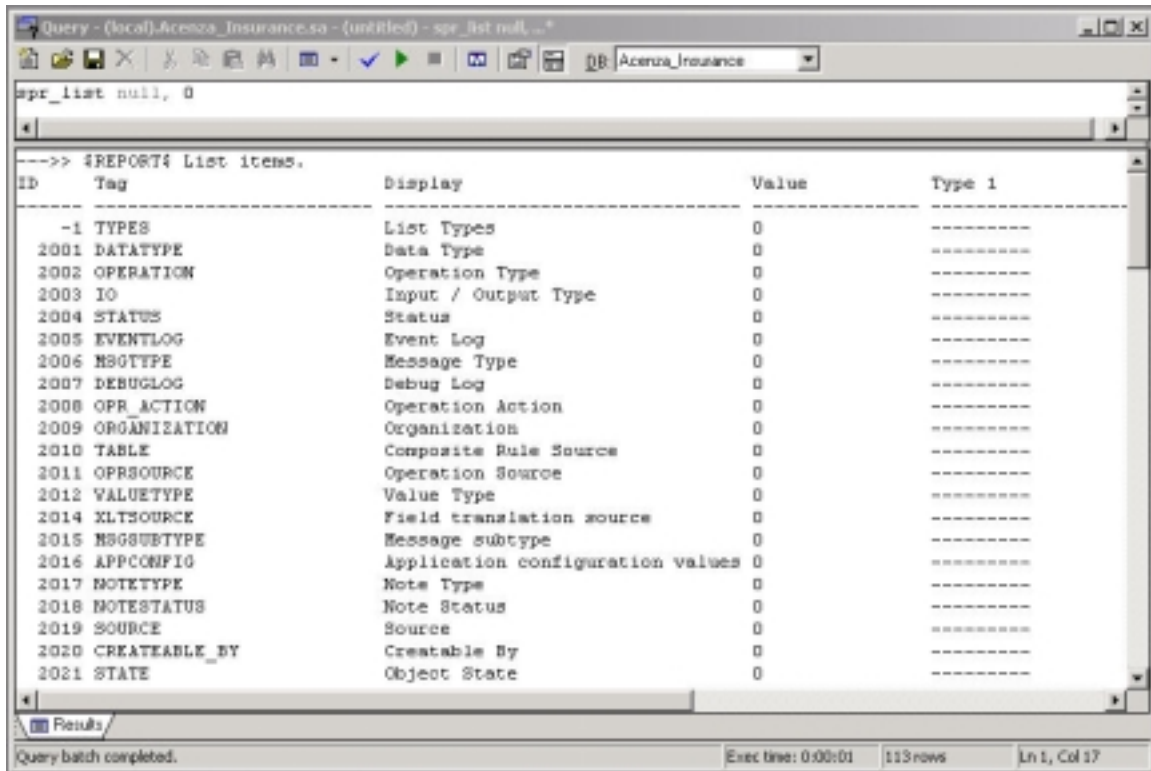
There are several ways to discover what list is associated with an ID.

1. From the Windows Start menu, select **Programs, Microsoft SQL Server**. Then choose **Query Analyzer**. The Connect to SQL Server dialog box appears.
2. Enter the server and connection information that is appropriate for your site and click OK.
3. In the Query window, choose the **Acenza_Insurance** database and enter the following query:

```
spr_list null, 0
```



4. Press F5 or the Play button to execute the query. A list of IDs and associated lists appears.



ID	Tag	Display	Value	Type 1
-1	TYPES	List Types	0	-----
2001	DATATYPE	Data Type	0	-----
2002	OPERATION	Operation Type	0	-----
2003	IO	Input / Output Type	0	-----
2004	STATUS	Status	0	-----
2005	EVENTLOG	Event Log	0	-----
2006	MSGTYPE	Message Type	0	-----
2007	DEBUGLOG	Debug Log	0	-----
2008	OPR_ACTION	Operation Action	0	-----
2009	ORGANIZATION	Organization	0	-----
2010	TABLE	Composite Rule Source	0	-----
2011	OPRSOURCE	Operation Source	0	-----
2012	VALUETYPE	Value Type	0	-----
2014	XLTSOURCE	Field translation source	0	-----
2015	MSGSUBTYPE	Message subtype	0	-----
2016	APPCONFIG	Application configuration values	0	-----
2017	NOTEKTYPE	Note Type	0	-----
2018	NOTESTATUS	Note Status	0	-----
2019	SOURCE	Source	0	-----
2020	CREATEABLE_BY	Creatable By	0	-----
2021	STATE	Object State	0	-----

5. Scroll down in this list to 100004. This is the ID for the COUNTRY_STATE list.

Note: All IDs shown in this list, except for 2014, can be dynamically translated with the tilde (~) character. See Example 3 for instructions on how to translate 2014.

Another method of finding a list ID is to enter the spr_list query with the tag name. Tag is the name Admin uses for field name.

To find the tag name in Admin:

1. In Admin, select **General**. Then select **List**. The List Administration dialog box appears.
2. Select a list name, such as Availability, and click the **Modify** button. The Modify List Types dialog box appears. It shows that the tag for this list is AVAILABILITY.

Modify List Types

Details

Tag: AVAILABILITY

Display: Availability

Value: 0

Description: Availability

Properties

Cache to Client

Reset Ok Cancel

Use the SQL Query Analyzer and the query spr_list to find the ID for the tag AVAILABILITY.

1. In the Query window, enter the following query:
`spr_list AVAILABILITY`
2. Press F5 or the Play button to execute the query. The list shows that the ID for AVAILABILITY is 2022.

Query - (local)Acenza_Insurance.ss - (untitled) - spr_list AVAILA...

spr_list AVAILABILITY

```

-->> $REPORT$ List items.

```

ID	Tag	Display	Value	Type 1
2022	AVAILABILITY	Availability	0	
4201	FREE	Free	0	AVAILABILITY
4202	TENTATIVE	Tentative	1	AVAILABILITY
4203	BUSY	Busy	2	AVAILABILITY
4204	OUT	Out of Office	3	AVAILABILITY

(5 row(s) affected)

Query batch completed. Exec time: 0:00:00 5 rows ln 1, Col 22

Finding the ID in Example 4 is complicated because COUNTRY_STATE is a hierarchical list. There are two countries listed under it and all the states are listed under their respective countries. When the composite rule translates, it must have the ID of the top-level list. This is true for all hierarchical lists.

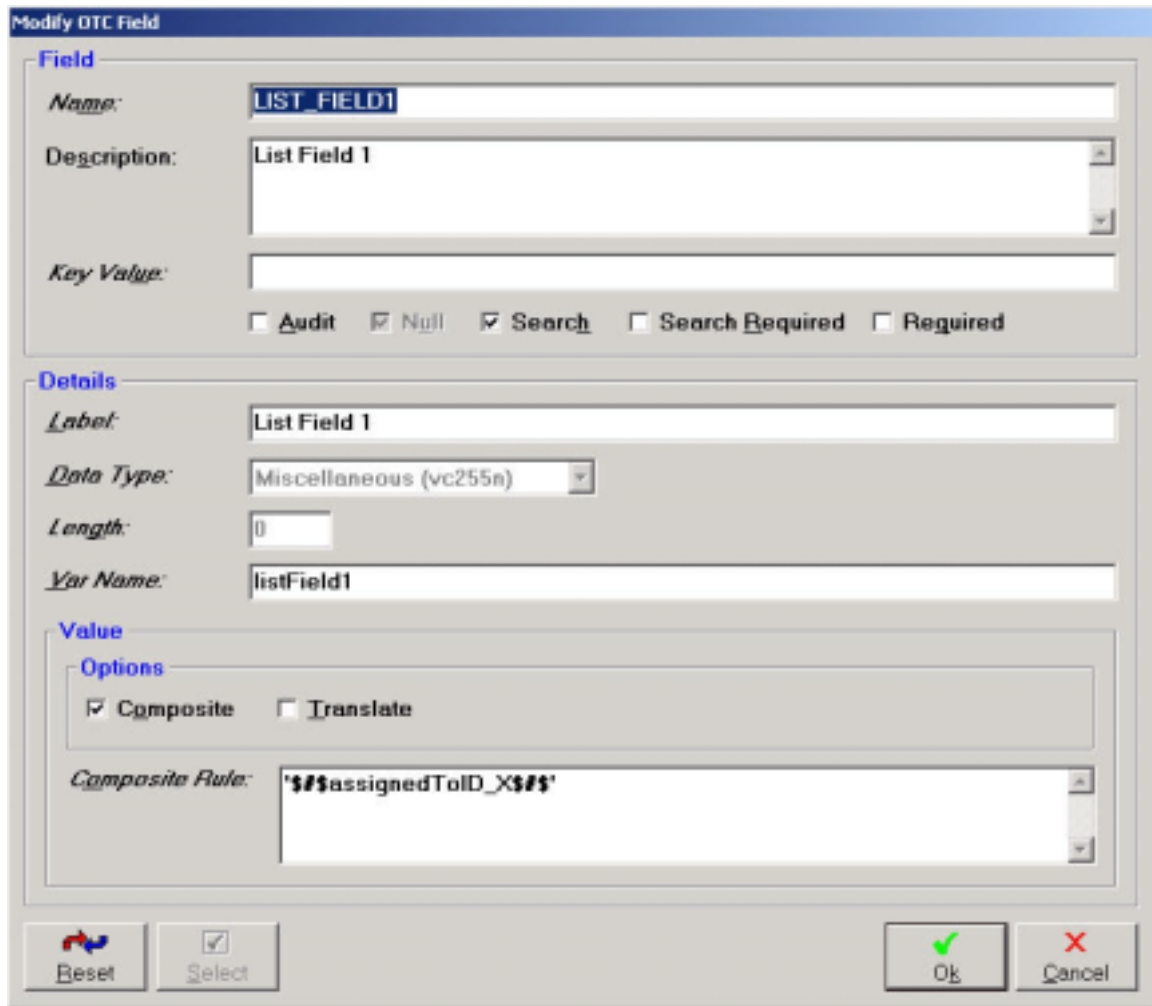
Example 5: Dynamic Translation of System Variables

This method finds if an OT is assigned to a particular user.

Scenario

To create the composite rule:

1. In Admin, select **Objects**. Then select **Tasks**. The Contact Administration dialog box appears.
2. Select an OTC, such as **Claim Tasks**, and click the **Modify** button. The Modify Object Type Class (OTC) dialog box appears.
3. Click the **Fields** button to expose the OTC Fields Administration dialog box.
4. Select ListField 1. Then click the **Modify** button. The Modify OTC Field dialog box appears.



Modify DTC Field

Field

Name: LIST_FIELD1

Description: List Field 1

Key Value:

Audit Null Search Search Required Required

Details

Label: List Field 1

Data Type: Miscellaneous (vc255n)

Length: 0

Var Name: listField1

Value

Options

Composite Translate

Composite Rule: '\$\$\$assignedToID_X\$\$\$'

Reset Select Ok Cancel

5. Under Options, select the **Composite** check box.
6. In the Composite Rule text box, enter the following composite rule:
`\$\$\$assignedToID_X\$\$\$`
7. Click OK to close the dialog box.
8. Update the cache by selecting **Reset Cache Information** from the File menu or by exiting Admin.
9. Run Merlin, steps 4 and 7, to generate the object and update the cache.
10. Open the Acenza WebTop.
11. Click the **New** button and create a Task of the same type that you created in Admin. That is, create a Claims Task and take ownership of it.
12. Click the **Search** button and search for Tasks, Claim Tasks.

13. Verify that your Search Result List displays a task that you owned. In the following figure, the task was assigned to a user who logged on as George Bush.



Notes on Example 5

Example 5 involves the dynamic translation of a special system variable that must be translated with the special characters \$#\$. There are four system variables that can be translated with \$#\$:

- assignedToID
- statusID
- objStateID
- ownerID

These variables are different for two reasons:

- They require a different translation method (\$#\$)
- A user cannot modify them

Note: For a composite rule that uses one of these system variables to work correctly, the characters “_X” must be appended to the end of the variable name, such as in ownerID_X. The “X” is required for the system to translate the ID to an actual value.

As in previous examples, the result appears in the Primary ID column. This column is mapped to ListField1. To determine how columns are configured or to change a configuration, open Admin and select **General, Group List Preferences**. Alternatively, open the WebTop and select **Preferences, Café, List Preferences**.

Example 6: Dynamic SQL

This method displays the OTC name of an object.

Scenario

To create the composite rule:

1. In Admin, select **Objects**. Then select **Documents**. The Document Administration dialog box appears.
2. Select an OTC, such as **Application Docs**, and click the **Modify** button. The Modify Object Type Class (OTC) dialog box appears.
3. Click the **Fields** button to expose the OTC Fields Administration dialog box.
4. Select ListField 5. Then click the **Modify** button. The Modify OTC Field dialog box appears.

Modify DTC Field

Field

Name: LIST_FIELD5

Description: List Field 5

Key Value:

Audit Null Search Search Required Required

Details

Label: List Field 5

Data Type: Miscellaneous (vc255n)

Length: 0

Var Name: listField5

Value

Options

Composite Translate

Composite Rule: NAME from ADM_OBJ_TYPE_CLASS where KEY_VALUE = 'CLAIM'

Reset Select Ok Cancel

5. Under Options, select the **Composite** check box.
6. In the Composite Rule text box, enter the following SQL statement as a composite rule:
NAME from ADM_OBJ_TYPE_CLASS where KEY_VALUE = 'CLAIM'
7. Click OK to close the dialog box.
8. Update the cache by selecting Reset Cache Information from the File menu or by exiting Admin.
9. Run Merlin, steps 4 and 7, to generate the object and update the cache.
10. Open the Acenza WebTop.

- Click the **New** button. Select **Document** as the Object, **Claims Division** as the Business, **Claim Correspondence Docs** and the Category, and **Insured Letter Claim** as the Type. Click **Continue**. The Create Document dialog box displays.



- Complete the required fields in the Create Document dialog box and click **Create**.
- Click the Search button and enter **Document**, **Claims Division**, **Claim Correspondence Docs**, and **Insured Letter Claim** in the required fields. Click **Continue**.
- Verify that Claim Letter Photo appears in the Search Result List.



ID	Type	Primary ID	Status	Attachment	Attachment ID
53	Insured Letter Claim	Claim Letter Photo	Active		

Notes on Example 6

Example 6 presents a simple use of dynamic SQL in a composite rule. You can create more complex rules with any valid SQL statement that uses typical SQL operations.



Changes to This Documentation

Updating and improving our documentation is an ongoing effort. To assure that you have the latest versions, you may want to visit the FileNET Worldwide Support web site (<http://www.css.filenet.com/>) from time to time.

At the site, click Documentation, log on, and navigate to Acenza. (If you do not have a CSS Web Account, click the New User button and follow the online instructions.)

Trademarks

Acenza is a trademark of FileNET Corporation. Microsoft and Windows are registered trademarks, and Windows NT and Internet Explorer are trademarks of Microsoft Corporation. All other products mentioned in this document are trademarks or registered trademarks of their respective owners.

Copyright

The content of this document, including all illustrations, is copyrighted © 2001, by FileNET Corporation. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage or retrieval systems, for any purpose other than the purchaser's personal use without the express written permission of FileNET Corporation. Federal and international laws governing intellectual property apply. All rights reserved.