

# **Unica Campaign V12.1.8 Administrator's Guide**



# Contents

- Chapter 1. Unica Campaign administration overview..... 17**
  - Logging in to Unica..... 19
- Chapter 2. Security in Unica Campaign..... 21**
  - How security policies work..... 21
  - The global security policy..... 23
  - How users are assigned to roles and security policies..... 23
  - The Owner and Folder Owner roles in security policies..... 24
  - Definitions of permission states..... 24
  - Guidelines for designing security policies..... 24
  - How Unica Campaign evaluates permissions..... 25
  - Security scenarios..... 26
    - Scenario 1: Allows access to all other employees' folders and objects..... 26
    - Scenario 2: Allows access to only some other employees' folders and objects..... 28
  - Implementing security policies..... 30
    - Creating security policies..... 30
    - Creating security roles..... 31
    - Deleting security policies..... 31
  - Reference: Administrative permissions in Unica Campaign ..... 32
    - Administration..... 32
    - Audience levels..... 34
    - Data sources..... 34
    - Dimension hierarchies..... 35
    - History..... 35
    - Logging..... 35
    - Reports (folder permissions)..... 36
    - System tables..... 36
    - User tables..... 37
    - User variables..... 38
  - Windows™ impersonation administration..... 38
    - Setting up Windows impersonation..... 39

How Unica Campaign uses data filters to restrict access to customer data.....	40
<b>Chapter 3. Database table administration.....</b>	<b>44</b>
Table administration concepts.....	44
What are system tables?.....	44
What are user tables? .....	44
About table mapping.....	45
Using flat files as data sources.....	46
Initial table administration tasks.....	46
Testing system table access.....	47
Testing user table access.....	47
User table administration.....	48
Adding new user data sources to Unica Campaign.....	48
Creating new user tables through an output process.....	50
Accessing user data sources from within a flowchart.....	51
Supported data types for user tables.....	52
Using Amazon Redshift user data sources with Unica Campaign.....	56
Using MariaDB datasources with Unica Campaign.....	56
Using Postgres datasources with Unica Campaign.....	57
Google BigQuery ODBC.ini configuration.....	58
Using Hive-based Hadoop big data sources with Unica Campaign.....	62
Using Cloudera Impala-based Hadoop big data sources with Unica Campaign.....	65
Mapping and unmapping user tables.....	67
Filtering the mapped user tables.....	77
System table administration.....	77
Mapping or remapping system tables.....	77
Unmapping system tables.....	78
Mapping Customer audience level system tables.....	78
About mapping segment membership tables.....	79
About mapping STO tables.....	80
Unmapping the segment membership table.....	80
Viewing system table contents.....	81
Data dictionary administration.....	81

What is a data dictionary?.....	81
Editing data dictionaries.....	82
Creating data dictionaries.....	82
Data dictionary syntax.....	83
Table catalog administration.....	84
What is a table catalog?.....	84
Creating table catalogs.....	84
Loading stored table catalogs.....	86
Deleting table catalogs.....	86
Updating pre-computed profiles for tables in a table catalog.....	87
Defining data folders for table catalogs.....	88
Housekeeping of Campaign table catalogs.....	88
Setting up Unica Campaign to use database load utilities.....	91
Amazon Redshift - Loader configurations .....	94
MariaDB - Loader configurations.....	97
Google BigQuery - Loader Configuration.....	99
PostgreSQL - Loader Configuration.....	100
Actian Db - Loader configurations.....	103
Fast loader repeated tokens.....	105
Using database load utilities with DB2 on z/OS.....	107
Troubleshooting database load utilities for Unica Campaign.....	107
Archiving campaigns and flowcharts.....	109
<b>Chapter 4. Customizing campaigns.....</b>	<b>111</b>
Custom campaign attributes.....	111
Custom cell attributes.....	111
Custom offer attributes.....	111
What is a static attribute?.....	112
What is a hidden static attribute?.....	112
What is a parameterized attribute?.....	112
Creating or editing custom attributes.....	113
Defining corporate initiatives for marketing campaigns.....	117
Adding products.....	117

<b>Chapter 5. Offer template administration.....</b>	<b>119</b>
What is an offer?.....	119
What is an offer template?.....	119
Offer templates and security.....	120
Planning offer templates and offers.....	120
Using custom attributes in offer templates.....	121
Standard offer attributes in Unica Campaign.....	121
Creating or editing custom attributes.....	122
Working with offer templates.....	126
Creating offer templates.....	126
Modifying offer templates.....	128
Using drop-down lists in offer templates.....	129
Defining a list of outbound communication channels.....	129
Changing the order in which offer templates appear.....	130
Retiring offer templates.....	130
Template icons.....	130
Default offer attributes.....	131
Introduction to using Unica Plan assets in Unica Campaign offers.....	131
Guidelines for using Unica Plan assets in Unica Campaign offers.....	132
<b>Chapter 6. Audience level administration.....</b>	<b>133</b>
About audience levels.....	133
Why different audience levels are needed in Unica Campaign.....	134
The default Customer audience level.....	134
About additional audience levels and system tables.....	134
Default Customer audience level system tables.....	135
About audience levels and strategic segments.....	135
Audience level unique identifiers.....	136
Required fields in audience-level-specific tables.....	136
About audience levels and user tables.....	139
User tables with a single audience level.....	139
User tables with multiple audience levels.....	140
Workflow for setting up a new audience level.....	140

Task 1: Create the required database tables for each new audience level.....	140
Task 2: Create the new audience levels in Unica Campaign.....	141
Task 3: Map the Unica Campaign system tables to the database tables.....	142
Task 4: Map the user tables containing relevant data to the appropriate audience level(s).....	143
Task 5: Save the mapped tables in a table catalog.....	143
Removing an audience level.....	143
To remove an audience level.....	143
About global suppressions and global suppression segments.....	144
Switching audiences with global suppressions.....	144
About creating global suppression segments.....	144
Updating global suppression segments.....	146
Deleting global suppression segments.....	146
Logging for global suppressions.....	146
<b>Chapter 7. Contact history administration.....</b>	<b>147</b>
Contact history concepts.....	147
What is contact history?.....	147
What is detailed contact history?.....	148
What is contact status?.....	148
About updating contact status.....	148
How does contact history relate to audience levels?.....	148
How does contact history relate to database and system tables?.....	149
What is offer history?.....	149
What is treatment history?.....	149
Creating contact history tables for new audience levels.....	150
Adding contact status codes.....	150
Deleting contact status codes.....	151
Writing to contact history.....	152
Updating contact history.....	152
Clearing contact history.....	153
Default contact status codes.....	153
<b>Chapter 8. Response history administration.....</b>	<b>154</b>
Response history and response types.....	154

Action tables.....	155
Sample action table (UA_ActionCustomer).....	156
Creating response history tables for new audience levels.....	157
Setting the number of days after an offer expires to record responses.....	157
Adding response types.....	157
Default response types.....	158
Logging response history.....	160
<b>Chapter 9. Monitoring and controlling flowchart runs.....</b>	<b>161</b>
Configuring operational monitoring.....	161
Using the <b>All monitored runs</b> page to control flowchart runs.....	161
Refreshing the <b>All monitored runs</b> page display.....	162
Stopping a running flowchart.....	162
Suspending a running flowchart.....	162
Resuming a suspended flowchart.....	163
Flowchart states and valid actions.....	163
<b>Chapter 10. Dimension hierarchy administration.....</b>	<b>166</b>
What is a dimension hierarchy?.....	166
Why use dimension hierarchies?.....	166
About dimension hierarchies and cubes.....	167
About dimension hierarchies and database tables.....	167
Dimension hierarchy design guidelines.....	168
Managing dimension hierarchies.....	168
Creating dimension hierarchies.....	169
Loading stored dimension hierarchies.....	170
Editing dimension hierarchies .....	170
Updating dimension hierarchies.....	171
Removing dimension hierarchies.....	171
<b>Chapter 11. Trigger administration.....</b>	<b>172</b>
What is an inbound trigger?.....	172
Why use an inbound trigger?.....	172
Inbound triggers and the Schedule process.....	172
What is broadcasting?.....	173

What is an outbound trigger?.....	173
Synchronous outbound triggers.....	173
Asynchronous outbound triggers.....	173
Why use an outbound trigger?.....	174
Return values for outbound triggers.....	174
How are triggers defined?.....	174
Creating and managing triggers.....	174
Creating triggers.....	174
Editing or moving triggers.....	175
Deleting triggers.....	176
Organizing triggers in folders.....	176
Moving trigger folders.....	177
Editing trigger folders.....	177
Deleting trigger folders.....	177
Setting up outbound triggers.....	178
Setting up a process to execute an outbound trigger.....	178
Setting up a flowchart to execute an outbound trigger on success.....	178
Setting up a flowchart to execute an outbound trigger on failure.....	178
Setting up inbound triggers.....	179
To set up inbound triggers.....	179
Configuring the Schedule process to run with an inbound trigger.....	179
Broadcasting a trigger to all flowcharts in a campaign.....	180
Broadcasting a trigger to specific flowcharts.....	180
Broadcasting a trigger to all campaigns.....	180
Setting up the trigger utility on a remote Windows machine.....	181
Tokens supported by triggers.....	181
Unica Campaign trigger utility syntax and options.....	183
<b>Chapter 12. Unica Campaign log files.....</b>	<b>185</b>
Unica Campaign log file names and locations.....	185
Flowchart logs.....	187
Configuring flowchart logging.....	187
Viewing and analyzing flowchart log files.....	188



Flowchart log file structure.....	189
Clearing a flowchart log file.....	190
The Unica Campaign web application log.....	190
Configuring Unica Campaign web application logging.....	191
The Unica Campaign and Unica Deliver ETL log file.....	193
Using log4j to configure web application and Unica Deliver ETL logging .....	193
Viewing and configuring Unica Campaign listener and master listener logs.....	194
The Unica Campaign Server Manager log.....	195
The session utility log.....	196
The sessions log.....	196
The web connections log.....	196
The cleanup utility log.....	196
Windows™ event logs.....	196
<b>Chapter 13. JVM Parameters.....</b>	<b>198</b>
<b>Chapter 14. Unique code administration.....</b>	<b>201</b>
About campaign codes.....	201
Changing the campaign code format.....	201
About cell codes.....	201
Changing the cell code format.....	202
About offer and treatment codes.....	202
Changing the offer or treatment code format in an existing offer template.....	203
Code format requirements.....	203
Default code formats.....	204
About code generators.....	205
Default code generators in Unica Campaign.....	205
About custom code generators.....	205
Requirements for custom code generators.....	206
About configuring Unica Campaign to use custom code generators.....	206
About creating custom code generators.....	207
About outputting unique codes.....	207
About outputting errors.....	208
About placing the custom code generators.....	208

To specify the location of the custom offer code generator.....	208
Properties related to code generation.....	209
Parameters for the default campaign and cell code generators.....	210
Parameters for the default offer code generator.....	210
Parameters for custom code generators.....	211
<b>Chapter 15. Advanced settings for individual flowcharts.....</b>	<b>213</b>
Adjusting General settings for individual flowcharts.....	213
Save flowchart run results.....	213
Setting in-database optimization to improve flowchart performance.....	214
Disable global suppression for this flowchart.....	216
Y2K Threshold.....	216
Auto-save (during user configuration).....	217
Checkpoint (during flowchart execution).....	217
Maximum data errors allowed.....	218
Send trigger(s) on flowchart run errors.....	218
Send Trigger(s) on Flowchart Success.....	219
Adjusting server optimization settings for individual flowcharts.....	219
Unica Campaign Virtual Memory Usage.....	219
Disallow use of temporary tables for this flowchart.....	219
Adjusting test run settings for individual flowcharts.....	219
<b>Chapter 16. Unica Campaign integration with other products.....</b>	<b>221</b>
Overview of IBM Watson Marketing Assistant integration with Unica Campaign .....	222
Unica Campaign and Unica Centralized Offer Management integration.....	223
Setting up Unica Plan assets for use in Unica Campaign offers.....	226
Configuring Unica Deliver offer integration with Unica Campaign.....	227
Adjusting Unica Campaign response tables for Unica Deliver offer integration.....	229
Integrating IBM Digital Analytics and Unica Campaign.....	230
Configuring Digital Analytics to enable Unica Campaign integration.....	234
Creating and populating the translation table.....	235
Translation table data sources.....	236
Mapping the translation table.....	240
Troubleshooting IBM Digital Analytics and Unica Campaign integrations.....	241

Overview of Opportunity Detect integration with Unica Campaign.....	244
How to integrate Unica Campaign with Opportunity Detect.....	245
Unica Campaign and Unica Journeys integration.....	251
Overview.....	251
Journey process box.....	252
Configuring Campaign web.....	254
Steps to generate client certificates to connect to Kafka .....	255
Unica Campaign and Unica Link integration.....	256
Unica Campaign and ContactCentral Integration.....	259
Unica Campaign integration with Audience Central and Segment Central.....	260
Unica Campaign integration with Callout Process Box .....	261
<b>Chapter 17. Unica Campaign integration with Kafka.....</b>	<b>264</b>
Campaign\partitions\partition[n]\Kafka\ (KafkaTemplate).....	267
<b>Chapter 18. Unica Campaign listeners.....</b>	<b>271</b>
Definition of listener terms.....	271
Front-end and back-end components.....	272
Unica Campaign Listener (unica_aclsnr).....	272
Unica Campaign listener requirements.....	273
Unica Campaign listener syntax and options.....	273
Configuration settings for a single-node listener configuration.....	274
Configuration settings for a clustered listener configuration.....	275
Listener clustering.....	275
Diagram of listener clustering.....	276
Supported listener clustering configurations.....	277
The master listener.....	278
Master listener priority.....	278
Load based routing load balancing.....	279
Listener failover.....	279
Listener failover scenario 1: Non-master listener node fails.....	280
Listener failover scenario 2: Master listener node fails.....	280
Log files for clustered listeners.....	281
Shared network location for clustered listeners: campaignSharedHome.....	281

Utilities for clustered listeners.....	283
Starting and stopping Unica Campaign listeners.....	284
Installing the Unica Campaign listener as a Windows service.....	284
Starting the Unica Campaign listener manually.....	285
Stopping the Unica Campaign listener.....	285
<b>Chapter 19. Unica Campaign utilities.....</b>	<b>287</b>
Unica Campaign advanced search utility (advSrchUtil).....	287
Unica Campaign advanced search agent (advSrchAgent).....	288
Unica Campaign listener shutdown utility (svrstop).....	289
Unica Campaign svrstop utility reference.....	289
Using the svrstop utility to shut down the Unica Campaign listener.....	290
Using the svrstop utility to shut down the Unica Optimize listener.....	291
Unica Campaign Server Manager (unica_svradm).....	291
Running the Unica Campaign Server Manager (unica_svradm).....	292
Unica Campaign Server Manager commands (unica_svradm).....	292
Killing a running flowchart .....	301
Unica Campaign session utility (unica_acsesutil).....	302
Unica Campaign session utility syntax and options.....	302
Exporting and importing objects between servers.....	307
Backing up sessions.....	310
Updating record counts and lists of distinct values.....	310
Manipulating table catalogs.....	311
Documenting catalog contents.....	312
Unica Campaign cleanup utility (unica_acclean).....	313
Environment variables required for unica_acclean.....	313
Unica Campaign cleanup utility syntax and options.....	314
Unica Campaign cleanup utility use cases.....	316
Unica Campaign report generation utility (unica_acgenrpt).....	319
Use case: capture cell counts from flowchart runs.....	320
Unica Campaign report generation utility syntax and options.....	320
Parameters used with the unica_acgenrpt -p option.....	322
Database testing utilities.....	323

Using the cxntest utility.....	323
Using the odbctest utility.....	325
Using the db2test utility.....	326
Using the odbctest Utility to test connection with MariaDB.....	327
Using the oratest utility.....	330
<b>Chapter 20. Configuring the ACOOptAdmin tool.....</b>	<b>331</b>
<b>Chapter 21. Non-ASCII data in Unica Campaign.....</b>	<b>332</b>
About using non-ASCII data or non-US locales.....	332
About character encodings.....	332
About interactions with non-ASCII databases.....	332
About the multi-locale feature.....	333
Configuring Unica Campaign for a non-ASCII language or a non-US locale.....	335
Set the operating system language and regional settings.....	335
Set encoding parameters for the web application server (WebSphere® only).....	336
Set Unica Campaign language and locale property values.....	336
Un-map and re-map system tables.....	338
Test your database and server configurations.....	338
Configuring Unica Campaign for multiple locales.....	341
Before you begin: Unica Campaign must be installed.....	341
Configuring for multiple locales on SQL Server.....	341
Configuring for multiple locales on Oracle.....	342
Configuring for multiple locales on DB2®.....	344
Configuring for multiple locales on MariaDB.....	347
Configuring for multiple locales on Postgres.....	348
Configuring for multiple locales on OneDB.....	349
<b>Chapter 22. Unica Campaign configuration properties.....</b>	<b>351</b>
Campaign.....	351
Campaign   collaborate.....	352
Campaign   navigation .....	353
Campaign   caching.....	357
Campaign   partitions.....	360
Campaign   partitions   partition[n]   WatsonAssistant.....	360

Campaign   partitions   partition[n]   Deliver   contactAndResponseHistTracking.....	361
Campaign   partitions   partition[n]   Deliver   sendTimeOptimization.....	364
Campaign   partitions   partition[n]   Acoustic Campaign.....	366
Campaign   partitions   partition[n]   Acoustic Campaign   contactAndResponseHistTracking.....	370
Campaign   partitions   partition[n]   Coremetrics.....	372
Campaign   partitions   partition[n]   reports.....	374
Campaign   partitions   partition[n]   validation.....	376
Campaign   partitions   partition[n]   audienceLevels   audienceLevel .....	378
Campaign   partitions   partition[n]   audienceLevels   audienceLevel   field[n].....	378
Campaign   partitions   partition[n]   dataSources.....	379
Campaign   partitions   partition[n]   systemTableMapping .....	454
Campaign   partitions   partition[n]   server.....	454
Campaign   partitions   partition[n]   offerCodeGenerator.....	493
Campaign   partitions   partition[n]   UBX.....	494
Campaign partitions partition[n] Kafka Journey.....	495
Campaign   partitions   partition[n]   Link .....	497
Campaign   monitoring.....	498
Campaign   ProductReindex.....	500
Campaign   unicaACLlistener.....	501
Campaign   unicaACLlistener   node [n].....	510
Campaign   campaignClustering.....	514
Campaign   unicaACOOptAdmin.....	517
Campaign   server .....	519
Campaign   logging .....	519
Campaign   proxy.....	520
Campaign   FlowchartEvents.....	521
Campaign   FlowchartEvents  ActiveMQ.....	521
Report configuration properties.....	522
Reports   Integrations   Cognos [version].....	522
Reports   Schemas   [product]   [schema name]   SQL Configuration.....	526
Reports   Schemas   Campaign.....	527
Reports   Schemas   Campaign   Offer Performance.....	527

Reports   Schemas   Campaign   [schema name]   Columns   [Contact Metric].....	529
Reports   Schemas   Campaign   [schema name]   Columns   [Response Metric].....	530
Reports   Schemas   Campaign   Campaign Performance.....	531
Reports   Schemas   Campaign   Campaign Offer Response Breakout.....	533
Reports   Schemas   Campaign   Campaign Offer Response Breakout   Columns   [Response Type] .....	533
Reports   Schemas   Campaign   Campaign Offer Contact Status Breakout.....	534
Reports   Schemas   Campaign   Campaign Offer Contact Status Breakout   Columns   [Contact Status] ....	535
Reports   Schemas   Campaign   Campaign Custom Attributes   Columns   [Campaign Custom Column]....	536
Reports   Schemas   Campaign   Campaign Custom Attributes   Columns   [Offer Custom Column].....	537
Reports   Schemas   Campaign   Campaign Custom Attributes   Columns   [Cell Custom Column].....	538
Reports   Schemas   Interact.....	539
Reports   Schemas   Interact   Interact Performance.....	540
Reports   Schemas   Unica Deliver.....	541
<b>Chapter 23. Special characters in Unica Campaign object names.....</b>	<b>542</b>
Special characters not supported.....	542
Objects with no naming restrictions.....	543
Objects with specific naming restrictions.....	543
Naming restrictions for derived fields.....	543
<b>Chapter 24. Internationalization and character encodings.....</b>	<b>544</b>
Character encodings in Unica Campaign.....	544
Western Europe.....	544
Unicode encodings.....	545
Arabic.....	545
Armenian.....	545
Baltic Rim.....	545
Celtic.....	545
Central Europe.....	546
Chinese (both Simplified and Traditional).....	546
Chinese (Simplified).....	546
Chinese (Traditional).....	546
Cyrillic.....	546
English.....	547

Georgian.....	547
Greek.....	547
Hebrew.....	547
Icelandic.....	547
Japanese.....	547
Korean.....	548
Lao.....	548
Northern Europe.....	548
Romanian.....	548
Southern Europe.....	548
Thai.....	548
Turkish.....	548
Vietnamese.....	549
Miscellaneous.....	549
Date and time formats.....	549
Formats for DateFormat and DateTimeFormat.....	549
Formats for DateOutputFormatString and DateTimeOutputFormatString.....	553
<b>Chapter 25. Unica Campaign Error Codes.....</b>	<b>555</b>
List of Unica Campaign error codes.....	555
<b>Chapter 26. Troubleshooting.....</b>	<b>658</b>
<b>Index.....</b>	



# Chapter 1. Unica Campaign administration overview

The **Settings** menu provides access to most of the tasks that Unica Campaign administrators typically perform.

**Table 1. Templates and customization (Settings > Campaign Settings page)**

## Templates and customization

Option	Description
Custom attribute definitions	Define attributes that can be used in campaigns, offers, and cells. For example, you can define an offer attribute of "Interest Rate" to store values for mortgage offers.
Offer template definitions	An offer template defines the structure for an offer. Offer templates are required. Users cannot create an offer without basing it on a template.


**Table 2. Data source operations (Settings > Campaign Settings page)**

## Data source operations

Option	Description
Manage table mappings	<ul style="list-style-type: none"><li>• User tables contain data about your company's customers, prospects, or products, for use in marketing campaigns. You must map user tables or files to make the data accessible for use in flowcharts.</li><li>• System tables contain Unica Campaign application data. They are configured during installation.</li></ul>
View datasource access	View the system table database and all configured customer databases. Select a database to see its configuration details. Log in or out of customer databases.
Manage dimension hierarchies	Use dimension hierarchies to group data based on value ranges. Examples are age, income, product, and distribution channel. You can create any kind of hierarchy that relates to your business or campaign.
Manage audience levels	An audience level is an identifiable group that can be targeted by a marketing campaign. Examples are Household, Prospect, Customer, and Account. Flowchart designers can target and switch between audiences or scope one audience level by another. For example, you can target one individual per household.

**Table 2. Data source operations (Settings > Campaign Settings page)****Data source operations**

(continued)

Option	Description
View system log	<p>This option opens the Unica Campaign listener log (<code>unica_aclsnr.log</code>).</p> <p> <b>Note:</b> If there are multiple partitions, this option is not available, for security reasons.</p>

**Table 3. Other administrative tasks****Other administrative tasks**

Task	Description
Manage users, groups, role assignments, security policies, and permissions	<p>Use the <b>Settings</b> menu to adjust security and permissions.</p> <p>Instructions are provided in the Unica Platform Administrator's Guide and the Unica Campaign Administrator's Guide.</p>
Adjust configuration properties	<p>Choose <b>Settings &gt; Configuration</b> to access configuration properties.</p> <ul style="list-style-type: none"> <li>• Use the <b>Campaign</b> category to adjust properties for Unica Campaign.</li> <li>• Use the <b>Reports</b> category to adjust reporting properties.</li> <li>• Use the <b>General</b> and <b>Platform</b> categories to adjust properties that affect the Unica suite. For information, see the online help or the <i>Unica Platform Administrator's Guide</i>.</li> <li>• Configuration categories for other products, such as Unica Deliver, are described in the documentation for those products.</li> </ul>
Adjust settings for individual flowcharts	<p>Use the flowchart <b>Admin</b> menu to perform administrative actions for individual flowcharts.</p>
Manage contact history and response history	<p>Modify the contact history and response history system tables that are supplied with Unica Campaign to capture information about communications with your customers. Information is provided in the Unica Campaign Administrator's Guide.</p>

**Table 3. Other administrative tasks****Other administrative tasks**

(continued)

Task	Description
Run utilities to complete administrative functions	Use the command-line utilities to perform server, session, and database tasks.
Use the Scheduler to schedule flow-chart runs	See the Unica Platform Administrator's Guide.

**Table 4. Unica Campaign integration tasks****Unica Campaign integration tasks**

Task	Description
Install and configure reports that are Cognos® based	See the Unica Reports Installation and Configuration Guide that is supplied with Unica Platform.
Integrate Unica Campaign with other HCL products	See the following documents: <ul style="list-style-type: none"> <li>• Installation and upgrade guides</li> <li>• Integration guides supplied with the product that you are integrating</li> <li>• <i>Unica Platform Administrator's Guide</i></li> <li>• <i>Unica Campaign Administrator's Guide</i></li> </ul>

**Configuring the web application server for your JDBC driver**

For details on configuring the web application server for your JDBC driver, see the Campaign Installation Guide.

## Logging in to Unica

Use this procedure to log in to Unica.

You need the following.

- An intranet (network) connection to access your Unica server.
- A supported browser installed on your computer.

- User name and password to sign in to Unica.
- The URL to access Unica on your network.

The URL is:

```
http://host.domain.com:port/unica
```

where

*host* is the machine where Unica Platform is installed.

*domain.com* is the domain in which the host machine resides.

*port* is the port number where the Unica Platform application server is listening.



**Note:** The following procedure assumes that you are logging in with an account that has Admin access to Unica Platform.

Access the Unica URL using your browser.

- If Unica is configured to integrate with Windows™ Active Directory or with a web access control platform, and you are logged in to that system, you see the default dashboard page. Your login is complete.
- If you see the login screen, log in using the default administrator credentials. In a single-partition environment, use `asm_admin` with `password` as the password. In a multi-partition environment, use `platform_admin` with `password` as the password.

A prompt asks you to change the password. You can enter the existing password, but for good security you should choose a new one.

- If Unica is configured to use SSL, you may be prompted to accept a digital security certificate the first time you sign in. Click **Yes** to accept the certificate.

If your login is successful, Unica displays the default dashboard page.

With the default permissions assigned to Unica Platform administrator accounts, you can administer user accounts and security using the options listed under the **Settings** menu. To perform the highest level administration tasks for Unica dashboards, you must log in as **platform\_admin**.



**Note:** Starting with version 12.1.4 Unica Campaign has migrated to OpenSSL 3.x.

Although we expect all functionality to work normally, there may be cases that any third-party software running on same machine as Campaign server may not yet support OpenSSL 3.x.

In such cases it may be necessary to have installed Campaign version to switch to OpenSSL 1.x.

# Chapter 2. Security in Unica Campaign

Security policies control user access to the objects and features in Unica Campaign.

Administrators use the Unica Platform security interface to configure the user accounts, group memberships, roles, and permissions required for users to access Unica Campaign.

## Security terms

The following terms are used in discussing Unica Campaign security roles and policies.

### Security policy

A set of roles that defines security for folders and objects in Unica Campaign.

### Role

A set of permissions within a security policy that defines application access for users. Roles are typically aligned with job functions such as Administer, Review, Design, Execute.

### Permission

The access assigned to a role: Granted, Denied, or Not Granted.

### Application access

The set of actions that a user is allowed to perform within Unica Campaign.

### User

An account that allows an individual user to log in to Campaign. Accounts are managed in Unica Platform.

### Group

A collection of user accounts that have the same application access needs.

### Object

Items that users can create within Unica Campaign. Examples of objects are campaigns, offers, and templates.

## How security policies work

Security policies are the "rule books" that govern security for folders and objects in Unica Campaign. They are consulted each time a user performs an action in the application.

You can create your own security policies or use the default global security policy included with Unica Campaign.

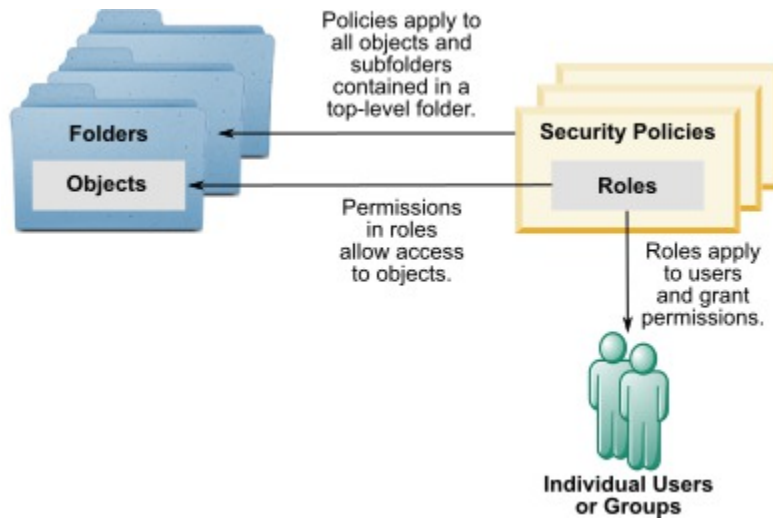
In Unica Campaign, security policies are assigned to folders. When you create a top-level folder, you are required to apply a security policy to the folder. Any objects or subfolders within that folder inherit the folder's security policy.

Because the top-level folder determines the security policy of the objects in the folder, you cannot directly assign a security policy to objects. To change the security policy of an object, you must move the object into a folder with the desired security policy or into the top-level root folder.

You also cannot directly assign a security policy to a user. Unlike objects and folders, which are assigned to security policies as a whole, users are assigned to roles within security policies. To control what users can do, you assign users to roles within security policies. In this way, you control user access to objects within folders that use those security policies.

If a user is not explicitly assigned to at least one role in a security policy, that user cannot create folders and objects under a top-level folder that uses that policy, and that user has no access to objects under that folder or its sub folders.

The following diagram illustrates the relationship between security policies, folders, objects, roles, and users.



### Top-level Administrative roles

Administrative roles in Unica Campaign are assigned for each partition. Users with these roles can perform the allowed actions on any objects within the partition, regardless of the security policy used in the folders that contain the objects.

### Security policies and partitions

Security policies are created per partition. There is no sharing of security policies across partitions.

Each partition in Unica Campaign can have multiple security policies.

### Security policy changes when folders and objects are moved or copied.

Objects and folders can be moved or copied across security policies, but the user performing the move or copy must have permissions to do so, in both the source and destination policies.

After an object or folder is moved or copied to a folder that is assigned to a different security policy from its source, the security policy of the lower-level objects or subfolders is automatically changed to the security policy of the new folder.

## The global security policy

Unica Campaign includes a default global security policy. You cannot delete this policy; it always applies. However, you can customize your security scheme as follows.

- Modify the roles and permissions in the global policy to suit the needs of your organization.
- Create custom policies and assign users only to your custom policies rather than the global policy.
- Use both custom policies and the global policy.

Any custom policy you create exists under the global policy. If you choose not to create your own security policies, the global security policy is applied by default to the folders and objects that users create in Unica Campaign.

The global security policy contains six pre-defined roles. You cannot delete the pre-defined roles, but you can modify their permissions.

The pre-defined roles in the global security policy are:

- **Folder Owner** - All permissions enabled for the folders a user has created. All users have this role; you do not need to assign users to it.
- **Owner** - All permissions enabled for the objects a user has created. All users have this role; you do not need to assign users to it.
- **Admin** - All permissions enabled. The default user `asm_admin` has this role.
- **Execute** - All permissions enabled.
- **Design** - Read and write permissions on all objects. This role cannot schedule flowcharts or sessions.
- **Review** - Read-only permissions.

## How users are assigned to roles and security policies

Unlike objects and folders, which are assigned to security policies as a whole, users are assigned to roles within security policies.

You can assign users to roles individually or through groups.

- You can assign a user to a role individually on the **Settings > User roles and permissions** page when you view the role details, or on the **Settings > Users > Edit roles** page for each user.
- You can assign users through a group by making the user a member of a group that is assigned to that role. See the *Unica Platform Administrator's Guide* for details on creating and using groups.

For large numbers of users, assigning roles through groups is the easier method to manage.

If your environment is integrated with an LDAP server such as Windows™ Active Directory, group memberships are imported from the LDAP server. Groups in Unica Platform are mapped to the groups on the LDAP server, and roles are assigned to these groups to manage application access. See the Unica Platform Administrator's Guide for details.

## The Owner and Folder Owner roles in security policies

The Owner and Folder Owner roles exist in the global policy and they are also created by default when you create a custom security policy. These roles automatically apply to all users who have been made members of a security policy by being explicitly assigned to any other role within that policy.

By default, the Owner role applies to all objects that a user creates, and it grants all permissions for those objects. The Folder Owner role applies to all objects in a folder that a user owns, and grants all permissions for those objects.

You can modify the permissions of these roles or use the default permissions.

See the scenarios for examples of how to design security policies that use the default Owner and Folder Owner roles to restrict user access within a security policy to only the objects and folders that they own.

## Definitions of permission states

For each role, you can specify which permissions are granted, not granted, or denied. You set these permissions on the **Settings > User roles and permissions** page.

These states have the following meanings.

- **Granted** - indicated with a check mark . Explicitly grants permission to perform this particular function as long as none of the user's other roles explicitly denies permission.
- **Denied** - indicated with an "X" . Explicitly denies permission to perform this particular function, regardless of any other of the user's roles which might grant permission.
- **Not granted** - indicated with a circle . Does not explicitly grant nor deny permission to perform a particular function. If this permission is not explicitly granted by any of a user's roles, the user is not allowed to perform this function.

## Guidelines for designing security policies

Follow these guidelines when designing security policies for Unica Campaign.

### Keep the design simple

Unica Campaign allows you to create multiple security policies and roles, but you should keep the security design as simple as possible, and use as few policies and roles as possible to achieve your security needs. At the most minimal level, for example, you can use the default global security policy as is, without creating new roles or policies.



## Avoid potential conflicts among security policies

If your organization implements more than one security policy, keep in mind potential conflicts when designing the policies.

For example, users with Move and Copy permissions in more than one security policy are able to move or copy objects and folders to locations across the policies in which they have these permissions. In doing so, because the moved objects or folders take on the security policy of their destination (if under another folder), they might cause situations where the rightful users in one division are no longer able to access the moved objects because they have no roles in the destination security policy, or where users with roles in the destination security policy who were not intended to access the objects, find that they now can.

## Assign view permissions to allow users to modify objects

To modify the following objects in Unica Campaign, grant users both view and modify permissions for the object:

- campaigns
- flowcharts
- offers
- offer lists
- offer templates
- sessions
- strategic segments

## How Unica Campaign evaluates permissions

When a user performs a task or tries to access an object, Unica Campaign performs the following steps.

1. Identifies all groups and roles to which this user belongs within the global security policy.

Users can belong to one, many, or no roles. Users belong to the Owner role if they own an object; they belong to the Folder Owner role if they own the folder in which an object resides.

Users belong to other roles only if they have been specifically assigned to that role (either directly or because they belong in a group assigned to that role).

2. Identifies whether the object being accessed is assigned to a custom-defined policy. If so, the system identifies all groups and roles to which the user belongs within this custom policy.
3. Aggregates the permissions for all roles to which the user belongs, based on results from steps 1 and 2. Using this composite role, the system evaluates the permissions for the action are evaluated as follows:

- a. If any roles have **Denied** permission for this action, then the permissions are aggregated as follows:
  - i. Consider a Global Policy, 1 Custom Policy, and a permission DENIED for the Custom Policy role. Then, DENIAL of any permission for a Custom policy Role takes precedence over permissions assigned to the Global Policy Role.
  - ii. Consider a Global Policy, 2 or more Custom Policies, a permission DENIED for one of the Custom policy roles, and the same permission GRANTED to the other Custom policy role. Then, GRANT of any permission of Custom policy takes precedence over DENIAL of permission of the Custom policy.
- b. If no roles have **Denied** permission for this action, then it checks to determine whether any roles have **Granted** permission for this action. If so, the user is allowed to perform the action.
- c. If neither a nor b is true, the user is denied the permission.

### Example for one custom policy

Consider one custom policy under Global Policy : CustomPolicyA. CustomPolicyA has CustomPolicyARole, that has Add/Edit Unica Campaign permission DENIED.

Consider UserA who has CustomPolicyARole assigned. DENIAL of Add/Edit Unica Campaign permission for a CustomPolicyARole takes precedence over permissions assigned to the Global Policy Role. Hence, the Add/Edit Unica Campaign objects are not visible to UserA.

### Example for two custom policies

Consider two custom policies under Global Policy: CustomPolicyA and CustomPolicyB. Both CustomPolicyA and CustomPolicyB have CustomPolicyARole and CustomPolicyBRole respectively. CustomPolicyARole has Add/Edit Unica Campaign permission GRANTED. CustomPolicyBRole has Add/Edit Unica Campaign permission DENIED.

UserA has both CustomPolicyARole and CustomPolicyBRole assigned. GRANT of Add/Edit permission of CustomPolicyARole takes precedence over DENIAL of permission of the CustomPolicyBRole. Hence, the Add/Edit Unica Campaign objects are visible to UserA.

## Security scenarios

This section provides security policy examples and explains how they can be used to address common security needs.

### Scenario 1: Allows access to all other employees' folders and objects

All employees in your company work with the same set of objects (campaigns, offers, templates, and so on). Sharing and re use of objects are encouraged; there is no need to make sure that groups of employees cannot access each other's objects. Access is restricted only by the employees' roles within the organization.

**Solution: Use the global security policy**

Only a single security policy is required, as objects do not have to be separated by group or division. In the existing global security policy, examine the default roles and modify them as needed to correspond to the requirements for the employee jobs. You can also create custom roles as needed.

The default Owner and Folder Owner roles automatically allow users full permissions on the objects they create. You can define additional roles to restrict access to objects created by other users.

For example, the following table illustrates a subset of the permissions you can configure. In the example, a Manager has full access and editing permissions for campaigns and offers. A Reviewer can view campaigns and offers, but cannot perform any other action with them.

After you define roles, assign employees to the roles that correspond to their job requirements. You can assign employees individually, or by creating several groups. Assign each group to a separate role and make employees members of the group that has the role appropriate for their job.

**Table 5. Scenario 1: Object permissions by role**

	Folder Owner role	Owner role	Manager role	Designer role	Reviewer role
Campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Add campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Edit campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Delete campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Run campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• View campaign summary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Table 5. Scenario 1: Object permissions by role (continued)**

	Folder Owner role	Owner role	Manager role	Designer role	Reviewer role
• Add campaign folders	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• View batch flow-charts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Add offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Edit offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Delete offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Retire offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• View offer summary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### Scenario 2: Allows access to only some other employees' folders and objects

Your company has two business divisions, Eastern and Western, that do not share data between them. Within each division, people performing different jobs need to access the same objects (campaigns, offers, templates) but with differing permissions to act on these objects, depending on their job. Access is restricted both by the employees' roles within the organization and by their division.

**Solution: Create a custom security policy for each division**

Define two separate security policies, one for each division. Each policy has the roles and permissions appropriate for its division.

For most employees, assign roles within their division's policy only. Do not assign any role in the global policy. Create top-level folders that belong to each policy, to hold campaigns, offers, and so on. These folders are specific to each division. Users with roles in one policy cannot see the objects belonging to the other policy.

The default Owner and Folder Owner roles automatically allow users full permissions on the objects they create. The other roles you define can allow restricted access to the objects created by other users within the same division and policy.

For employees who need to work across both divisions (for example, the controller, cross-divisional managers, or the CEO), assign a role in the global policy and modify it if necessary to grant the desired permissions. Users with roles in the global policy can see the objects in both divisions.

The following table illustrates a subset of the roles and permissions you can configure for a division's security policy.

**Table 6. Scenario 2: Example policy for one division**

	Folder Owner role	Owner role	Manager role	Designer role	Reviewer role
Campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Add campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Edit campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Delete campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• View campaign summary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Table 6. Scenario 2: Example policy for one division (continued)**

	Folder Owner role	Owner role	Manager role	Designer role	Reviewer role
• View batch flow-charts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Add offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Edit offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• Delete offers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
• View offer summary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## Implementing security policies

You can create and delete security policies in Unica Campaign and apply security policies to folders and objects.



**Note:** You must have permission to administer the User Roles and Permissions page in Unica Platform to work with Unica Campaign security policies. In a multi-partition environment, only the platform\_admin user, or another account with the PlatformAdminRole role, can work with security policies in all partitions.

## Creating security policies

Follow these steps to create security policies. Each partition in Unica Campaign can have one or more security policy.

1. Click **Settings > User roles and permissions**.
2. Under the **Campaign** node, select the partition where you want to add a security policy.
3. Click **Global policy**.
4. On the right of the page, click **Add policy**.

5. Enter a policy name and description.
6. Click **Save changes**.

The new policy is listed under the Global Policy on the User Roles and Permissions page. By default, the policy contains a Folder Owner role and an Object Owner role.

## Creating security roles

Follow these steps to create security roles. Each security policy in Unica Campaign can have one or more roles.

1. Click **Settings > User roles and permissions**.
2. Under the **Campaign** node, select the partition where you want to add a role.
3. Click the policy where you want to add a role.
4. On the right of the page, click **Add roles and assign permissions**.
5. Click **Add a role**.
6. Enter a role name and description.
7. Click **Save and edit permissions**.

The complete set of role permissions is listed in edit mode.

8. Set permissions as needed and click **Save changes**.

The new role is listed under the policy.

## Deleting security policies

You can delete user-created security policies in Unica Campaign if the policies are not in use. You cannot delete the Global Policy.

Do not delete any security policies that have been applied to objects in Unica Campaign.

To delete a security policy that is in use, first set the security policy of every folder or object that is using that security policy to a different policy (for example, the global policy). Otherwise, the objects using the deleted policy might become inaccessible. To change the security policy of an object, you must move the object into a folder with the desired security policy or into the top-level root folder.

Follow the steps below to delete a security policy that is not being used.

1. Click **Settings > User roles and permissions**.
2. Under the **Campaign** node, select the partition where you want to delete a security policy.
3. Click the plus sign next to **Global policy**.
4. Click the policy that you want to delete.
5. Click **Delete policy**.
6. Click **OK** to confirm the deletion.

## Reference: Administrative permissions in Unica Campaign

For each partition, you can assign administrative permissions to determine feature access by role. For example, you can allow the Design role to view flowchart logs but not clear the logs.

Each partition includes four pre-defined Administrative roles:

- **Admin:** All permissions enabled. The default user `asm_admin` is assigned this role.
- **Execute:** Most permissions enabled, with the exception of administrative functions, such as performing cleanup operations, changing object/folder ownership, running the `genrpt` Command Line Tool, Manage Global Suppressions, and Disable Suppression in Flowchart.
- **Design:** Same permissions as the Execute role.
- **Review:** Read-only access to all objects. For flowcharts, these users are allowed to access the edit mode of a flowchart, but save is not allowed.

You can add other administrative roles for each partition as needed.

To access administrative permissions, choose **Settings > User roles and permissions**. Under the **Campaign** node, select a partition. Click **Add roles and assign permissions**. On the **Properties for administrative roles** page, click **Save and edit permissions**.

Unica Campaign includes administrative permissions in the following categories:

- Administration
- Audience Levels
- Data Sources
- Dimension Hierarchies
- History
- Logging
- Reports (folder permissions)
- System Tables
- User Tables
- User Variables





**Note:** To set permissions for all features in a category, click the category heading box. For example, click the box next to **Logging** to adjust all logging settings at the same time.

### Administration

The permissions in the Administration category provide access to reports, tools, and utilities that can have system-wide effects on Unica Campaign.



**Table 7. Administration (Administrative permissions)**

Permission	Description
Access monitoring area	Allows access to the Campaign Monitoring area.
Perform monitoring tasks	Allows users to use the monitoring tasks in the Campaign Monitoring area.
Access analysis area	Allows access to reports in the Campaign Analytics area.
Access optimizations link	If Unica Optimize is installed, allows access to that application.
Run <code>svradm</code> Command line tool	Allows a user to use the Unica Campaign Server Manager ( <code>unica_svradm</code> ) for administrative functions.
Run <code>genrpt</code> Command line tool	Allows running of the Unica Campaign report generation utility ( <code>unica_acgenrpt</code> ).
Takeover flowcharts in edit mode	Allows taking over control of flowcharts in <b>Edit</b> or <b>Run</b> mode from other users.   <b>Note:</b> Taking over control of a "locked" flowchart locks out the other user and all changes in the flowchart since the last save are lost.
Connect to running flowcharts	Allows attaching to running flowcharts through Unica Campaign Server Manager ( <code>unica_svradm</code> ) or the Unica Campaign user interface.
Terminate server processes	Allows a user to stop the Unica Campaign Server ( <code>unica_acsvr</code> ) by using the Unica Campaign Server Manager ( <code>unica_svradm</code> ).
Terminate campaign listener	Allows a user to stop the Unica Campaign Listener ( <code>unica_aclsnr</code> ) by using the Unica Campaign Server Manager ( <code>unica_svradm</code> ) or using the <code>svrstop</code> utility.
Run <code>sesutil</code> Command line tool	Allows a user to run the Unica Campaign session utility ( <code>unica_acsesutil</code> ).
Override virtual memory Settings	Allows a user to override the Virtual Memory setting in flowchart Advanced Settings.
Access custom attributes	Allows access to, and managing of, custom attribute definitions from the <b>Campaign Settings</b> page.
Cell report access	Allows access to cell reports from the <b>Reports</b> icon  on a flowchart <b>Edit</b> page. Excludes access to the Cell Content Report unless this permission is also explicitly granted.
Cell report export	If cell report access is granted, allows printing and exporting of cell reports.
Cell content report Access	Allows access to the Cell Content report from the <b>Reports</b> icon on a flowchart <b>Edit</b> page.

**Table 7. Administration (Administrative permissions) (continued)**

Permission	Description
Cell content report export	If Cell Content Report Export is granted, allows printing and exporting of the Cell Content report.
Perform cleanup operations	Allows a user to use <code>unica_acclean</code> or a custom tool for cleanup operations.
Change object/folder ownership	Allows a user to change the ownership of an object or folder.

## Audience levels

The permissions in this category allow manipulation of audience levels, which represent targets of campaigns, such as Customer or Household.

**Table 8. Audience levels (Administrative permissions)**

Permission	Description
Add audience levels	Allows creation of new audience levels under <b>Manage audience levels</b> on the Campaign Settings page.
Delete audience levels	Allows deleting of existing audience levels under <b>Manage audience levels</b> on the Campaign Settings page.
Manage global suppressions	Allows creation and configuration of global suppression segments in Unica Campaign.
Disable suppression in flowchart	Allows clearing or selecting the <b>Disable global suppressions for this flowchart</b> check box on the flowchart Advanced Settings dialog.

## Data sources

The permissions in this category affect access to data sources.

**Table 9. Data sources (Administrative permissions)**

Permission	Description
Manage datasource access	Allows managing data source logins from the Administration area and within flowcharts.
Set save with DB authentication	Allow enabling the <b>Save with database authentication information</b> flag in table catalogs and flowchart templates.

## Dimension hierarchies

The permissions in this category allow manipulation of dimension hierarchies, which can be used in reports and Cubes.

**Table 10. Dimension hierarchies (Administrative permissions)**

Permission	Description
Add dimension hierarchies	Allows creation of new dimension hierarchies.
Edit dimension hierarchies	Allows editing of existing dimension hierarchies.
Delete dimension hierarchies	Allows deletion of existing dimension hierarchies.
Refresh dimension hierarchies	Allows refresh of existing dimension hierarchies.

## History

The permissions in this category affect logging to Contact History and Response History tables.

**Table 11. History (Administrative permissions)**

Permission	Description
Log to contact history Tables	Allows enabling or disabling logging to contact history tables when configuring contact processes.
Clear contact history	Allows clearing entries from the contact history tables.
Log to response history Tables	Allows enabling or disabling logging to response history tables when configuring the Response process.
Clear response history	Allows clearing entries from the response history tables.

## Logging

The permissions in this category affect manipulation of system and flowchart logs and options.

**Table 12. Logging (Administrative permissions)**

Permission	Description
View system and flowchart Logs	Allows viewing of flowchart logs and the system log
Clear flowchart logs	Allows clearing of flowchart logs.
Override flowchart log Options	Allows override of default flowchart logging options.

## Reports (folder permissions)

The Reports node appears on the partition permissions page after running **Sync report folder permissions** from the **Settings** menu for the first time. The synchronize process determines the folder structure of the reports physically located on the IBM® Cognos® system, and then lists the names of those folders under this node.

The settings under this node grant or deny access to the reports in the folders that appear in the list.

### Configuring report folder permissions

In addition to controlling access to the **Analytics** menu item and the **Analysis** tabs for object types (campaigns and offers, for example), you can configure permissions for groups of reports that are based on the folder structure in which they are physically stored on the IBM® Cognos® system.

Before you run Sync Report Folder Permissions, you must ensure that the following conditions exist:

- Reporting is enabled.
- The Cognos® server where reports are configured is up and running.

Complete the following steps to configure report folder permissions:

1. Log in as a Unica Campaign administrator who has the **ReportSystem** role.
2. Select **Settings > Sync report folder Permissions**.

The system retrieves the names of the folders that are located on the IBM® Cognos® system, for all partitions. (This means that if you decide to configure folder permissions for any partition, you must configure it for all of them.)

3. Select **Settings > User roles and permissions > Campaign**.
4. Under the **Campaign** node, select the first partition.
5. Select **Add roles and assign permissions**.
6. Select **Save and edit permissions**.
7. On the **Permissions** form, expand **Reports**.

The Reports entry does not exist until after you run the **Sync report folder permissions** option for the first time.

8. Grant permission for **Performance reports** to the appropriate role.
9. Configure the access settings for the report folders appropriately and then save your changes.
10. Repeat steps 4 through 8 for each partition.

## System tables

The permissions in this category control the ability to map, unmap, and manipulate Unica Campaign system tables.

**Table 13. System tables (Administrative permissions)**

Permission	Description
Map system tables	Allows mapping system tables.
Remap system tables	Allows remapping system tables.
Unmap system tables	Allows unmapping system tables.
Delete system table records	Allows deletion of records from system tables.

## User tables

The permissions in this category control the ability to map, unmap, and manipulate Unica Campaign user tables. User tables contain data about customers and prospects for use in flowcharts.

**Table 14. User tables (Administrative permissions)**

Permission	Description
Map base tables	Allows mapping base tables.
Map dimension tables	Allows mapping dimension tables.
Map general tables	Allows mapping general tables.
Map delimited files	Allows mapping user tables to delimited files.
Map fixed-width flat files	Allows mapping user tables to fixed-width flat files.
Map database tables	Allows mapping user tables to database tables.
Remap user tables	Allows remapping of user tables.
Unmap user tables	Allows unmapping of user tables.
Recompute counts and values	Allows using <b>Compute</b> button in table mapping to recompute table counts and values.
Use raw SQL	<p>Allows the use of raw SQL in Select process queries, custom macros, and dimension hierarchies.</p> <p>It is strongly recommended that you restrict the ability to use raw SQL to a user who has proper ACL privileges within your organization and disable this functionality from the server side for other users. The ability to use raw SQL in a flowchart process box presents an inherent security risk if not properly managed. Malicious use of raw SQL and stored procedures could impact the availability and security of the system. To reduce the risk of data or privacy loss, administrators are strongly advised to restrict user queries to a set of predefined or precompiled prepared statements and stored proce-</p>

**Table 14. User tables (Administrative permissions) (continued)**

Permission	Description
	dures. Also, restrict confidential and sensitive data from the user database to the Unica Campaign application.

## User variables

The permissions in this category control the ability to manipulate user variables, which can be used in queries and expressions in flowchart processes.

**Table 15. User variables (Administrative permissions)**

Permission	Description
Manage user variables	Allows creating, deleting, and setting default values for user variables in flowcharts.
Use user variables	Allows use of user variables in output files or tables.

## Windows™ impersonation administration

Windows™ impersonation is a mechanism that allows Unica Campaign administrators to associate Unica Campaign users with Windows™ users, so that Unica Campaign processes invoked by an Unica Campaign user run under the credentials of the corresponding Windows™ user.

For example, if Windows™ impersonation is enabled, when the Unica Campaign user `jsmith` edits a flowchart, a `unica_acsvr` process starts under the Windows™ user ID associated with the Unica Platform login name `jsmith`.

### Why use Windows™ impersonation?

By using Windows™ impersonation, you are able to leverage the Windows™-level security permissions for file access. If your system is set up to use NTFS, you can then control access to files and directories for users and groups. Windows™ impersonation also allows you to use Windows™ system monitoring tools to see which users are running which `unica_acsvr` processes on the server.

### What is the relationship between Unica Campaign users and Windows™ users?

To use Windows™ impersonation, you must establish a one-to-one relationship between Unica Campaign users and Windows™ users. That is, each Unica Campaign user must correspond to a Windows™ user with the exact same user name.

Typically, administration begins with a set of existing Windows™ users who will use Unica Campaign. You must create Unica Campaign users in Unica Platform with the exact same names as the associated Windows™ users.

### The Windows™ impersonation group

Each Windows™ user for whom you have set up a Unica Campaign user must be placed in a special Windows™ impersonation group. You must then assign the group to specific policies.

To ease administrative tasks, you can then grant `read/write/execute` privileges to the Unica Campaign partition directory for the group.

## Windows™ impersonation and logging into Unica

When Windows™ impersonation is set up, once users have logged into Windows™, Unica Campaign users are automatically logged into Unica using a single sign-on. When they open a browser and go to the Unica URL, they do not need to log in again, and they immediately see the Unica start page.

## Setting up Windows impersonation

Follow these instructions to set up Windows™ impersonation for Unica Campaign.

LDAP and Active Directory are required to run Windows™ impersonation. For details about setting up LDAP and Active Directory, see the *Unica Platform Administrator's Guide*.

To create a Windows™ impersonation group and assign it to policies, you must have administration privileges on the Windows™ server.

1. On the Configuration page, set the value of the **enableWindowsImpersonation** property in the `Campaign > unicaACLlistener` category to `TRUE`.



**Note:** There might be additional property requirements based on your Windows™ Domain Controller setup. For more information, see the single sign-on section of the *Unica Platform Administrator's Guide*.

2. Create Unica Campaign users:

You can use Unica Platform to create Unica Campaign internal or external users.

Create external users by configuring Active Directory users and group synchronization. Each user you create must have the same login name as the user's Windows™ user name.

3. Create the Windows™ impersonation group:

Create a Windows™ group specifically for Unica Campaign users. Then add the Windows™ users that correspond to Unica Campaign users to this group.

For more information about creating groups, see your Microsoft™ Windows™ documentation.

4. Assign the Windows™ impersonation group to policies:

After you create a Windows™ group to store users that correspond to Unica Campaign users, you must add the group to the following policies:

- Adjust memory quotas for a process
- Create Token object
- Replace a process level token

For more information about assigning groups to policies, see your Microsoft™ Windows™ documentation.

#### 5. Assign rights to the Windows™ impersonation group:

Using Windows™ Explorer, grant `read/write/execute` access to the **partitions/partition\_name** folder under your Unica Campaign installation to the Windows™ impersonation group.

For more information about assigning rights to folders, see your Microsoft™ Windows™ documentation.

## How Unica Campaign uses data filters to restrict access to customer data

Administrators can define data filters in Unica Platform to restrict Unica users from accessing specific customer data. In Unica Campaign, data filters affect flowchart output.

To restrict data access, Unica Platform administrators define data filters and then assign users or groups of users to different data filters. For example, administrators can control access to customer data based on the geographical sales territory to which the users are assigned.

To learn how to set up data filters, see the Unica Platform Administrator's Guide.

### Effect of data filters in Unica Campaign

Data filters apply to the Select, Extract, and Audience processes in Unica Campaign flowcharts. For example, if a database table includes 2000 records but 500 of them are restricted by a data filter, a Select All in Unica Campaign returns only 1500 records.

Data filters are applied to downstream processes when multiple processes of the same type are used in the same flowchart. For example, if the Select process is included at two different positions in the flowchart, one of which is downstream, data filters apply to both processes.

For downstream Select and Extract process boxes, when multiple tables are used as an input, the data from tables that are not filtered is added to the data from the filtered tables. This improves the performance of the processes.

`TEMP TABLE` can be `On` or `Off` for downstream data filtering to work.

Consider the following points about data filtering for all process boxes:

- Data filters work with only a single table in Select and Extract processes.
- When two different type of filters are used, the resultant data increases as the OR condition is used for the two filters.



- Data filters do not affect design-time activities. For example, a data filter does not hide the values that are displayed when a field is profiled. Although users can see restricted data when they profile a field or build queries in a process configuration dialog, the query results do not include restricted data. Data filters are integrated into the SQL that the process box uses to query the table with which the filter is associated.
- Data filters do *not* apply to raw SQL queries or to custom macros that use raw SQL. For example, if you use **Select customer IDs using SQL** to create a raw SQL query in the Select Process Configuration dialog, any data filters are ignored when you run the query. This behavior is intentional, to provide advanced users with the ability to perform SQL queries without restrictions.



**Important:** Raw SQL queries override data filters, so users who perform SQL queries can access records regardless of data filters. If you do not want Unica Campaign users to be able to use raw SQL, you must restrict their permissions.

## Example

In this example, we are using the table `DATAFILTER_TEST` and the corresponding `DATAFILTER_TEST.xml` XML file. You must make the required changes to the instructions based on the database and the tables that you are using.

Consider that data filtering is to be applied to the table `DATAFILTER_TEST`. The following prerequisites must be completed for the table:

1. Create the table `DATAFILTER_TEST` in the user database.

For Oracle:

```
CREATE TABLE DATAFILTER_TEST
(
  ID NUMBER,
  NAME VARCHAR2(20),
  COUNTRY VARCHAR2(20),
  AGE NUMBER,
  ACCT_TYPE VARCHAR2(20),
  RETAIL_ACCT VARCHAR2(10),
  HOUSEHOLD VARCHAR2(50)
);
```

For DB2:

```
CREATE TABLE SB6.DATAFILTER_TEST ( ID BIGINT, NAME VARCHAR(20), COUNTRY VARCHAR(20), AGE BIGINT,
ACCT_TYPE VARCHAR(20), RETAIL_ACCT VARCHAR(10), HOUSEHOLD VARCHAR(50));

CREATE TABLE SB6.DATAFILTER_TEST
(
  ID BIGINT,
  NAME VARCHAR(20),
  COUNTRY VARCHAR(20),
```

```

AGE BIGINT,
ACCT_TYPE VARCHAR(20),
RETAIL_ACCT VARCHAR(10),
HOUSEHOLD VARCHAR(50),
BIRTHDAY TIMESTAMP,
FIRSTOCCUPATION DATE
)

```

2. Add data to the table.
3. In the `DATAFILTER_TEST.xml` file, add the user table name `AUTODCC.DATAFILTER_TEST`. If you are using the SQL Server database, the table name must be specified as `dbo.DATAFILTER_TEST`.

Make the required changes to the XML file based on the user table. The XML file must contain data level filter table information, logical fields, data level filter, audience information and so on.

4. Add the XML file to the folder `<Platform_Home>/tools/bin`.
5. Create Audience levels in Unica Campaign application. For example, Customer, Account, Composite, Household.
6. Run `ManagerSchema_PurgeDataFiltering.sql` located at `<Platform_Home>/db` on the Platform database.



**Note:** This step removes all existing data level filters that are applied.

7. In the Unica Platform application, go to **Configurations > General > Data Filtering** and clear the **Default table name**. There **Default table name** must be blank. It **Default table name** is filled when the XML file is imported.
8. Go to `<Platform_Home>/tools/bin` and run following command.

```
./datafilteringScriptTool.sh -r <xmlfileName>
```

For example:

```

./datafilteringScriptTool.sh -r DATAFILTER_TEST.xml
bash-3.2# cd /opt/HCL/Campaign/SB8606/Platform/tools/bin/
bash-3.2# ./datafilteringScriptTool.sh -r DATAFILTER_TEST.xml
Script started
Script completed successfully
bash-3.2#

```

9. In the Platform application, go to **Settings > Configuration > General > Data filtering** and ensure that the **Default table name** is `AUTODCC.DATAFILTER_TEST`. This is the same name as the `<Schemaname.TableName>` in the XML file.
10. Set the default **Audience** to `Customer`.

11. Go to **Settings > Data Filters > Assign Users or Groups**. Select a filter criteria and assign the filter to a corresponding user.

For example, select **Country** as `India, USA` and the **User** as `Test`. The default `Test` user in Unica Campaign has permissions for table mappings and all permissions for all Campaign objects.

# Chapter 3. Database table administration

Unica Campaign administrators must perform initial one-time tasks and routine administration tasks that pertain to database tables.

There are two main types of database tables:

- System tables contain Unica Campaign application data.
- User tables contain customer data for use in marketing campaign flowcharts.

Most administration tasks pertain to user tables, because system table setup is typically handled by the installation process.

User tables must be mapped in Unica Campaign to make them available to flowcharts.

To efficiently manage mapped user tables, you can set up table catalogs.

Unica Campaign can also make use of customer data that is stored in flat files. Data dictionaries define the structure of user tables that are based on flat files.

## Table administration concepts

The following concepts pertain to system tables, user tables, flat files, and table mapping.

### What are system tables?

System tables are database tables that contain Unica Campaign application data.

System tables store metadata about campaign objects, including campaigns, sessions, flowcharts, offers, templates, custom macros, stored derived fields, and triggers. Contact and response history information is also stored in system tables.

The Unica Campaign installation and configuration process includes setting up the Unica Campaign system tables. For more information, see the installation documentation.

### What are user tables?

User tables are tables that contain data that you use in processes within a Unica Campaign flowchart. You can map user tables to tables in a relational database or to ASCII flat files.



**Note:** Before mapping a user table in Unica Campaign, ensure that the table uses only data types that Unica Campaign supports. For a list of supported data types for each database, see [Supported data types for user tables \(on page 52\)](#).

Typically, a user table contains data about your company's customers, prospects, or products. For example, a user table might contain columns for customer account data such as Account ID, Account Type, and Balance. You might want to use this data in a campaign that targets customers with certain account types and balances.

There are three types of user tables: Base tables, Dimension tables, and General tables.

## What is a base record table?

A base record table is a table that contains data about potential contacts of campaigns, such as individual customers, businesses, accounts, or households.

Each base record table can be mapped to either a database table or an ASCII flat file (fixed-width or delimited) and must contain an identifier for that contact; that is, there must be one or more columns, which together store a unique identifier for the audience entity. These columns cannot be null for any record in the table.

You map the identifier in the base record table to one or more audience levels.

Processes in a flowchart select these audience level identifiers from base record tables when campaigns run.

## What is a dimension table?

A dimension table is a database table that augments the data in a base record table mapped to a database table.



**Note:** Dimension tables cannot be mapped to flat files and they cannot be joined to base tables mapped to flat files. Dimension tables and their corresponding base tables must be mapped to database tables in the same physical database (that is, the same data source).

For example, a dimension table might contain demographic information based on zip code, accounts belonging to a customer, or the customer's transactions, product information, or purchase transaction details.

When you define a dimension table, you specify the key fields for joining the dimension table to the base record table.

## What is a general table?

A general table is a free-format table to which you can export data from Unica Campaign. It is the simplest table type that can be created and is used only to export data from Unica Campaign for use by other applications (general tables cannot be subsequently accessed in Unica Campaign, unless you map them as base tables).

You can define a general table in a relational database, as a delimited flat file, or as a flat file with a data dictionary. General tables do not have keys or audience levels.

You use general tables in the **Snapshot** process to capture campaign data for use by other applications. For example, you can define the **Snapshot** process to store historical data or mailing lists in the exported general table.

You use general tables only to export data; you cannot query or manipulate data in general tables in Unica Campaign.

## About table mapping

Mapping tables is the process of making external customer or system tables accessible in Unica Campaign.

A table mapping is the metadata that is used to define a base, dimension, or general table. It contains information on the data source, the table name and location, the table fields, audience levels, and data. Table mappings can be stored for re-use in table catalogs.

The following capabilities are provided for table mappings:

- By default, all the table mappings defined in the parent interactive channel will be inherited. (Including dimension tables)
- Users cannot edit, un-map, or add a new dimension table to inherited table mapping in the child interactive channel.
- Users can either reuse existing table mapping or can define a new table mapping in the child interactive channel.
- The inherited table mapping from the parent interactive channel will be serialized and deployed to run-time if not defined in the current interactive channel.
- In the case of reloading to design time, the table mappings will not be considered if inherited from parent interactive channels.

## Using flat files as data sources

Typically, most of the marketing data that you will access from a flowchart will reside in a database, but it may be convenient to access data directly from a flat file. Unica Campaign supports the ability to work with data stored in delimited ASCII flat files or fixed-width ASCII flat files with a data dictionary.

You can map flat files as base tables and access them from within a flowchart. You cannot map flat files as dimension tables.

Direct access to flat files removes the need to first upload data into a database to use with Unica Campaign. This is useful for data exported from third-party applications (such as Excel or SAS) and for transient, one-time use data (such as a campaign-specific seed list, last-minute suppressions, predictive model scores, and other uses).

For additional information, read about data dictionaries.

## Initial table administration tasks

Administrators must perform several initial table-related tasks after Unica Campaign is installed.

The Unica Campaign installation must be complete, including:

- Setup and configuration of the Unica Campaign system database.
- Configuration of Unica Campaign to access any databases containing user tables (data sources are defined).

For instructions, see the *Unica Campaign Installation Guide*.

1. Test system table access. See [Testing system table access \(on page 47\)](#).
2. Test user table access. See [Testing user table access \(on page 47\)](#).

3. Define any additional audience levels that you require. (Unica Campaign provides the `Customer` audience level but you might want to define others, such as `Household`.) See [Audience level administration \(on page 133\)](#).
4. Map the audience level system tables for all audiences that you want to report on, such as `Customer`. See [Mapping Customer audience level system tables \(on page 78\)](#).
5. Map user tables. See [Mapping and unmapping user tables \(on page 67\)](#).

## Testing system table access

After Unica Campaign is installed, administrators must confirm that the Unica Campaign system tables are mapped and database connections are working properly.

1. Select **Settings > Campaign settings**.
2. Click **Manage table mappings**.

The **Table Mappings** dialog opens, with **Show system tables** selected.

The Unica Campaign system tables are mapped automatically when you set up the Unica Campaign database, if you used `UA_SYSTEM_TABLES` as the ODBC name. For more information, see the installation documentation.

Each Unica Campaign system table entry should have a populated database table name in the right column. However, if your implementation is not using certain features, some system tables might remain unmapped.

If the system tables are not mapped, contact the person who performed the Unica Campaign installation and configuration.

## Testing user table access

After Unica Campaign is installed, administrators must confirm that Unica Campaign is configured properly to access the necessary user tables. You can also test user table access whenever you configure a new data source.

Follow the steps below to view the customer databases that Unica Campaign is set up to access.



**Note:** As an alternative to the steps below, you can open a flowchart for editing, click **Settings > Campaign settings** and select **View datasource access**. The **Database sources** dialog opens. This dialog lists the system table database and all configured customer databases. From this dialog, you can log in and out of customer databases.

1. Select **Settings > Campaign settings**.
2. Click **Manage table mappings**.
3. In the Table Mappings dialog, click **Show user tables**.  
Initially, there are no mapped user tables and the list is empty.
4. Click **New table**.  
The New Table Definition dialog opens.
5. Click **Next**.

You are prompted to map to a file or database.

6. **Map to existing table in selected database** is selected by default. You should see one or more databases in the **Select data source** list. If no entries appear in the Select Data Source box after initial installation, you must create and configure data sources in Unica Campaign.
7. If you are using flat files for user data, select **Map to existing file**, then click **Next**. The New Table Definition window now contains fields for the flat file and data dictionary location. Click **Browse** to locate the necessary files, or enter the relative path and file name. Files must be located under the partition root of Unica Campaign to be accessible.

If you have not already done so, you can now map the user tables into Unica Campaign.

## User table administration

Unica Campaign administrators typically perform the following tasks pertaining to user tables.

### Adding new user data sources to Unica Campaign

Organizations can use their own databases and flat files as data sources in Unica Campaign flowcharts. For example, an organization might store customer names and addresses in DB2® and customer purchase history in SQL Server. Other organizations might have a distributed big data system, along with some flat files.

#### About this task

The person who performs the Unica Campaign installation typically prepares the existing data sources for use with Unica Campaign. Over time, other data sources may become available. This topic explains how to make these additional user data sources available to Unica Campaign. This topic does not pertain to system tables.

Perform the tasks below to make new user data sources available in Unica Campaign. When these tasks are complete, Unica Campaign users can build flowcharts that access data from all configured data sources, including big data such as Apache Hadoop Hive or Amazon Redshift.



**Important:** This topic provides general information. It does not explain how to prepare individual data sources. For detailed instructions for individual database types, including big data, see the *Unica Campaign Installation Guide*.

#### A. Create an ODBC or native connection to each database that will provide user data to Unica Campaign.

The Unica Campaign listener server must have an ODBC or native connection to each database or schema that contains customer user tables.

For instructions, see the Unica Campaign Installation Guide.



For details about supported user database sources, see the Unica Recommended Software Environments and Minimum System Requirements.

## B. Import the provided XML template into Unica Campaign.

Unica Campaign provides XML templates to use as the basis for each data source that you create in Unica Campaign.

During installation, at least one XML template was imported (SQL, DB2®, or Oracle) for use by the Unica Campaign system tables. If the user database that you plan to use is of the same type, you do not need to import the XML template again. However, if the template that you need does not exist in Unica Campaign, you must import it. You only have to do this once for each template type that you will use. For example, if you have several Teradata databases that will provide information to Unica Campaign, import the Teradata XML template once. Then, in a later step, you will create two data sources based on that template.

To import a template into Unica Campaign, use the `configTool` utility:

- `configTool` is in `<Marketing_Platform_Home>/tools/bin`. For more information, *Unica Platform Administrator's Guide*.
- XML templates are in `<Campaign_Home>/conf`. Each template file is named in such a way that you can recognize its purpose (OracleTemplate.xml, SQLServerTemplate.xml, NetezzaTemplate.xml, and so on).

The following example imports `OracleTemplate.xml` into the default Unica Campaign partition, `partition1`.



**Note:** `configTool` has an extension of `.bat` for Windows™ and `.sh` for Unix.

```
./configTool -i -p "Affinium|Campaign|partitions|partition1|dataSources" -f <Campaign_Home>/conf/
OracleTemplate.xml
```

## C. Create and configure a data source in Unica Campaign based on the XML template that you imported.

In this task, you create a new data source based on an XML template, then configure the new data source by filling in the template

Perform this task once for each data source that will provide information to Unica Campaign. For example, if you have two Teradata databases (one that contains contact information and another one that contains purchase history), create two separate datasources based on the Teradata XML template.

1. In Unica Campaign, choose **Settings > Configuration**.
2. Go to Campaign | partitions | partition[n] | dataSources.
3. Select the desired template.
4. In the **New** field, provide a category name that will help you remember which data source this is, such as **TD\_Customers** or **DB2\_Customers**.
5. Adjust the field values to set the configuration properties for the new user data source.

The template provides appropriate default values for most of the properties. However, some of the properties do not have default values, such as: `ASMUserForDBCredentials`, `DSN`, `SystemTableSchema`, `OwnerForTableDisplay`, `CustomMacroSchema`. You must provide values for your database instance. Additionally, you might need to adjust other properties for your specific database. For details, see [Campaign | partitions | partition\[n\] | dataSources \(on page 379\)](#).

6. Save your changes.
7. Restart the Unica Campaign listener to make the changes take effect.



**Note:** If you need to adjust the properties later, choose **Settings > Configuration**. Each data source that you added in Unica Campaign is listed as `<data-source-name>` under `Campaign | partitions | partition [n] | dataSources`.

## D. Perform table mapping in Unica Campaign for the new data source.

You must map user tables to make the data accessible in Unica Campaign flowcharts.

Prerequisites:

- Define audience levels before you map user tables. See [Audience level administration \(on page 133\)](#).
- Verify that the user table contains only the data types that Unica Campaign supports. See [Supported data types for user tables \(on page 52\)](#).

To map user tables:

1. Choose **Settings > Campaign settings > Manage table mappings**. (Or open a flowchart for editing and choose **Admin > Tables**.)
2. In the Table Mappings dialog, click **Show user tables**.
3. Click **New table**. The **New table definition** dialog opens.
4. Click **Next**.
5. Select **Map to existing table in selected database**.
6. Select the datasource that you created previously, then click **Next**.
7. You are prompted to log in to the database.
8. Follow the prompts to map the table. For complete instructions, see [Mapping and unmapping user tables \(on page 67\)](#).

Optionally, store the table mapping in a catalog (.cat) so it is available to all flowcharts. By using a catalog, you avoid mapping the same table again and again across flowcharts. Users can open a flowchart for editing, select **Stored table catalogs** from the **Options** menu, and load a stored catalog. They can use any of the user tables in that catalog as flowchart input (for example, as input to a Select process).

## Creating new user tables through an output process

You can create new user tables by exporting data from an output process, such as Snapshot, Call List, or Mail List.

1. While editing a flowchart, open the output process from which you want to create the new user table.
2. In the **Export to** list, select **New mapped table**.  
The New Table Definition window opens.
3. Select **Base record table**, **Dimension table**, or **General table**. Typically, you would export the data to a new base record table in an existing flat file or database. If you need to read the exported data back into Unica Campaign, you must export it as a base record table.
4. Click **Next**.
5. Select **Create new file** or **Create new table in selected database**.
6. If you selected **Create new table in selected database**:
  - a. Select the database in which to create the table, then click **Next**.
  - b. Select the **Source table fields** to export. You can select Unica Campaign Generated Fields, the audience-level identifier, and fields from the input cell. Use the **Add**, **Remove**, **Up**, and **Down** buttons to specify and order fields in the **New table fields** list.
  - c. Click **Next**.
  - d. Specify a **Database table name** and **Unica Campaign Table name** for the new table.
  - e. Optional: Select the new table fields and modify the **Unica Campaign Field name**.
  - f. Click **Next**.
  - g. Select the **Audience level** for the new table and specify the audience level field in the new table, then click **Next**.
  - h. Optional: Use **Add** to select additional audience levels for the new table, then click **Next**.
  - i. Define profiling for the new table. Profiling enables users to see and select table values when editing a flowchart or building a query. See [Configuring profiling when mapping user tables \(on page 75\)](#).
  - j. Click **Finish**.
7. If you selected **Create new file**:
  - a. Click **Next**.
  - b. Select **Fixed-Width flat file** or **Delimited file**, specify the **Settings** fields appropriately, and click **Next**.
  - c. Select the **Source table fields** to export to the new table or file. You can select Unica Campaign Generated Fields, the audience-level identifier, and fields from the input cell. Use the **Add**, **Remove**, **Up**, and **Down** buttons to specify and order fields in the **New table fields** list.
  - d. Click **Next**.
  - e. Select the **Audience level** for the new table, specify the audience level field in the new table, then click **Next**.
  - f. Optional: Click **Add** to select additional audience levels for the new table, then click **Next**.
  - g. Define profiling for the new table. Profiling enables users to see and select table values when editing a flowchart or building a query. See [Configuring profiling when mapping user tables \(on page 75\)](#).
  - h. Click **Finish**.

## Accessing user data sources from within a flowchart

To access user data from within a flowchart, ensure that you are logged in to each database that is acting as a data source.

1. Open a flowchart for editing, then click the **Admin** menu  and select **Database Sources**.

The **Database Sources** window shows all user databases that Unica Campaign is configured to access, plus the database that contains system tables.

2. To log in to a database, select it and click **Login**.
3. Click **Close**.

You now have access to tables in that database. To query a table in that database, the table must be mapped.

## Supported data types for user tables

Before you map user tables in Unica Campaign, ensure that your tables use only the supported data types for each supported database. Data types that are not listed here are not supported.



**Note:** A column of datatype DATE, DATETIME, or TIMESTAMP of a table, when mapped in an Unica Campaign flowchart, appears as type TEXT with the format of the DATE, DATETIME, or TIMESTAMP stored in brackets, such as [DELIM\_D\_M\_Y] or [DT\_DELIM\_D\_M\_Y]. Although its data type in table mapping in the flowchart shows as TEXT, the application knows the format and handles it accordingly. Do not map columns of these three data types or any of the date- or time-related data types as an Audience ID column to a TEXT audience level. Mapping the date-related column as TEXT audience level is not supported.

### Supported datatype with Amazon Redshift

BIGINT  
CHAR  
DATE  
DECIMAL  
DOUBLE PRECISION  
INTEGER  
REAL  
SMALLINT  
VARCHAR

### Supported datatype with Apache Hadoop Hive

BIGINT  
CHAR  
DATE  
DECIMAL  
DOUBLE  
FLOAT  
INT  
SMALLINT  
STRING

TIMESTAMP  
TINYINT  
VARCHAR

### **Supported datatype in Unica Campaign with GBQ database**

BOOL  
DATE  
TIMESTAMP  
FLOAT64  
INT64  
NUMERIC  
STRING  
INT64  
INT  
SMALLINT  
INTEGER  
TINYINT  
NUMERIC  
DECIMAL

### **Supported datatype with DB2®**

bigint  
char  
date  
decimal  
double  
float  
int  
numeric  
real  
smallint  
timestamp  
varchar

\*includes dashDB™ and DB2® BLU.

### **Supported datatype with HP Vertica**

All basic (standard) data types.

### **Supported datatype with Netezza®**

bigint  
byteint

char(n) [1]  
date  
float(p)  
int  
nchar(n) [2]  
numeric(p, s)  
nvarchar(n) [2]  
smallint  
timestamp  
varchar(n) [1]

[1] Not supported when used in the same table with `nchar` or `nvarchar`.

[2] Not supported when used in the same table with `char` or `varchar`.

### Supported datatype with Oracle

DATE  
FLOAT (p)  
NUMBER [(p, s)] [1]  
TIMESTAMP  
VARCHAR2(size BYTE)

[1] Precision is required for NUMBER unless you set the data source property **Campaign > partitions > partition[n] > dataSources > [dataSourceName] > UseSQLToRetrieveSchema** to TRUE. If you do not specify a precision and you do not set **UseSQLToRetrieveSchema** to TRUE, Unica Campaign assumes it can store the values in a data type that preserves 15 digits of precision. This can be problematic for fields that hold values whose precision exceeds 15 digits because precision will be lost when the value is brought into Unica Campaign.

### Supported datatype with SQL Server

bigint  
bit  
char(n) [1]  
datetime  
decimal  
float  
int  
nchar [2]  
numeric  
nvarchar(n) [2, 3]  
real  
smallint  
text

tinyint

varchar(n) [1]

[1] Not supported when used in the same table with `nchar` or `nvarchar`.

[2] Not supported when used in the same table with `char` or `varchar`.

[3] `nvarchar(n)` is supported but `nvarchar(max)` is not.

### **Supported datatype with Teradata**

bigint

byteint

char

date

decimal

float

int

numeric

smallint

timestamp

varchar

### **Supported datatype with MariaDB**

varchar

tinyint

text

date

smallint

mediumint

int

bigint

float

double

decimal

datetime

timestamp

time

year

char

mediumtext

## Supported datatype with Postgres

INT  
SMALLINT  
BIGINT  
BIGSERIAL  
DOUBLE PRECISION  
FLOAT  
CHAR  
VARCHAR  
TEXT  
BOOLEAN  
TIMESTAMP  
DATE

## Using Amazon Redshift user data sources with Unica Campaign

Unica Campaign supports Amazon Redshift as a user data source. Redshift is not supported for system tables.

The integration is achieved by using the PostgreSQL ODBC driver for ODBC connectivity and SQL and configuring the **PostgreSQL** template in Unica Campaign for each datasource.

- For initial configuration instructions, see the *Unica Campaign Installation Guide*.
- Details about configuration settings are in the Unica Campaign Administrator's Guide.
- For information about supported versions, see the *Recommended Software Environments and Minimum System Requirements*.
- For information about Amazon Redshift, see <http://aws.amazon.com/redshift/>.

## Using MariaDB datasources with Unica Campaign

Unica Campaign supports MariaDB as System and user tables. The MariaDB database connection with Unica Campaign is supported with unixODBC and MariaDB ODBC Connector. Users must install unixODBC and MariaDB ODBC Connector on Unica Campaign Analytical server (listener) machine.



**Note:** For information on supported versions, see the Recommended Software Environments and Minimum System Requirements document.

## MariaDB Server On Linux

On Linux, add the following lines to `/etc/my.cnf`

```
[mysqld]
innodb_page_size=32768
character_set_server=utf8
sql_mode='NO_BACKSLASH_ESCAPES'
```



```
innodb_strict_mode=0
wait_timeout=2592000
```



**Note:** V12.1 FP2 onwards we need to add `table_definition_cache=1024` MariaDB database level parameter in configuration file.

## MariaDB server on Windows

Configure `my.ini` by adding the following lines to section `mysqld`.

```
[mysqld]
innodb_page_size=32768
character_set_server=utf8
sql_mode='NO_BACKSLASH_ESCAPES'
innodb_strict_mode=0
wait_timeout=2592000
```



**Note:** V12.1 FP2 onwards we need to add `table_definition_cache=1024` MariaDB database level parameter in configuration file.

## Using Postgres datasources with Unica Campaign

Unica Campaign supports Postgres as System and user tables. The Postgres database connection with Unica Campaign is supported with unixODBC and Postgres ODBC Connector. Users must install unixODBC and Postgres ODBC Connector on Unica Campaign Analytical server (listener) machine.



**Note:** For information on supported versions, see the Recommended Software Environments and Minimum System Requirements document.

### Postgres Server On Linux

To create database in Postgres DB use below command:

```
CREATE DATABASE <your_db_name>;
```

```
CREATE USER <your_user_name> WITH PASSWORD <your_user_password>;
```

Set the following parameters in Postgres configuration (`postgresql.conf`) file for Platform and Campaign.

- max\_connections: set this value to 1000



**Note:** While running application if there is error **Campaign Server: loadFolderDetailItemsByEntity: Database connection error (Error Code = 11518)**. Please increase the value of parameter max\_connections as per application usage.

### Postgres server on Windows:

To create database in Postgres DB use below command:

```
CREATE DATABASE <your_db_name>;
```

```
CREATE USER <your_user_name> WITH PASSWORD <your_user_password>;
```

Set the following parameters in Postgres configuration (**postgresql.conf**) file for Platform and Campaign.

- max\_connections: set this value to 1000



**Note:** While running application if there is error **Campaign Server: loadFolderDetailItemsByEntity: Database connection error (Error Code = 11518)**. Please increase the value of parameter max\_connections as per application usage.

## Google BigQuery ODBC.ini configuration

Navigate to /opt/simba/googlebigqueryodbc/Setup and Configure odbc.ini

Make sure to add DefaultDataset as mentioned below:

```
[GBQ]
```

```
_ # Description: DSN Description._
```

```
# This key is not necessary and is only to give a description of the data source.
```

```
Description=Simba ODBC Driver for Google BigQuery (64-bit) DSN
```

```
—
```

```
# Driver: The location where the ODBC driver is installed to.
```

```
Driver=/opt/simba/googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
```

```
—
```

```
# These values can be set here, or on the connection string.
```

```
# Catalog: The catalog to connect to. This is a required setting.
```

Catalog=<Project\_Name>

DefaultDataset=<Dataset\_TOBE\_Used>

—

# SQLDialect: The SQL Dialect to use. There are two SQL dialects:

# 0 = BigQuery Legacy SQL

# 1 = BigQuery Standard SQL (SQL 11)

SQLDialect=1

—

# OAuth Mechanism: The OAuth mechanism to use. There are two choices:

# 0 = Service Authentication

# 1 = User Authentication

# This is a required setting.

OAuthMechanism=0

—

# RefreshToken: The Refresh Token used. This can be generated from the Windows connection dialog.

# It can also be generated by executing the following steps:

1. Get an Authentication by logging into Google from the following URL

[https://accounts.google.com/o/oauth2/auth?scope=https://www.googleapis.com/auth/](https://accounts.google.com/o/oauth2/auth?scope=https://www.googleapis.com/auth/bigquery&response_type=code&redirect_uri=urn:ietf:wg:oauth:2.0:oob&client_id=977385342095.apps.googleusercontent.com&h)

[bigquery&response\\_type=code&redirect\\_uri=urn:ietf:wg:oauth:2.0:oob&client\\_id=977385342095.apps.googleusercontent.com&h](https://accounts.google.com/o/oauth2/auth?scope=https://www.googleapis.com/auth/bigquery&response_type=code&redirect_uri=urn:ietf:wg:oauth:2.0:oob&client_id=977385342095.apps.googleusercontent.com&h)

2. Run the get\_refresh\_token.sh shell script and pass in the Authentication Token received in step 1
3. Copy the Refresh Token (the text on the right-side of the colon, without the trailing or leading spaces) from the output of the script.

This is a required setting.

RefreshToken=<REFRESH\_TOKEN\_GENERATED>

—

Email: For Service Authentication, this is a required setting. It is your GENERATED service account email (not a typical Gmail account).

It is unique and associated with at least one public/private key pair.

Email=<EMAIL\_FROM\_GBQ\_Project>

—

KeyFile Path: For Service Authentication, this is a required setting. This is the path to the stored keyfile (.p12).

KeyFilePath=<.JSON or .P12 PATH Downloaded from your GBQ project>

—

Used to specify the full path of the PEM formatted file containing trusted SSL CA certificates.

If an empty string is passed in for the configuration, the driver expects the trusted SSL CA

Certificates can be found in the file named cacerts.pem located in the same directory as the driver's shared library.

TrustedCerts=/opt/simba/googlebigqueryodbc/lib/64/cacerts.pem

—

AllowLargeResults: When set to 1, the driver allows for result sets in responses to be larger than 128 MB.

AllowLargeResults=0

—

LargeResultsDataSetId: DataSetId to store temporary tables created. This is a required setting if AllowLargeResults is set to 1.

LargeResultsDataSetId=\_bqodbc\_temp\_tables

—

LargeResultsTempTableExpirationTime: Time in milliseconds before the temporary tables created expire. This is a required setting if AllowLargeResults is set to 1.

LargeResultsTempTableExpirationTime=3600000

### **Configure odbcinst.ini**

[Simba ODBC Driver for Google BigQuery 64-bit]

Description=Simba ODBC Driver for Google BigQuery(64-bit)

Driver=/opt/simba/googlebigqueryodbc/lib/64/libgooglebigqueryodbc\_sb64.so

- Copy "SimbaODBCDriverforGoogleBigQuery.lic" from the mail you would have received while downloading simba driver and copy it to "/opt/simba/googlebigqueryodbc/lib/64"

## Update Setenv.sh

```
LD_LIBRARY_PATH=<CAMPAIGN_HOME>/bin:<SYSTEM_DB_LIB_PATH>/:/opt/simba/googlebigqueryodbc/lib/64:/usr/lib64:/lib64
export LD_LIBRARY_PATH
ODBCINI=/opt/simba/googlebigqueryodbc/Setup/odbc.ini
Export ODBCINI
```

## Configure Google BigQuery datasource

Campaign has provided new template **BigQueryTemplate.xml** to use to connect Google Bigquery user datasource.

1. Import BigQueryTemplate.xml from <CAMPAIGN\_HOME>/conf using configtool.
2. Create data source using BigQueryTemplate.xml and specify below properties:

ASMUserForDBCredentials	<Platform_User>
DeleteAsRecreate	TRUE
DeleteAsTruncate	FALSE
BulkInsertRequiresColumnType	TRUE

Assign Data source created in step 2 to platform user specified in ASMuserDBCredentials and provide service account id as user and path of .json file as password.

## Google BigQuery - Not supported features with Campaign

1. Campaign will not be able to support any new functions (Like Struct, UNNEST, ARRAY Data type, ARRAY\_Length(), etc) available in Big Query in any form be it SQL Advance query, Derived field , Pre- Post SQL , RAW SQL or custom macros.
2. Campaign can support View and user can take input of view in Campaign and do segmentation, sampling, etc but please make a note - View created based on any new functions like Struct, UNNEST, ARRAY\_LENGTH(), etc will not be supported and will not work as expected.
3. Arithmetic functions such as - (minus) , + (Plus) ,etc are not supported for Date or date related functions if compared with Date data type column by GBQ itself , hence it will not be supported via Campaign as well , GBQ support various date functions like DATE\_ADD , DATE\_SUB , DATE\_DIFF , etc all this will be supported in Campaign as RAW SQL query.

For Example - In campaign below query will not work.

Current\_DATE() + 30 instead user can use below :

DATE\_ADD(CURRENT\_DATE(), INTERVAL 30 day)

OR

As Point and Click method as well :

```
DATE_STRING(CURRENT_JULIAN()+30 ,YYYYMMDD)
```

## Using Hive-based Hadoop big data sources with Unica Campaign

Unica Campaign supports the ability to use Hive™ based implementations of Apache Hadoop® as a user data source.



**Note:** To learn how to prepare Apache Hadoop Hive user data sources for Unica Campaign, see the *Unica Campaign Installation Guide*.

If your system is configured to support a Hive-based Hadoop big data system, you have the following capabilities:

- **Bring data into Unica Campaign:** Use your Hive-based Hadoop big data system as a user data source for Unica Campaign. For example, create a marketing campaign flowchart that uses customer account data from your big data instance to target customers with specific account types and balances. For initial configuration instructions, see the *Unica Campaign Installation Guide*.
- **Export data from Unica Campaign:** Send content from Unica Campaign to your Hive-based Hadoop big data system. You can create a marketing campaign flowchart that pulls user data from other data sources, such as DB2® or Oracle databases. Use the Unica Campaign flowchart to create a specific market segment, then use the Snapshot process in the flowchart to export that segment back to your big data instance. Configuring data export to Hive is explained in the *Unica Campaign Administrator's Guide*.
- The ability to **create temp tables for in-database optimization** is supported. Using the Unica Campaign in-database optimization feature can improve flowchart performance. When in-database optimization is on, processing is done on the database server and output is stored in temporary tables on the database server whenever possible. For more information, read about **useInDbOptimization**.

## Requirements and restrictions for Hive-based Hadoop data sources

The following requirements and restrictions apply to using Hive-based Hadoop data sources with Unica Campaign.

- One of the following drivers is required. The customer is responsible for obtaining the driver.
  - DataDirect Apache Hive ODBC driver from Progress.com: DataDirect Connect64(R) for ODBC Release 7.1.5.
  - Cloudera ODBC Driver for Apache Hive Version 2.5.16 for Apache Hadoop Hive from Cloudera, Inc.
  - Hortonworks 64-bit ODBC driver for HDP 2.3 (v2.0.5) available at <http://hortonworks.com/hdp/addons/>.
- The following Hadoop distributions are supported, with Apache Hive as the connection point: Cloudera, Hortonworks, IBM® BigInsights® ™, MapR.
- Minimum supported Hive version: 0.14
- The big data integration is currently supported on Linux™ RHEL 6.3 or higher.

- Hive-based Hadoop is supported as a user data source only. It is not supported for Unica Campaign system tables.
- The big data integration does not currently support the Unica Campaign Cube, Optimize, or Interact List process boxes or Deliver Landing Pages in an Extract process box.



**Note:** It is mandatory for Hive database users to configure the Loader.

## Exporting data from Unica Campaign to a Hive-based Hadoop system

You can send data from to your Hive-based Hadoop big data system.

To send data from Unica Campaign to your Hive-based Hadoop big data system, create a flowchart that pulls user data from one or more data sources, such as DB2® and Oracle databases. Configure the Snapshot process in a flowchart to export the data to your big data instance. When you run the flowchart, the snapshot data is exported to the Hive database.

The Unica Campaign configuration settings for the Hive dataSource determine how the data is transferred from Unica Campaign to Hive.

1. An administrator must configure the Hive data source (in Campaign | Partitions | Partition[n] | dataSources) to specify the required SCP and SSH commands:
  - The **LoaderPreLoadDataFileCopyCmd** value uses SCP to copy data from Unica Campaign to a temp folder called `/tmp` on your Hive-based Hadoop system. The location must be called `/tmp` and it must be on the Hive server (the file system location, not the HDFS location). This value can either specify the SCP command or call a script that specifies the SCP command. See the two examples below.
  - The **LoaderPostLoadDataFileRemoveCmd** value must specify the SSH "rm" command to remove the temporary file after it is loaded into Hive.

To support this functionality, SSH must be configured on the Unica Campaign listener server. For instructions, see the *Unica Campaign Installation Guide*.

2. Configure the Snapshot process in a flowchart to obtain input data from one or more data sources and export the data to your Hive database. Design the flowchart as you normally would, including any desired processes such as Select and Merge.
3. Run the flowchart.

The entire dataset is exported to a temporary data file at `<Campaign_Home>/partitions/partition[n]/tmp`. The temporary file is copied to the Hive server using **LoaderPreLoadDataFileCopyCmd** and the data is loaded into a Hive table. The temporary file is removed from the Hive server using **LoaderPostLoadDataFileRemoveCmd**.

**Example 1: Configuring export to MapR:** This example shows Unica Campaign configured for export to MapR, using a datasource called Hive\_MapR. **LoaderPreLoadDataFileCopyCmd** uses SCP to copy the data file from the local machine running Unica Campaign to a temp directory on the remote machine running the Hive server (the MapR machine). **LoaderPostLoadDataFileRemoveCmd** uses SSH rm to remove the file.

```
Campaign | Partitions | Partition[n] | dataSources | Hive_MapR | LoaderPreLoadDataFileCopyCmd = scp <DATAFILE>
mapr@example.company.com/tmp
```

```
Campaign | Partitions | Partition[n] | dataSources | Hive_MapR | LoaderPostLoadDataFileRemoveCmd = ssh
mapr@example.company.com "rm/tmp/<DATAFILE>"
```

**Example 2: Configuring export to Cloudera using a script:** Using a script can be useful to avoid file permission issues. If there are any issues related to file permissions, the LOAD command cannot access the data file and the command fails. To avoid this type of issue, you can write your own shell or command-line script to SCP the data file to Hive and update the file permissions of the data file. The following example shows Unica Campaign configured to use a script for export to Cloudera. **LoaderPreLoadDataFileCopyCmd** calls a script that uses SCP to copy the data file from the local machine running Unica Campaign to a temp directory on the remote Cloudera machine. **LoaderPostLoadDataFileRemoveCmd** removes the file.

```
Campaign | Partitions | Partition[n] | dataSources | Hive_Cloudera | LoaderPreLoadDataFileCopyCmd = /opt/HCL/
CampaignBD/Campaign/bin/copyToHadoop.sh <DATAFILE>
```

```
Campaign | Partitions | Partition[n] | dataSources | Hive_Cloudera | LoaderPostLoadDataFileRemoveCmd = ssh
cloudera@example.company.com "rm /tmp/<DATAFILE>"
```

Here is the script that is called by **LoaderPreLoadDataFileCopyCmd**:

```
copyToHadoop.sh:
#!/bin/sh
scp $1 cloudera@example.company.com:/tmp
ssh cloudera@example.company.com "chmod 0666 /tmp/'basename $1'"
```

The script is on the Unica Campaign listener machine. The script executes the SCP command as the user "cloudera" on the destination server (example.company.com) to copy the file to the tmp directory. The SSH command connects as the same user to make sure that the permissions are correct for the load and removal processes that will follow.



**Note:** It is mandatory for Hive database users to configure the Loader.

## Hive query language conformance

The following guidelines apply when Unica Campaign is integrated with Hive-based big data sources.

Apache Hive has its own query language called HiveQL (or HQL). While based on SQL, HiveQL does not strictly follow the full SQL-92 standard. HiveQL offers extensions not in SQL, including multitable inserts and create table as select, but only offers basic support for indexes. Also, HiveQL lacks support for transactions and materialized views, and only limited subquery support.

Therefore, the following guidelines apply when using Hive-based big data sources with Unica Campaign:

- The SQL must conform with HiveQL.
- If you are writing raw SQL queries for use in Unica Campaign, confirm that the queries work on Hive.



- Using multiple SQL statements for raw SQL queries is not supported.
- You might need to modify existing queries for Hive if you are using raw SQL in Unica Campaign process boxes, custom macros, or derived fields, for pre- and post-processing.

## Using Cloudera Impala-based Hadoop big data sources with Unica Campaign

Unica Campaign supports the ability to use Cloudera Impala™ based implementations of Hadoop® as a user data source.



**Note:** To learn how to prepare Hadoop Impala user data sources for Unica Campaign, see the *Unica Campaign Installation Guide*.

If your system is configured to support a Impala-based Hadoop big data system, you have the following capabilities:

- **Bring data into Unica Campaign:** Use your Impala-based Hadoop big data system as a user data source for Unica Campaign. For example, create a marketing campaign flowchart that uses customer account data from your big data instance to target customers with specific account types and balances. For initial configuration instructions, see the *Unica Campaign Installation Guide*.
- **Export data from Unica Campaign:** Send content from Unica Campaign to your Impala-based Hadoop big data system. You can create a marketing campaign flowchart that pulls user data from other data sources, such as DB2® or Oracle databases. Use the Unica Campaign flowchart to create a specific market segment, then use the Snapshot process in the flowchart to export that segment back to your big data instance. Configuring data export to Impala is explained in the *Unica Campaign Administrator's Guide*.
- The ability to **create temp tables for in-database optimization** is supported. Using the Unica Campaign in-database optimization feature can improve flowchart performance. When in-database optimization is on, processing is done on the database server and output is stored in temporary tables on the database server whenever possible. For more information, read about **useInDbOptimization**.

## Requirements and restrictions for Impala-based Hadoop data sources

The following requirements and restrictions apply to using Impala-based Hadoop data sources with Unica Campaign.

- The following drivers is required. The customer is responsible for obtaining the driver.
- Cloudera ODBC Driver for Impala Version 2.5.41.
- Minimum supported Impala version: 2.9.0.
- Impala based Hadoop user data source integration is supported with Cloudera.
- The big data integration is currently supported on Linux™ RHEL 6.3 or higher.
- Impala-based Hadoop is supported as a user data source only. It is not supported for Unica Campaign system tables.
- The big data integration does not currently support the Unica Campaign Cube, Optimize, or Interact List process boxes or Unica Deliver Landing Pages in an Extract process box.

## Exporting data from Unica Campaign to a Impala-based Hadoop system

You can send data from Unica Campaign to your Impala-based Hadoop big data system.

To send data from Unica Campaign to your Impala-based Hadoop big data system, create a flowchart that pulls user data from one or more data sources, such as DB2® and Oracle databases. Configure the Snapshot process in a flowchart to export the data to your big data instance. When you run the flowchart, the snapshot data is exported to the Impala database.

The Unica Campaign configuration settings for the Impala dataSource determine how the data is transferred from Unica Campaign to Impala.

1. An administrator must configure the Impala data source (in Campaign | Partitions | Partition[n] | dataSources) to specify the required SCP and SSH commands:
  - The **LoaderPreLoadDataFileCopyCmd** value uses SCP to copy data from Unica Campaign to a location specified under configuration "DataFileStagingFolder" on your Impala-based Hadoop system. This location must be an HDFS location on the Impala server. This value can either specify the SCP command or call a script that specifies the SCP command. See the example below.
  - The **LoaderPostLoadDataFileRemoveCmd** value must specify the SSH "rm" command to remove the temporary file after it is loaded into Impala.

To support this functionality, SSH must be configured on the Unica Campaign listener server. For instructions, see the *Unica Campaign Installation Guide*.

2. Configure the Snapshot process in a flowchart to obtain input data from one or more data sources and export the data to your Impala database. Design the flowchart as you normally would, including any desired processes such as Select and Merge.
3. Run the flowchart.

The entire dataset is exported to a temporary data file at <Campaign\_Home>/partitions/partition[n]/tmp. The temporary file is copied to the Impala server using **LoaderPreLoadDataFileCopyCmd** and the data is loaded into a Impala table. The temporary file is removed from the Impala server using **LoaderPostLoadDataFileRemoveCmd**.

**Example: Configuring export to Cloudera using a script:** Using a script can be useful to avoid file permission issues. If there are any issues related to file permissions, the LOAD command cannot access the data file and the command fails. To avoid this type of issue, you can write your own shell or command-line script to SCP the data file to Hive and update the file permissions of the data file. The following example shows Unica Campaign configured to use a script for export to Cloudera. **LoaderPreLoadDataFileCopyCmd** calls a script that uses SCP to copy the data file from the local machine running Unica Campaign to a HDFS directory on the remote Cloudera machine.

**LoaderPostLoadDataFileRemoveCmd** removes the file.

```
Campaign | Partitions | Partition[n] | dataSources | Impala_Cloudera | LoaderPreLoadDataFileCopyCmd = /opt/HCL/CampaignBD/Campaign/bin/copyToHadoop.sh <DATAFILE>
```

```
Campaign | Partitions | Partition[n] | dataSources | Impala_Cloudera | LoaderPostLoadDataFileRemoveCmd = ssh cloudera@example.company.com "rm /tmp/<DATAFILE>"
```

Here is the script that is called by **LoaderPreLoadDataFileCopyCmd**: `copyToHadoop.sh: #!/bin/sh scp $1 cloudera@example.company.com:/tmp ssh cloudera@example.company.com "chmod 0666 /tmp/'basename $1'"`

The script is on the Unica Campaign listener machine. The script executes the SCP command as the user "cloudera" on the destination server (example.company.com) to copy the file to the HDFS directory. The SSH command connects as the same user to make sure that the permissions are correct for the load and removal processes that will follow.

## Impala query language conformance

The following guidelines apply when Unica Campaign is integrated with Impala-based big data sources. Please refer to Cloudera documentation for HiveQL features not available in Impala

- The SQL must conform with HiveQL.
- If you are writing raw SQL queries for use in Impala Unica Campaign, confirm that the queries work on Impala .
- Using multiple SQL statements for raw SQL queries is not supported.
- You might need to modify existing queries for Impala if you are using raw SQL in Unica Campaign process boxes, custom macros, or derived fields, for pre- and post-processing.

## Mapping and unmapping user tables

User tables contain customer data for use in marketing campaign flowcharts. User tables must be mapped in Unica Campaign to make them available in flowcharts.

You can map, unmap, and remap user tables by selecting **Settings > Campaign settings > Manage table mappings**. Or, if you are editing a flowchart, select **Admin > Tables**. You can also map user tables when you configure a Select process in a flowchart.

Before you map user tables:

- Verify that the user tables contain only the data types that Unica Campaign supports. See [Supported data types for user tables \(on page 52\)](#).
- Define audience levels because you must select an audience level during the user table mapping process. See [Audience level administration \(on page 133\)](#).

After you map user tables, you can store the table mappings in table catalogs for re-use.



**Note:** When you are editing a flowchart, if you make any changes to the table mapping and close the flowchart without saving it, the table mapping changes might be visible in the **Table mappings** window after you re-open the flowchart. However, these changes are not saved until you save the flowchart.

## Naming guidelines for mapping user tables

Follow these guidelines when you create mapped table and field names.

- Do not include spaces in the name.
- Begin the name with an alphabetic character.
- Do not use any unsupported characters. For more information about unsupported characters and naming restrictions for Unica Campaign objects, see [Special characters in Unica Campaign object names \(on page 542\)](#).
- Do not to use function names or keywords from the Macro Language for column headings on tables mapped from a database or a flat file. These reserved words can cause errors if used in column headings on mapped tables. For details about these reserved words, see the *Macros for Unica User's Guide*.
- Field names are not case-sensitive. If a field has been mapped, you can change the case in the field name without affecting the mapping.
- Do not use the following reserved keywords to map user tables: `AGF`, `DF`, `ICGF`, `UCGF PDF`, `ZN`, `UserVar`. If the reserved keywords are already used to map tables, remap the tables by using different names.

## Mapping a base record table to an existing database table

Map a new base record table to make data accessible to processes in flowcharts. You can map a new base record table to an existing database table.

Before you map a user table, ensure that the table uses only the data types that Unica Campaign supports.

1. Select **Settings > Campaign settings**, then click **Manage table mappings**.



**Note:** If you are editing a flowchart: Either begin configuring a Select process or open the **Admin** menu and select **Tables**. When you access the table mapping wizard from a Select process, the **Dimension table** and **General table** options are not listed.

2. In the **Table mappings** dialog, click **Show user tables**.
3. Click **New table**.
4. In the **New table definition - Select the table type** dialog, select **Base record table** and click **Next**.
5. Select **Map to existing table in selected database**, select the data source name, then click **Next**.
6. In the **Source table** list, select the existing table that you want to map.

The tables are listed in alphabetical order by <owner> . <table\_name>. If you do not see the table that you are looking for, check whether the data source is configured to filter specific table entries. From 12.1.7 onwards, user can perform more versatile searches by looking for specific strings or substrings within the table names.

The source fields in the table that you selected are mapped automatically to fields in the base record table that you are creating. To change the automatic mappings, select fields from the **Source table fields** list or the **New table fields** list, and use the **Add >>**, **Remove <<**, **Move up**, and **Move down** buttons, until the table is mapped as required.

You can click on the **Field name** column heading in the **New table fields** section to automatically sort column names alphabetically in ascending (or descending) order.

7. Click **Next**.

8. Optionally, use the **Unica Campaign Table name** field to change the name that Unica Campaign will use for the base record table to a more user-friendly value. You can also optionally change the field names in the **Unica Campaign Field name** field.
9. Click **Next**.
10. Select an Audience level from the list. The **Audience fields** list is automatically populated with the fields required to define the selected audience level. You must use the **Fields from this table** field to match one or more fields in the new base table that correspond to each required key.
11. If each unique audience ID appears no more than once in the current base table, check **This audience level is normalized in this table**.  
Setting this option correctly is important for correct configuration of options in an **Audience** process. If you don't know the correct setting, leave this option unchecked.
12. Click **Next**.
13. (Optional) On the **Specify additional audience levels** screen:
  - a. Click **Add** if you want to specify one or more additional audience levels contained in the base record table. Adding additional audience levels allows users to use this table as a "switching table" to translate from one audience level to another using the **Audience** process in a flowchart.
  - b. Use the **Audience level and ID fields** dialog to select an **Audience level name**. For example, select Household or Customer. For each field in the **Audience fields** list, use the **Fields from this table** field to select a corresponding field. In this way, you match appropriate field(s) from the base table to the corresponding key(s) in the audience level.
  - c. If each unique audience ID appears at most once in the current base table, check **This audience level is normalized in this table**.
  - d. Click **OK**.
  - e. Repeat these steps for each audience level that you want to add for the base table.
14. Click **Next** in the **Specify additional audience levels** dialog.
15. If dimension tables exist in the current table catalog, the **Specify relationship to dimension tables** dialog opens:
  - a. Check the box to the left of any dimension table that is related to the base record table you are creating.
  - b. For each related dimension table, in the **New table's key fields(s) to use** list, select a field from the base table to match each listed key in the **Dimension table's key field(s)** list, then click **Next**.
16. In the **Specify the fields for which a stored list of values will be computed** dialog: You can choose to pre-compute the distinct values and frequency counts for specific fields, or allow users to perform real-time profiling of data in the base record table.
17. Click **Finish**.
18. Click **Close**. You are prompted to save the table mappings to a catalog file. A table catalog is a collection of mapped user tables. Saving mappings to a catalog makes it easier to access them in the future.

You have created a base record table based on an existing database table. The new base table is part of the current table catalog, which you can manage through the table manager.

## Mapping a base record table to an existing fixed-width flat file

Map a new base record table to make data accessible to processes in flowcharts. You can map a new base record table to an existing fixed-width flat file on the Unica Campaign server within your partition. The file must be located under your partition root.

Before you map a user table, ensure that the table uses only the data types that Unica Campaign supports.

1. Use one of the following methods to open the **New table definition - Select the table type** dialog:
  - If you are editing a flowchart, open the **Admin** menu and select **Tables**, or
  - If you are editing a flowchart, begin configuring a Select process, or
  - Select **Settings > Campaign settings**, then click **Manage table mappings**.
2. If necessary, select **Base record table**, then click **Next**.
3. Select **Map to existing file**, then click **Next**.
4. Keep the default **File type** selection of **Fixed-width flat file**.
5. In the Settings section of the window, click **Browse** to select the **Source file** from within your campaign partition root directory. Unica Campaign automatically populates the **Dictionary file** field with the same path and file name, except with the **.dct** extension. You can override this entry if necessary.

## Mapping a base record table to an existing delimited file

Map a new base record table to make data accessible to processes in flowcharts. You can map a new base record table to an existing delimited file on the Unica Campaign server within your partition. The file must be located under your partition root.

Before you map a user table, ensure that the table uses only the data types that Unica Campaign supports.



**Important:** Unica Campaign does not support the use of the double quote character (") in field entries of delimited files. If any of the field entries contain double quotes, change them to another character before mapping a table to the file.

1. Use one of the following methods to open the **New table definition - Select the table type** dialog:
  - If you are editing a flowchart, open the **Admin** menu and select **Tables**, or
  - If you are editing a flowchart, begin configuring a Select process, or
  - Select **Settings > Campaign settings**, then click **Manage table mappings**.
2. If necessary, select **Base record table**, then click **Next**.
3. Select **Map to existing file**, then click **Next**.
4. For **File type**, select **Delimited file**.
5. In the Settings section, check **First row of data contains field names** if you want to automatically use the first row of data to define the fields of the base table.  
You can override these values later.
6. Select a **Field delimiter** to indicate which character separates each field in a row of data: **TAB**, **COMMA**, or **SPACE**.

7. Select the **Qualifier** that is used to indicate how strings are delimited in the file: **None**, **Single quote**, or **Double quote**.

This setting is important if you have a space-delimited file with spaces in field entries. For example, if you have a row of data such as: "John Smith" "100 Main Street", set the Field Delimiter to **Space** and set the Qualifier to **Double quote** so the record is correctly parsed as two fields (name and street address).

8. Click **Browse** to select the **Source file** from within your partition directory.
9. Define the fields for the new table:

Use the **Add** and **Remove** buttons to specify which **Source table fields** to include in the new table. By default, all fields in the file are listed.

Use the **Up 1** and **Down 1** buttons to adjust field order. Click the **Field name** column in the **New table fields** section to automatically sort column names alphabetically in ascending or descending order.

You can adjust the field type (Numeric or Text) and width, which are automatically detected based on the **numRowsReadToParseDelimitedFile** configuration setting. For example, if IDs are detected to be 2 characters wide, but you know that IDs can consist of up to 5 characters, increase the value to 5.



**Important:** If the width value is too small, an error may occur.

10. Click **Next**.
11. Accept the defaults on the **Specify table name and field information** screen, or edit the **Unica Campaign Table name** field to change the name of the table as it will appear in Unica Campaign. You can also change the **Unica Campaign Field name** mapped to each source field name by selecting the field name and editing the text in the **Unica Campaign Field name** text box in the **Edit selected field information** section.
12. Click **Next**.
13. On the **Specify the selected table's audience level and assign ID field(s) to it** screen, select an **Audience level** from the list. The **Audience fields** list is automatically populated. You must select a field in the new base table that is the corresponding key for each listed entry.
14. Click **Next**.  
The **Specify additional audience levels** screen opens.
15. Optionally, you can specify one or more additional audience levels contained in the base record table. Adding additional audience levels allows users to use this table as a "switching table" to translate from one audience level to another using the **Audience** process in a flowchart.
  - a. Click **Add**.
  - b. Select an **Audience level name**.
  - c. For each **Audience field**, match appropriate field(s) from the base table to the corresponding key(s) in the audience level.
  - d. If each unique audience ID appears at most once in the current base table, check **This audience level is normalized in this table**.

- e. Click **OK**.
  - f. Repeat steps a through e for each audience level that you want to add for the base table, then click **Next**.
16. You can choose to pre-compute the distinct values and frequency counts for specific fields, or allow users to perform real-time profiling of data in the base record table.
  17. Click **Finish**.

You have created the base record table based on an existing file. The new base table is part of the current table catalog, which you can manage through the table manager.

## Mapping dimension tables

Map a new dimension table to make data that augments data in a base table, such as demographics based on postal codes, accessible to processes in flowcharts.

Before you map a user table, ensure that the table uses only the data types that Unica Campaign supports.

Dimension tables must be mapped to a database table. Also, dimension tables must be related to one or more base tables that are mapped to tables in the same data source (that is, the same database). In defining the dimension table, you can specify the join conditions between the base and dimension tables.

1. Use one of the following methods to open the **New table definition - Select the table type** dialog:
  - If you are editing a flowchart, open the **Admin** menu and select **Tables**, or
  - Select **Settings > Campaign settings**, then click **Manage table mappings**.



**Note:** You cannot map a dimension table from a Select process.

2. Select **Dimension table**, then click **Next**.
3. Select the table that you want to map from the **Source table** list.

The source fields in the selected table are mapped automatically to fields in the base dimension table that you are creating. To change the default selections, select fields from the **Source table fields** list or the **New table fields** List, and use the **Add**, **Remove**, **Up 1**, and **Down 1** buttons, until the table is mapped as required, then click **Next**.



**Note:** You can click on the Field name column in the New table fields section to automatically sort column names alphabetically in ascending or descending order.

4. (Optional) Change the name Unica Campaign will use for the dimension table and its fields.
  - a. To change the table name, edit the name in the **Unica Campaign table name** field.
  - b. To change a field name, select the mapping in the **New table fields** list, and edit the text in the **Unica Campaign Field name** field, then click **Next**.
5. Specify the dimension table's key(s), and how the table will be joined to base record tables.
6. Select one or more keys in the **Key field(s)** list.



7. Check **Key field(s) are normalized in this table**, if this is the case.
8. Select the **Table join method**, then click **Next**.
  - The option **Always use inner join** will always use an inner join between the base table and this dimension table, returning only audience IDs from the base table that appear in the dimension table.
  - The **Always use outer join** option will always perform an outer join between the base table and this dimension table (which provides the best results if the dimension table is known not to contain at least one row for every audience ID in the base table).
  - The default setting of **auto** uses an inner join in the Select and Segment processes and an outer join in output processes (Snapshot, Mail List, and Call List). This typically provides the desired behavior where values in the dimension table are required for selection criteria, but NULLs should be output for missing audience IDs for any dimension table fields that are output.
9. If base record tables exist, the Specify Relationship to Base Tables screen opens. Check the box to the left of any base record table that is related to the dimension table you are creating Specify the join fields, then click **Next**.
10. You can choose to pre-compute the distinct values and frequency counts for specific fields, or allow users to perform real-time profiling of data in the base record table.
11. Click **Finish**.

You have created the dimension table. The data can now be used in flowchart processes.

## Mapping a general table to a database table

You can map a new general table to an existing database table. You map a new general table to export Unica Campaign data for use by other applications.

1. Use one of the following methods to open the **New table definition - Select the table type** dialog:
  - If you are editing a flowchart, open the **Admin** menu and select **Tables**, or
  - Select **Settings > Campaign settings**, then click **Manage table mappings**.
2. Select **General table**, then click **Next**.
3. Select **Map to existing table in selected database**, select the customer database name, then click **Next**.
4. Select the table you want to map from the **Source table** list.

The source fields in the table you select are mapped automatically to new table fields in the general table you are creating. To change the automatic mappings, select fields from the **Source table fields** list or the **New table fields** list, and use the **Add**, **Remove**, **Up 1**, and **Down 1** buttons, until the table is mapped as required, then click **Next**.

5. (Optional) Change the name which Unica Campaign will use for the general table and its fields.

To change the table name, edit the name in the **Unica Campaign table name** field.

To change a field name, select the mapping in the **New table fields** list, and edit the text in the **Unica Campaign field name** field.

6. Click **Finish**

You have created the general table based on a database table.

## Mapping a general table to a file

You map a new general table to export Unica Campaign data for use by other applications.

1. Use one of the following methods to open the **New table definition - Select the table type** dialog:
  - If you are editing a flowchart, open the **Admin** menu and select **Tables**, or
  - Select **Settings > Campaign settings**, then click **Manage table mappings**.
2. Select **General table**, then click **Next**.
3. Select **Map to existing file**, then click **Next**.
4. To map the table to a fixed-width flat file:
  - a. Keep the default **File type** selection.
  - b. Click **Browse** to select the **Source file**. Unica Campaign automatically populates the **Dictionary file** field with the same path and file name, except with the **.dct** extension. You can override this entry if necessary.
5. To map the table to a delimited file:
  - a. Select **Delimited file** for the **File type**.
  - b. Check **First row of data contains field names** if this is the case.
  - c. Select a **Field delimiter** to indicate which character separates each field in a row of data: **TAB**, **COMMA**, or **SPACE**.
  - d. Select the **Qualifier** used to indicate how strings are delimited in the file: **None**, **Single quote**, or **Double quote**.
  - e. Click **Browse** to select the **Source file**, then click **Next**.  
The **Specify the new table fields** window opens.
6. Specify the fields to use in the new table. By default, all fields in the file are listed.

Use the **Add**, **Remove**, **Up 1** and **Down 1** buttons to specify the **Source table fields** that are included in the new table and their order.

You can adjust the field type (Numeric or Text) and width, which are automatically detected based on the **numRowsReadToParseDelimitedFile** configuration setting. For example, if IDs are detected to be 2 characters wide, but you know that IDs can consist of up to 5 characters, increase the value to 5.



**Important:** If the width value is too small, an error may occur.



**Note:** When you export data to a fixed-width flat file on disk, you can override the preset field lengths by editing the data dictionary for that file.

7. Click **Next**.

The **Specify table name and field information** window opens.

8. Accept the defaults, or edit the **Unica Campaign table name** field to change the name of the table as it will appear in Unica Campaign; and change the Unica Campaign field names that are mapped to the source field names.
9. Click **Finish**.

You have created a general table based on a file.

## Configuring profiling when mapping user tables

When you map a user table, you can either pre-compute the distinct values and frequency counts for specific fields, or allow users to perform real-time profiling of data in the base record table.

Profiling enables users to see a table's values when editing a flowchart, without having to view raw data and allows users to easily select from valid values when building a query. Pre-computed profiles provide quick access to the distinct field values and counts without querying the database. Real-time profiling provides access to the most up-to-date data and may be helpful if your database updates frequently. If you pre-compute profiles, you can control how often profiles are regenerated.

You can choose to both pre-compute profiles and allow users to dynamically profile in real-time, or you can choose to disallow real-time profiling and force users to always use the pre-computed profiles.

1. When mapping a user table, check the fields for which you want Unica Campaign to pre-compute distinct values and frequency counts.

By default, Unica Campaign stores the pre-computed profiles in the `Campaign > partitions > partition[n] > profile category as data source_table name_field name`.

2. If you have distinct values and counts stored in a separate database table that Unica Campaign should use, click **Configure Data Sources**. Select **Data pre-defined in a table**, select the table name, the field containing the values, and the field containing the counts. Then click **OK**.
3. Check **Allow real-time profiling** to have Unica Campaign update its records of values for the selected fields in real-time. This option allows users editing a flowchart to see the current values for these fields. However, it also requires a database query each time a user clicks **Profile**, which can potentially degrade performance.



**Note:** Enabling or disabling the **Allow real-time profiling** option applies to all table fields, not just the checked ones.

If you disallow real-time profiling and do not specify an alternate means for pre-generated profiles, users will not be able to view any values or counts for any fields in this table.

If you disallow real-time profiling and provide pre-computed profiles for one or more fields, users will be able to access the pre-computed profiles, which is for the entire table. Users will not be able to profile only the values for the input cell of a process.

To provide the greatest flexibility, you should allow real-time profiling.

## Remapping user tables

You can remap a user table at any time.

You may want to remap a user table to:

- Remove unnecessary fields to simplify working with a table.
- Add new fields that need to be available.
- Rename the table or its fields.
- Add an audience level.
- Change profiling characteristics.

If you remove a field that is referenced in a flowchart or change the name of the table or referenced fields, you will cause the flowchart to become unconfigured. You must then manually edit each process box where the table is used to correct references.

Remapping a user table changes only the local table mapping for the current flowchart. To save the updated table mapping to the table catalog, you must save the table catalog. Once saved into the table catalog, any flowchart subsequently using or importing that table catalog will see the changes.

1. Use one of the following methods:
  - If you are editing a flowchart, open the **Admin** menu and select **Tables**, or
  - Choose **Settings > Campaign settings > Manage table mappings**.
2. In the **Table mappings** dialog, click **Show user tables**.
3. Select the mapped table that you want to remap.
4. Click **Re-map table**.
5. Perform the same steps as you would when mapping a table.

## Unmapping user tables

You can unmap a user table at any time. Unmapping a user table causes any process in the current flowchart referencing that user table to become unconfigured. However, unmapping a table does not delete the original underlying data or affect any other flowchart.



**Important:** You cannot reverse this process. To restore an unmapped table, you must map it as if for the first time, or import a stored table catalog that contains the mapped table definition. If you are uncertain about



permanently unmapping a table, you can always save your current table mappings into a table catalog for later restoration if needed.


1. Use one of the following methods:
  - If you are editing a flowchart, open the **Admin** menu and select **Tables**, or
  - Choose **Settings > Campaign settings > Manage table mappings**.
2. Select the table to unmap.
3. Click **Unmap table**. You are prompted to confirm.
4. Click **OK** to unmap the table.

## Filtering the mapped user tables

You can filter the mapped user tables that appear in the Table mappings window. The mapped user tables can be filtered by table name.

In the Select process box, you can filter the mapped user tables that appear in the Available Fields section in the Select process configuration window. For more information, see the Unica Campaign User Guide.

To filter the mapped user tables by name, complete the following steps:

1. Go to **Settings > Campaign settings > Manage table mappings > User Tables**.
2. You can also access table mappings from a flowchart. Open a flowchart for editing and click **Admin > Tables**.
3. Click the Filter icon  next to **User table name**.
4. Enter the text to filter the tables and click **Apply**.

You cannot use any special characters except underscore in the filter text. The search is not case sensitive.

The filtered list of tables is displayed. The Filter icon changes to a dark green color.

5. To clear the filter, click the Filter icon , and in the **Filter** window, click **Clear**.

The filter is applicable until you click **Clear** or you close the **Table mappings** window.

The filter is also cleared if you complete any other action in the **Table mappings** window. For example, if you click **Load** or **Compute**, the filter is cleared and all the user tables are listed.


## System table administration

Unica Campaign administrators may perform the following tasks pertaining to system tables.

### Mapping or remapping system tables

Most system tables are mapped automatically during initial installation and configuration if you used the system table data source `UA_SYSTEM_TABLES`. Use caution when mapping Unica Campaign system tables because they contain application data.

For important information about mapping system tables, see the installation documentation. Abbreviated instructions are provided below for convenience.

 **Important:** Do not map or remap system tables when users are using Unica Campaign.

1. Select **Settings > Campaign settings**.
2. Click **Manage table mappings**.
3. In the Table Mappings dialog, select **Show system tables**.
4. Select the table to map from the **Unica Campaign system table** list and double-click it or click **Map table** or **Re-map table**.

The **Select source database and match required fields** dialog opens.

5. If the table is not automatically selected in the **Source table** list, select it. Entries are alphabetically listed by owner.table names. Source table fields in the Unica Campaign database are automatically mapped to required fields. For system tables, you do not need to add or remove any field mappings; all field entries are matched automatically.




**Note:** When you map a system table, do not select a different table from the Source table list or you cannot complete the mapping. If you do this accidentally, click **Cancel** and select the correct table in the **Table Mappings** dialog.

6. Click **Finish**.

## Unmapping system tables

Unmapping a system table can cause features and existing campaigns to stop working. If you must unmap a system table, unmap it only when no users are using Unica Campaign.

 **Important:** Unmapping a system table without remapping the table can cause significant application problems.

1. Select **Settings > Campaign settings**.
2. Click **Manage table mappings**.
3. In the Table Mappings dialog, select **Show system tables**.
4. Select a table in the **Unica Campaign system table** list, then click **Unmap table**.  
You are prompted to confirm the unmapping.

Immediately remap the system table unless you are sure that it does not need to be mapped in your environment.

## Mapping Customer audience level system tables


Unica Campaign is delivered with an audience level called `Customer`. If you plan to use this audience level, you must map the `Customer` audience level tables after Unica Campaign is installed.

The system database tables that support the `Customer` audience level are created when you run the provided system table creation script, as described in the installation documentation. After installation, you must map the tables as follows.



**Note:** If your key for selection is different, you can modify the supplied contact history and response history tables or create your own for your specific needs.

1. Select **Settings > Campaign settings**.
2. Click **Manage table mappings**.
3. In the **Table mappings** dialog, click the heading Unica Campaign System Table to sort the list alphabetically.
4. Double-click an item in the list of system tables to map it to the appropriate Database table name, as indicated below.

Unica Campaign system table	Database Table Name
Customer Contact History Table	UA_ContactHistory
Customer Detailed Contact History Table	UA_DtlContactHist
Customer Response History Table	UA_ResponseHistory
Customer Segment Membership Table	UA_SegMembership
 <b>Note:</b> If you are not using strategic segments, do not map this table. For more information, see <a href="#">About mapping segment membership tables (on page 79)</a> .	
Send Time Optimization	UA_STO_METRICS

5. Close the **Table mappings** dialog.

## About mapping segment membership tables

The segment membership table is one of the audience level system tables that Unica Campaign creates when you define a new audience. If you use strategic segments in Unica Campaign flowcharts or Optimize sessions within Unica Optimize, you must map the segment membership table to a database table that defines segment members.

For example, if you plan to use the default `Customer` audience in conjunction with strategic segments, you must map the `Customer Segment Membership` system table to the `UA_SegMembership` segment membership database table. For other audiences used with strategic segments, you would map the system table `<audience name> Segment Membership` to the database table that defines the segment members. You can use `UA_SegMembership` as a template for the database table.

Running the Create Seg process populates a database table if you have mapped it to a segment membership system table. If you run the Create Seg process when the database table is not mapped to the segment membership system

table, you must re-run the Create Seg process to populate the table if you map it later. Otherwise, Optimize sessions in Unica Optimize that use strategic segments may provide inaccurate results.

### If you are not using strategic segments in flowcharts or Optimize sessions

Using strategic segments in Unica Campaign flowcharts and Unica Optimize sessions is optional. If you are not using strategic segments, the best practice is that you do not map the segment membership table. Mapping an audience's segment membership system table causes Unica Campaign or Unica Optimize to refresh the table each time you run the flowchart or Optimize session that includes the audience. This is unnecessary processing overhead if you are not using strategic segments.

## About mapping STO tables

For Send time optimization feature, Deliver will populate best send time of each user being targeted via Deliver in a table called as UACE\_STO\_METRICS and Campaign will use this time to populate OLT.

Since identification of user in Campaign is based on AudienceLevel, so this table needs to be AudienceLevel specific. User needs to create this table per audience level and map it under Campaign Settings -> Manage Audience levels section. This table needs to have audience id fields and STO calculation related fields.

### If you are not using send time optimization

Using send time optimization feature in Campaign flowchart is optional. If you are not using Send time optimization features in flowcharts, the best practice is that you do not map the send time optimize table. Mapping send time optimization system table causes Unica Campaign to add two new columns 'preferredSendTime' and 'TimeZone' in deliver's OLT table. This is unnecessary processing overhead if you are not using send time optimization feature.

## Unmapping the segment membership table

The segment membership table is one of the audience level system tables that Campaign creates when you define a new audience. When you unmap the segment membership table you must also clear existing cache files and restart the Unica Campaign and Unica Optimize listeners.



**Note:** If you are using Unica Optimize, do not change the mapping of the segment membership table for an audience while an Optimize session that uses the audience is running.

1. In Unica Campaign, unmap the segment membership table for the audience: Select **Settings > Campaign settings**, click **Manage table mappings**, select **Show system tables**, select a table, and click **Unmap table**.
2. Delete `unica_tbmgr.cache` from the `conf` directory of your Unica Campaign installation.

By default, this file is located at `Campaign\partitions\<partition[n]>\conf`.

3. Delete `unica_tbmgr.cache` from the `conf` directory of your Unica Optimize installation.

By default, this file is located at `Optimize\partitions\<partition[n]>\conf`.



4. Restart the Unica Campaign listener (`unica_aclsnr`).
5. Restart the Unica Optimize listener (`unica_aolsnr`).

## Viewing system table contents

For convenience, you can view the contents of most system tables from the Unica Campaign table manager.

You can only view the first 1000 rows of data in a table. Therefore, this function is of limited use for very large tables such as contact and response history tables. You cannot edit the system table data while viewing it.

1. Select **Settings > Campaign settings**.
2. Click **Manage table mappings**.
3. Select **Show system tables**.
4. Select a system table and click **Browse**.  
A window opens showing the table data.



**Note:** Browse table feature in campaign table mapping has a design limitation. There could be performance issues in displaying the data for large tables.

5. Click any column to sort on it. Click the column again to reverse the sort order. To close the window, click **X** in the upper-right corner.



**Note:** User needs to be careful while using the delete button.



**Note:** The information provided by the **Column In Use** column is designed to help data cleanup. This enables the user to remove columns or tables that are no longer in use and maintain data efficiency and organization.

## Data dictionary administration

A data dictionary defines the format of data in a fixed-width ASCII flat file. You use data dictionaries in Snapshot processes to ensure that any fixed-width output files that you create adhere to a specific structure.

You can edit the data dictionary for a base or general table, or create a new data dictionary from an existing fixed-width flat file.



**Note:** The data dictionary must be stored on the Unica Campaign server or be accessible from the server to be used for table mapping.

## What is a data dictionary?

A data dictionary is a file that defines the format of data in a fixed-width ASCII flat file that is used in Unica Campaign either as a base table or as a general table.

A data dictionary is required to interpret the structure and format of a fixed-width ASCII text file. It defines the field names, their order, their data type (string or numeric) and the byte positions they occupy in the file. Data dictionaries are automatically created for fixed-width flat files created by Unica Campaign and these typically do not need to be manually created or edited.

You use data dictionaries in output processes such as Snapshot, Mail List, and Call List to ensure that the flat file tables you create adhere to a specific structure.

A data dictionary defines the table fields, data types, and sizes. You can develop data dictionaries for vendor or channel-specific output and reuse them to create output of a predetermined format.

If you are using a fixed-width flat file created by a non- third-party application, you may need to manually or programmatically create an associated data dictionary. Or you may want to copy an existing data dictionary and edit it to create a new file. You also can edit a data dictionary to change field names. You must be sure that you are not corrupting data if you choose to edit any of the other fields in the data dictionary.

## Editing data dictionaries

Follow these instructions to edit a data dictionary for use with a Snapshot process. A data dictionary defines the format of data in a fixed-width ASCII flat file to ensure that any fixed-width output files that you create adhere to a specific structure.

1. Find the data dictionary you want, then open it using Notepad or any other text editor.
2. Change information as needed in the file, making sure that the data that will be stored in the associated table can use the parameters you are setting.
3. To apply changes to the data dictionary, you must save, close, and reopen the flowchart.

A data dictionary file looks similar to the following example:

```
CellID, ASCII string, 32, 0, Unknown,
MBRSHP, ASCII string, 12, 0, Unknown,
MP, ASCII Numeric, 16, 0, Unknown,
GST_PROF, ASCII Numeric, 16, 0, Unknown,
ID, ASCII Numeric, 10, 0, Descriptive/Names,
Response, ASCII Numeric, 10, 0, Flag,
AcctAge, ASCII Numeric, 10, 0, Quantity,
acct_id, ASCII string, 15, 0, Unknown,
src_extract_dt, ASCII string, 50, 0, Unknown,
extract_typ_cd, ASCII string, 3, 0, Unknown,
```

## Creating data dictionaries

You can create a new data dictionary manually. It may be easier to start with an existing data dictionary that was created by Unica Campaign.

A data dictionary defines the format of data in a fixed-width ASCII flat file. You use data dictionaries in Snapshot processes to ensure that any fixed-width output files that you create adhere to a specific structure.

1. Create an empty .dat file (length = 0) and a corresponding .dct file.
2. In the .dct file, define fields in the format:

```
<Variable_name>, <"ASCII string" or "ASCII Numeric">, <Length_in_bytes>, <Decimal_point >, <Format>,
<Comment>
```

Use `Unknown` for the format and leave the comment field blank, as shown in the following example:

```
acct_id, ASCII string, 15, 0, Unknown,
hsehld_id, ASCII Numeric, 16, 0, Unknown,
occpth_cd, ASCII string, 2, 0, Unknown,
dob, ASCII string, 10, 0, Unknown,
natural_lang, ASCII string, 2, 0, Unknown,
commun_lang, ASCII string, 2, 0, Unknown,
```

3. You can now map a new table to a file using this data dictionary.

## Data dictionary syntax

Each line in a data dictionary defines a field in fixed-width flat file, using the syntax described here.

```
<Variable_name>, <"ASCII string" or "ASCII Numeric">, <Length_in_bytes>, <Decimal_point >, <Format>, <Comment>
```

The `<Decimal_point >` value specifies the number of digits following the decimal point and is valid for ASCII Numeric fields only. For ASCII string fields, the value should always be 0.

Unica Campaign does not use the Format and Comment fields. For best results, use "Unknown" for the format value and leave the Comment field blank.

A data dictionary file looks similar to the following example:

```
CellID, ASCII string, 32, 0, Unknown,
MBRSHP, ASCII string, 12, 0, Unknown,
MP, ASCII Numeric, 16, 0, Unknown,
GST_PROF, ASCII Numeric, 16, 0, Unknown,
ID, ASCII Numeric, 10, 0, Descriptive/Names,
Response, ASCII Numeric, 10, 0, Flag,
AcctAge, ASCII Numeric, 10, 0, Quantity,
acct_id, ASCII string, 15, 0, Unknown,
src_extract_dt, ASCII string, 50, 0, Unknown,
extract_typ_cd, ASCII string, 3, 0, Unknown,
```

For example, the line:

```
acct_id, ASCII string, 15, 0, Unknown,
```

means that a record in the file has a field called `acct_id`, which is a 15-byte string with no decimal point (as the field is a string) with an unknown format and empty comment string.

## Table catalog administration

A table catalog is a collection of mapped user tables. Administrators can create and load table catalogs and perform other operations to make table catalogs available to Unica Campaign users.

### What is a table catalog?

A table catalog is a collection of mapped user tables. Table catalogs store all user table mapping metadata information for re-use across flowcharts. You can also map contact and response history tables in a table catalog for inclusion and exclusion rules.

Table catalogs are stored by default in a binary format using a `.cat` extension. They can also be saved as XML files.

You use table catalogs to:

- Easily save, load, and update commonly used user tables.
- Create alternative data mappings (for example, to switch between running against a sample database and the production database).

You can save mapped user tables to a table catalog, then use the same table catalog in other flowcharts. This means you can:

- Make changes to the table catalog in one flowchart, then propagate these changes to other flowcharts by importing the updated table catalog into each flowchart.
- Retain the internal catalog that you first loaded for that flowchart, although you copy it to other flowcharts and make changes there.
- Make different changes to the internal catalogs of a number of different flowcharts, starting from one "template" table catalog.

### Creating table catalogs

You create a table catalog by saving the user tables that are in the current flowchart's internal table catalog. Saving table catalogs with commonly defined table mappings makes it easy to share or restore table mappings.



**Note:** You can also access table catalogs from the **Options** menu while editing a flowchart.

Follow the steps below to create a table catalog.

1. Select **Settings > Campaign settings**.
2. Click **Manage table mappings**.

3. In the Table Mappings dialog, select **Show user tables**. The user tables that you want to save as a table catalog must be mapped in Unica Campaign.
4. In the Table Mappings dialog, User tables can be set in a specific order using **up** and **down** arrow keys, users can place the frequently used user tables at the top of the table mapping list using up and down arrow keys. All process boxes which are showing the table listing maintain the same order of user tables which is set by the user during table mapping so that users can easily access the frequently used user tables immediately.
5. Select the user tables that you want to save as a catalog, and click **Save**.
6. In the Save Tables dialog, specify whether to save all table mappings to the table catalog or save only the selected table mappings to the table catalog, then click **OK**.

The Save Table Mappings to Catalog File dialog opens.

7. Enter a name for the table catalog. If you use .XML as the extension, the table catalog will be stored in XML format rather than as a binary .cat file.

Saving a table catalog as XML makes it possible to view and interpret the values. XML format is especially useful for editing purposes. A common use of the XML format is to globally search and replace all references to a production data source name with a test data source name. This makes it easy to make table catalogs portable between data sources.



**Note:** The name must be unique within the folder; otherwise you will be prompted to overwrite the existing table catalog with the same name. The name cannot contain any periods, apostrophes, or single quotation marks, must begin with a letter and can only contain the letters A-Z, the numbers 0-9, and the underscore character (\_).

8. (Optional) Describe the table catalog in the **Note** field.
9. Decide whether to store authentication information with the catalog:
  - If you leave **Save with database authentication information** unchecked, users of the table catalog will need to provide a database login and password for any data sources referenced in the table catalog. These passwords may be already stored in their ASM user profile. If a user does not already have a stored valid login and password, the user is prompted to provide them. This setting is the best practice for security purposes.
  - If you check **Save with database authentication information**, the authentication information that you are currently using to access the data sources is saved with the table catalog. Anyone with access permissions to this table catalog will automatically be connected to the data sources using the authentication stored in the table catalog. Therefore, users of this catalog will not need to provide a login or password to access the data sources and will have all of the privileges of the stored login for reading and writing to the data source. You may want to avoid this setting for security reasons.
10. Use the **Save under** option to specify where to save the catalog.

If you do not select a specific folder, or choose **None**, the catalog is saved at the top level. If you prefer to organize table catalogs in folders, select a folder from the **Items list**, or create a folder using the **New folder** button.

11. Click **Save**.

The table catalog is saved as a binary `.cat` file if no extension was provided, or as an XML file if you included `.xml` in the file name.

## Loading stored table catalogs

If you saved mapped user tables to a table catalog, the catalog can be loaded for use in flowcharts.



**Note:** If you defined the `default.cat` table catalog, it is loaded by default each time that you create a new flowchart. However, if you set your browser to accept cookies and you load a different table catalog, that catalog is then loaded by default instead of `default.cat`. This is also true for stored dimension hierarchies.

Follow the steps below to load a stored table catalog.

1. Select **Settings > Campaign settings**.
2. Click **Manage table mappings**.
3. In the Table Mappings window, select **Show user tables**.
4. Click **Load**.
5. Choose one of the following options:
  - **Load table mappings from stored table catalog (existing mappings are cleared):** Completely replace the current mappings (the mapped tables in the flowchart) with the mappings in the catalog that is being loaded. This is the default option.
  - **Merge table mappings from stored table catalog (old mappings are overwritten):** Keep any existing mappings and add any new mappings. Existing table mappings that are not in the new table catalog will be preserved.
6. Click **OK**.

The **Stored table catalogs** dialog opens.

7. Select the name of the table catalog that you want to load.
8. Click **Load catalog**.

## Deleting table catalogs

You can permanently remove a table catalog so that it is no longer available to any flowcharts in any campaigns.

Removing a table catalog deletes the `.cat` file, which points to database tables and possibly flat files. Removing a table catalog does not affect the underlying tables in the database. However, it does permanently remove the catalog file.

**!** **Important:** Only use the Unica Campaign interface to remove table catalogs or perform table operations. If you remove tables or change table catalogs directly in the file system, Unica Campaign cannot guarantee data integrity.

1. Open a flowchart in **Edit** mode.

2. Open the **Options** menu  and select **Stored table catalogs**.

The **Stored table catalogs** dialog opens.

3. Select a table catalog in the **Items list**.

The **Info** area shows the detailed information for the selected table catalog, including the table catalog name and file path.

4. Click **Remove**.

You see a confirmation message asking you to confirm removal of the selected table catalog.

5. Click **OK**.

6. Click **Close**.

The catalog is removed from the **Items list** and is no longer available to any flowcharts in any campaigns.

## Updating pre-computed profiles for tables in a table catalog

If the underlying marketing data has changed, and you are using Unica Campaign to pre-compute profile information for table fields, you must update the table catalog by recomputing the record counts and any pre-computed values that you specified in your tables.

1. Select **Settings > Campaign settings**.

2. Click **Manage table mappings**.

3. In the Table Mappings dialog, select **Show user tables**.

4. To update record counts and values for a subset of user tables, select these tables in the list of tables. You can use Ctrl+click to select multiple tables.

To update record counts and values for all user tables, you do not need to select any tables.

5. Click **Compute**.

The **Recompute** dialog opens.

If you did not select any user tables, **Recompute record counts and list of distinct values for all tables** is selected by default.

If you selected a subset of tables, **Recompute record counts and list of distinct values for selected tables** is selected.




**Note:** If you did not select any tables and you want to enable the option to recompute values for selected tables, click **Cancel** in the **Recompute** dialog. The dialog closes and you return to the **Table mappings** dialog. Now you can select the tables for which you want the record counts and values to be computed.

6. When you are satisfied with your selection, click **OK**.

When the computations are complete, you return to the **Table mappings** dialog.

## Defining data folders for table catalogs

When you create a table catalog, you can specify one or more data folders to associate with that table catalog. In output processes such as Snapshot, these named folders appear in the file location selection dialog as pre-defined folder locations.


1. In a flowchart in Edit mode, open the **Admin** menu  and select **Tables**.
2. In the **Table mappings** dialog, select the mapped user tables that you want to save to a catalog.
3. Click **Save**.
4. In the **Save table mappings to catalog file** dialog, click in the **Unica Campaign data folders** section to add an item.
5. Enter a name and folder location for the data folder that you are adding, relative to the current partition's home directory.  
For example, if you are working in partition1, the folder location that you specify is relative to the `partitions/partition1` folder.
6. Click **Save**.

When you reload the catalog in a flowchart that contains output processes such as Snapshot, the folders appear as options in the file location selection dialog.

For example, say that you add a data folder named `MyFolder` in the folder location `temp`. When you configure a Snapshot process, **File in MyFolder** appears in the **Export to** list. When you select **File in MyFolder**, the **File name** field in the **Specify output file** dialog is automatically populated with the relative path `temp/`.

## Housekeeping of Campaign table catalogs

The Campaign table catalog usage is reported at two levels:

1. Flowchart level - The flowchart level reporting flags the tables used in the flowchart.
  - In a flowchart in Edit mode, open the **Admin** menu  and select **Tables**.
  - In the **Table mappings** dialog, a column called **In Use** is present. This column indicates whether the user table is used in any of the process boxes in the flowchart. If it is in use, then the column value is Yes, otherwise it is No.



- In the Table Mapping dialogue, User tables can be set in a specific order using **up** and **down** arrow keys, users can place the frequently used user tables at the top of the table mapping list using up and down arrow keys, save the order using the **Save Order button**, and the order can be reset to previous one using the **Reset Order button**. All process boxes which are showing the table listing maintain the same order of user tables which is set by the user during table mapping so that users can easily access the frequently used user tables immediately.
- In the **Table mappings** dialog, a new column called **Column In Use** is introduced. This feature provides users with insights about the utilization of columns within the Flowchart. It shows whether a column is used in a process box's expression, which is used during the export of fields to different tables within a process box, or utilized in custom macros, derived fields (df), and persistent derived fields (pdf). On table mapping screen, if the value in Column In Use is Yes, then you can click on the table name and details of the table including the column which are currently in use will be displayed. The **In Use** column is activated only when the user executes the **save and run flowchart** in production mode. The table `UA_TableMappingXColRef` exhibits the column currently in use, accompanied by its respective column name. **UA\_TableMappingXRef** table should be mapped with field **Table Mappings Ref Table & UA\_TableMappingXColRef** table should be mapped with field **Table Mapping Col Ref Table** at **Settings > Campaign Settings > TableMappings**. Users seeking additional information regarding the flowchart, like user table name, type, audience, etc can refer to the database query by combining it with the `UA_TableMappingXRef` table. . The information provided by the **Column In Use** column is designed to help user with data cleanup, enabling the removal of columns or tables that are no longer in use, and maintaining data efficiency and organization.



**Note:** Column In Use feature does not detect columns in use from raw sql. This feature is targeted towards database tables only. Columns from CSV files will not be reported as in use.

2. Global level – The global level reporting flags the tables used across all the flowcharts.
  - Once any flowchart is run, the Licence utility (which inherently calls `unica_acsesutil.exe`), scans all the flowcharts' ses files which have run at a configured interval of time, and saves the table mappings that are 'in use' in the database table `UA_TableMappingXRef`.
  - Refer the section above **Loading stored table catalogs**, to load catalog.
  - Once the table mapping is loaded, a new column **In Use** indicates whether the table is used in any flowchart or not.
  - Details of flowcharts using a table catalog can be obtained using an API -

<http://<host>:<port>/Campaign/api/campaign/rest/v3/configuration/tables-usage-detail>

Sample input-

```
{
  "idFields": "ACCT_ID",
  "audience": "AutoALNum",
```

```
"userTableName": "Accounts",  
"type": "Base Record Table",  
"dataSource": "[Baseinfo] AUTODCC.ACCT"  
}
```

Sample output –

```
{  
  "content": [  
    {  
      "campaignId": 399,  
      "campaignName": "ACatCamp",  
      "sessionId": null,  
      "sessionName": null,  
      "flowchartId": 2877,  
      "flowchartName": "ACatFlow"  
    }  
  ],  
  "page": {  
    "pageNumber": 0,  
    "totalElements": 1,  
    "totalPages": 1,  
    "size": 20,  
    "hasNext": false,  
    "hasPrev": false
```

```
}
}
```



**Note:** Table Catalog In Use column : Does not mark table as used if it is not selected as input in any PB's but used only in extract fields.

## Setting up Unica Campaign to use database load utilities

You can improve performance by using a database load utility for all datasources.



**Note:** These instructions assume that you are using a DB2® database and MariaDB database on a supported operating system other than z/OS®. If you are using a different database, adjust the instructions accordingly. If you are using DB2® on z/OS®, see [Using database load utilities with DB2 on z/OS \(on page 107\)](#).

Unica Campaign supports the use of database load utilities, which are available from your database vendors. You must obtain licensed copies of any database load utilities.

Database load utilities can improve performance when pushing ID lists into temporary tables and when exporting data from Unica Campaign to the database. For example, data is exported during a Snapshot, Mail List, or Call List process.

Load utilities can provide significant performance improvements. Testing on DB2® indicates that without the load utility, insertion of a million rows requires about 5 times more CPU utilization and significant disk I/O. Results vary depending on the hardware in use.



**Important:** The following adjustments can impact system resources and potentially impact performance numbers.

To set up Unica Campaign to use a database load utility, there are three main steps to perform for each datasource: Create two load control file templates; create a script or executable to start the load utility; then set the loader configuration properties in Unica Campaign.

### 1. Create two load control file templates.

Most database load utilities require the use of a control file. Unica Campaign can generate control files dynamically based on control file templates that you create.

- a. Create a load control file template for adding records. The template must consist of the following lines. This example template is named `loadscript.db2`:

```
connect to <DATABASE> user <USER> using <PASSWORD>;
load client from <DATAFILE> of del modified by coldel| insert into <TABLE>(<FIELDNAME><,>
```

```
)
nonrecoverable;
```

- b. Create a load control file template for appending records. The template must consist of the following lines. This example template is named `loadappend.db2`:

```
connect to <DATABASE> user <USER> using <PASSWORD>;
load client from <DATAFILE> of del modified by coldel| insert into <TABLE>(
<FIELDNAME><,>
)
nonrecoverable;
```

You now have templates for loading data into a new or empty database table and appending data to an existing database table.

Unica Campaign will fill in the `DATABASE`, `USER`, `PASSWORD`, `DATAFILE`, `TABLE` and `FIELDNAME` tokens in the templates and create a configuration file called `CONTROLFILE` for the DB2® Load.

## 2. Create a script or executable to start the load utility.

To invoke the load utility, Unica Campaign uses a shell script (or an executable, for Windows), which is identified in the **Loadercommand** configuration property. You can specify either a direct call to the database load utility executable or a call to a script that launches the database load utility.

- a. For this example, create a shell script called `db2load.sh`, to start the loader. For the `/tmp` path, you can substitute the directory of your choice:

```
#!/bin/sh
cp $1 /tmp/controlfile.tmp
cp $2 /tmp/db2load.dat
db2 -tvf $1 >> /tmp/db2load.log
```

- b. Change permissions on the script file so it has execute permission:

```
chmod 755 db2load.sh
```

## 3. Set loader configuration properties in Unica Campaign.

The loader configuration properties identify the control file templates and indicate the location of the script or executable file. Be sure to adjust the configuration settings for each data source.

- a. Choose **Settings > Configuration**, then select `Campaign|partitions|partition1|dataSources|<datasourcename>`.



**Note:** When using `UA_SYSTEM_TABLES` (such as contact history) the bcp loader is invoked on any number of records.



But when using other data sources, it is invoked based on the value of the parameter "MinReqForLoaderCommand".

b. Set the properties that begin with the word **Loader**. For important information, see [Campaign | partitions | partition\[n\] | dataSources \(on page 379\)](#).

- **LoaderCommand:** The path to the script or executable to invoke the database load utility. The script must be in `CAMPAIGN_HOME/partition/partition[n]`. Most database load utilities require several arguments to be launched successfully. The tokens required by DB2® are shown in angle brackets in the following example. Type the tokens exactly as shown. They are replaced by the specified elements when the command is run. Example: `/HCL/Campaign/partition/partition1/db2load.sh <CONTROLFILE> <DATAFILE>`
- **LoaderCommandForAppend:** The path to the script or executable to invoke the database load utility for appending records to a database table. The script must be in `CAMPAIGN_HOME/partition/partition[n]`. Example: `/HCL/Campaign/partition/partition1/db2load.sh <CONTROLFILE> <DATAFILE>`
- **LoaderDelimiter** and **LoaderDelimiterForAppend:** The delimiter used in the loader control file template.
- **LoaderControlFileTemplate:** The control file template that is configured for Unica Campaign. Example: `loadscript.db2`
- **LoaderControlFileTemplateForAppend:** The control file template for appending records. Example: `loadappend.db2`
- All other **Loader** settings: Specify as required for your implementation, according to the information provided in the topic [Campaign | partitions | partition\[n\] | dataSources \(on page 379\)](#).

c. Perform this step if you are also using Unica Optimize.



**Note:** Unica Optimize does not update any *user* database datasources, so the following information does NOT apply to user database datasources.

Unica Optimize uses the UA\_SYSTEM\_TABLES datasource loader settings to update Unica Optimize tables during session runs. Because these settings are common to Unica Campaign and Unica Optimize, you must configure the loader as follows:

- Unica Optimize Loader settings: Avoid using relative paths for loader scripts in the UA\_SYSTEM\_TABLES datasource loader configuration. Instead, use absolute paths.
- If Unica Campaign and Unica Optimize are installed on different machines, create identical folder structures as per the absolute path on the Unica Campaign machine and the Unica Optimize. Make sure that the absolute path is accessible to both the Unica Campaign listener and the Unica Optimize listener from their respective machines.
- If Unica Campaign and Unica Optimize are installed on the same machine, you do not have to create the folder structure because it already exists.

**Example:**

In this example, Unica Campaign and Unica Optimize are installed on separate machines, and Unica Campaign has the following loader configuration:

**LoaderCommand:** /HCL/Campaign/partitions/partition1/db2load.sh <CONTROLFILE> <DATAFILE>

**LoaderCommandForAppend:** //HCL/Campaign/partitions/partition1/db2load.sh <CONTROLFILE>  
<DATAFILE>

In this example, you would create the directory /HCL/Campaign/partitions/partition1/ on the Unica Optimize machine and copy all necessary loader-specific script files to that directory on the Unica Optimize machine. For details, read about configuring the database load utility in the Unica Optimize User's Guide.

Unica Campaign performs the following actions when it writes to the database: First, it creates a temporary data file as fixed width or delimited text. If specified by the **LoaderControlFileTemplate** property, a temporary control file is created dynamically based on the template file and the list of fields to be sent to the database. Next, it issues the command specified by the **LoaderCommand** configuration property. Finally, it cleans up the temporary data file and control file.

## Amazon Redshift - Loader configurations

Steps to implement Copy Command/Loader utility for Amazon Redshift:

1. User should have Amazon Redshift database. Note down, Database Name, Port and DB user credentials.
2. Create S3 Bucket and make a note of the Bucket name
3. Need Access key ID and Secret access key to access S3 bucket.
4. Find ODBC URL, Database Host, Name and Port.
5. Configure Amazon Redshift in Campaign server. In case aws is not installed (in Linux). Please install `awscli.noarch` and `aws` configure packages.
6. Open <CAMPAIGN\_HOME>/bin/setenv.sh file and give path for `odbc.ini` file export  
`ODBCINI=<Path_for_odbc.ini_File>/odbc.ini`
7. In Campaign install location create a folder e.g. <CAMPAIGN\_HOME>/partitions/partition1/Redshift
8. Make sure below details are updated in 'RedShiftCopyCommand.sh' file

```
END_PT="<redshift-host-address>" #CHANGE ME
DB_NM="<redshift-db-name>" #CHANGE ME
DST_PATH="<s3-bucket-name>" #CHANGE ME
export AWS_ACCESS_KEY_ID="<keyid>" #CHANGE ME
export AWS_SECRET_ACCESS_KEY="<accesskey>" #CHANGE ME
export DSNNAME="<dsnname>" #CHANGE ME
```

9. Now copy 'RedShiftCopyCommand.sh' file in the <CAMPAIGN\_HOME>/partitions/partition1/Redshift folder.
10. Changes in Unica Application Configuration:

- Create Data source for Redshift database using 'PostgreSQLTemplate' (if not already created).
- In the Datasource provide below details.

```

LoaderCommand: <CAMPAIGN_HOME>/partitions/partition1/Redshift/RedShiftCopyCommand.sh <DATAFILE>
<TABLENAME> <DBUSER> <PASSWORD>

LoaderCommandForAppend:

<CAMPAIGN_HOME>/partitions/partition1/Redshift/RedShiftCopyCommand.sh <DATAFILE> <TABLENAME>
<DBUSER> <PASSWORD>

LoaderControlFileSpecifiesFields: TRUE

LoaderDelimiter: ,

LoaderDelimiterForAppend:

```

11. Restart application server and Campaign Listener.
12. Access application and verify of Copy/loader is working.
13. Once flowchart is completed successfully, verify the flowchart logs.
14. Make sure redshift loader is invoked.

Sample example logs:

```

[I] [TABLE ACC] [Snapshot1] USER (thread 0x7fefdb9a1700): Data retrieval completed; nnnnnn
records retrieved and returned to caller. [sdbtacc:439]

[I] [DB LOAD] [Snapshot1] Snapshot: INVOKING DATABASE-SPECIFIC LOADER:

[I] [DB LOAD] [Snapshot1]

<Campaign_home>/partitions/partition1/redshift/RedShiftCopyCommand.sh
<Campaign_home>/partitions/partition1/tmp/d829aaaq.t~# RedShiftDB_Copy

```

15. Verify logs for any error.

Sample example of RedShiftCopyCommand.sh loader utility script.

```

LOG_FILE="/tmp/log.$$"

cp $1 /tmp/

log () {
CMD=`basename $0`
DATE=`date '+%Y-%m-%d %T'`
echo "$DATE $1 $CMD $2" >> $LOG_FILE
}

END_PT="<Redshift_Server_Cluster_Name>"
DB_NM="<Database_name>"
DST_PATH="s3://s3bucketunica/" -> s3Bucket_name
export AWS_ACCESS_KEY_ID="XXXXXXXXXXXXXXXXXXXXXXXXX"

```

```

export AWS_SECRET_ACCESS_KEY="xXxXNXxXXNXxNX7xxxxNxXxNXXX/XXXXXXXXxxxxx"
export DSNNAME="<DSN_Name_Created>

#export AWS_DEFAULT_REGION=ap-northeast-1

PART_HOME=`dirname $0`
PART_HOME=${PART_HOME%/shell}

TABLE_NM=$2
USERNAME=$3
PASSWORD=$4
FILE_NM=`basename $1`
S3_FILE=$DST_PATH$FILE_NM

CMD_CPY="COPY "
CMD_FRM=" FROM '"
CMD_CRD="' CREDENTIALS 'aws_access_key_id="
CMD_SEC=";aws_secret_access_key="
CMD_END="' csv"

ERR_CD=1

log "INFO" "aws s3 copy command TABLE_NAME=""$2""
log "INFO" "aws s3 cp $1 $DST_PATH"

aws s3 cp $1 $DST_PATH >> $LOG_FILE 2>&1
RESULT=$?
if [ ${RESULT} -ne 0 ]; then
log "ERROR" "error executing s3 cp (${RESULT}) FILE_NAME=""$1" TABLE_NAME=""$2""
exit $ERR_CD
fi

COMMAND=${CMD_CPY}${TABLE_NM}${CMD_FRM}${S3_FILE}${CMD_CRD}${AWS_ACCESS_KEY_ID}${CMD_SEC}${AWS_SECRET_ACCESS_KEY}${CMD_END}
echo "$COMMAND" > /tmp/sql.$$
log "INFO" "$COMMAND"
isql $DSNNAME $USERNAME $PASSWORD -b < /tmp/sql.$$ > /tmp/isql.log.$$ 2>&1
cat /tmp/isql.log.$$ >> $LOG_FILE
errormsg="$(grep '\[ISQL\]ERROR' /tmp/isql.log.$$)"

if [ -n "$errormsg" ]
then
RESULT=1

```



```

fi
echo "RESULT is $RESULT"

if [ ${RESULT} -ne 0 ]; then
log "ERROR" "executing redshift copy (${RESULT}) FILE_NAME=\"$1\" TABLE_NAME=\"$2\""
exit $ERR_CD
fi

# remove file from s3?
aws s3 rm $S3_FILE
log "INFO" "aws s3 rm $S3_FILE"

log "INFO" "end script"

echo "LOG_FILE is $LOG_FILE"

exit 0

```

## MariaDB - Loader configurations

### MariaDB - Loader configurations

For Linux

MariaDB supports DB loader capabilities. See the following configurations.

### User load configurations

```

LoaderCommand = <Partition_HOME>/partition1/mysql.sh <CONTROLFILE>
LoaderCommandForAppend = <Partition_HOME>/partition1/mysql.sh <CONTROLFILE>
LoaderControlFileSpecifiesFields = FALSE
LoaderControlFileTemplate = load.ctr
LoaderControlFileTemplateForAppend = load_append.ctr

```

### Sample mysql.sh file

```

mysql -h 10.115.145.106 -u<UserName> -p<Password> <DB_Name> < $1
rc=$?
exit $rc

```

For Windows

MariaDB supports DB loader capabilities. See the following configurations.

## User load configurations

```
LoaderCommand = <Partition_HOME>/partition1/mysql.bat <CONTROLFILE>
<DSN> <DBUSER> <PASSWORD>
LoaderCommandForAppend = <Partition_HOME>/partition1/mysql.bat
<CONTROLFILE> <DSN> <DBUSER> <PASSWORD>
LoaderControlFileSpecifiesFields = TRUE
LoaderControlFileTemplate = load.ctr
LoaderControlFileTemplateForAppend = load_append.ctr
```

## Sample mysql.bat file

```
"C:\Program Files\MariaDB 10.6\bin\mysql.exe" -h localhost -u"%3" -p"%4" %2 < %1
exit %errorlevel%
```

## Sample load.ctr file

```
load data local infile '<DATAFILE>'
replace into table <TABLE>
fields terminated by '|'
OPTIONALLY ENCLOSED BY ''
```

## Sample load\_append.ctr file

```
load data local infile '<DATAFILE>'
IGNORE into table <TABLE>
fields terminated by '|'
OPTIONALLY ENCLOSED BY ''
```

## Fast loader repeated tokens

When you create a `LoaderControlFileTemplate` or a `LoaderControlFileTemplateForAppend`, a list of special tokens is repeated once for each field in the outbound table.

The available tokens are described in the following table.

**Table 16. Fast loader repeated tokens**

Token	Description
<CONTROLFILE>	This token is replaced with the full path and file name to the temporary control file that Unica Campaign gen-

**Table 16. Fast loader repeated tokens (continued)**

	erates according to the template that is specified in the <code>LoaderControlFileTemplate</code> parameter.
<DSN>	This token is replaced with the value of the <code>DSN</code> property. If the <code>DSN</code> property is not set, the <DSN> token is replaced by the data source name used in the category name for this data source (the same value used to replace the <DATABASE> token).
<DATABASE>	This token is replaced with the name of the data source that Unica Campaign is loading data into. This is the same data source name used in the category name for this data source.
<DATAFILE>	This token is replaced with the full path and file name to the temporary data file created by Unica Campaign during the loading process. This file is in the Unica Campaign Temp directory, <code>UNICA_ACTMPDIR</code> .
<NUMFIELDS>	This token is replaced with the number of fields in the table.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<TABLE>	This token is obsolete, but is supported for backward compatibility. See <TABLENAME>, which replaced <TABLE> as of version 4.6.3
<TABLENAME>	This token is replaced with the database table name that Unica Campaign is loading data into. This is the target table from your Snapshot process or the name of the Temp Table being created by Unica Campaign.
<USER>	This token is replaced with the database user from the current flowchart connection to the data source.

## Google BigQuery - Loader Configuration

For GBQ we have tested and supported using “bq” command line tool which is available when Google Cloud SDK is installed (Version 267.0.0), using below link you can download this.

<https://cloud.google.com/sdk/docs/#linux>

Make sure that your system has Python 2 with a release number of Python 2.7.9 or higher.

**Sample Loader script:**

```

export PATH=<PYTHON2.7_HOME>/bin:$PATH

#For Example

#export PATH=/usr/local/python2.7/bin:$PATH

"<GOOGLE_CLOUD_SDK_HOME>/bin/bq" show --schema "<PROJECT_NAME>:<DATASET_NAME>.$1" >
<CAMPAIGN_HOME>/partitions/partition1/tmp/schema.csv.$$

"<GOOGLE_CLOUD_SDK_HOME>/bin/bq" load --field_delimiter="|" " <DATASET>.$1" "$2"
<CAMPAIGN_HOME>/partitions/partition1/tmp/schema.csv.$$

exit 0

#Replace <CAMPAIGN_HOME> , <GOOGLE_CLOUD_SDK_HOME> , <PROJECT_NAME> , <DATASET_HOME> with the actual values

```



**Note:** BigQuery has lot many syntax and can be configured in different ways to load data In GBQ, we have configured and tested as above but this can be modified and used according to your need.

Name above loader script as bqload.sh (or anything relevant) and copy into <CAMPAIGN\_HOME>/partitions/partition1 directory.

**Table 17. Google BigQuery user data source configuration for loader**

LoaderCommand	<CAMPAIGN_HOME>/partitions/partition1/bqload.sh <TABLE> <DATAFILE>
LoaderCommandForAppend	<CAMPAIGN_HOME>/partitions/partition1/bqload.sh <TABLE> <DATAFILE>
LoaderControlFileSpecifiesFields	FALSE
LoaderControlFileTemplate	
LoaderControlFileTemplateForAppend	
LoaderDelimiter	
LoaderDelimiterAtEnd	FALSE
LoaderDelimiterAtEndForAppend	FALSE
LoaderDelimiterForAppend	
LoaderNULLValueInDelimitedData	
LoaderUseLocaleDP	FALSE

## PostgreSQL - Loader Configuration

PostgreSQL UserDB datasource configuration

LoaderCommand	<CAMPAIGN_HOME>/partitions/partition1/load.sh <PASSWORD> <TABLENAME> <DATAFILE>
LoaderCommandForAppend	<CAMPAIGN_HOME>/partitions/partition1/load.sh <PASSWORD> <TABLENAME> <DATAFILE>
LoaderDelimiter	,
LoaderDelimiterForAppend	,

PostgreSQL Loader script :

```
[hcluser@COMP-5059-1 partition1]$ cat load.sh

export PGPASSWORD="$1"

cp $3 /tmp/

/usr/pgsql-14/bin/psql -h <server hostname> -U postgres -d <Database_Name> -c "\COPY $2 FROM '$3' DELIMITER ','
CSV HEADER;"

exit $?
```

## OneDB - Loader configurations

You must use an OneDB database on the supported operating system. If you are using a different database, adjust the instructions accordingly.

You can use "dbload" (IDS utility) as a load utility. This utility is only available with IDS installation and not with ICSDK installation. So to achieve the load utility functionality, the preferable tool is "dbaccess".

The following procedure describes the dbaccess tool for load utility.

1. Create load control file template.
2. Create a load control file template for adding records. The template must consist of the following lines.

Sample template name : `informixLoad.ctr`

```
CONNECT TO 'hard_coded_db_name' USER '<USER>' USING '<PASSWORD>';
LOAD FROM '<DATAFILE>' DELIMITER '|' INSERT INTO <TABLENAME>;
```



**Note:** Tokens mentioned in the sample file should be in quotes. The DB name on which the loader utility is supposed to run, must be hard-coded, as mentioned in the example.

3. Create a script or executable to start the load utility. To invoke the load utility, Campaign uses a shell script for Unix or executable for Windows, which is identified in the `Loadercommand` configuration property. You can either specify a direct call to the database load utility executable or a call to a script that launches the database load utility.

### Sample shell script for Linux / AIX OS

```
informixLoad.sh: #!/bin/sh cp
$1 /opt/Camp_Informix1/Campaign/partitions/partition1/tmp/controlfile.sql dbaccess
- /opt/Camp_Informix1/Campaign/partitions/partition1/tmp/controlfile.sql
```

### Sample executable file for Windows

Case 1 : If OneDB server and OneDB client SDK are installed on same machine.

#### informixLoad.bat

```
set CTRL_FILE=%1
copy /Y "%CTRL_FILE%" "controlfile.sql"

set INFORMIXDIR=<Informix Directory>
set REGMACHINE=\\<Server Hostname>
set INFORMIXSERVER=<Server-name>
set ONCONFIG=onconfig.<server-name>
set INFORMIXSQLHOSTS=C:\Program Files\IBM Informix Software Bundle\etc\sqlhosts.<server-name>
set GL_USEGLU=1
set PATH=%INFORMIXDIR%\bin;%PATH%

dbaccess - controlfile.sql
```

If OneDB server and OneDB client SDK are installed on different machines.

#### informixLoad.bat

```
set CTRL_FILE=%1
copy /Y "%CTRL_FILE%" "controlfile.sql"
    set INFORMIXDIR=<CSDK installation path>
    set INFORMIXSERVER=<IDS server name>
    set INFORMIXSQLHOSTS=Path to sqlhosts file which is <CSDK_HOME>/etc/sqlhosts
    set GL_USEGLU=1
    set PATH=%INFORMIXDIR%\bin;%PATH%

dbaccess - controlfile.sql
```



**Note:** If the setup is Unicode, add the following additional lines before `dbaccess` command.

```
set LANG=en_US.utf8
    set DB_LOCALE=en_US.utf8
```



```
set CLIENT_LOCALE=en_US.utf8
set SERVER_LOCALE=en_US.utf8
```

In Case 2, the following are the other settings required at CSDK side:

- Open `C:\Windows\System32\drivers\etc\services` file and add a new entry for IDS like `IDS serviceName` which is a service name in IDS machine on which IDS is running and `IDS port number` which is a port number on IDS machine on which IDS is listening. For example: `<service-name>9091/tcp`
- If required, make the following changes.
  - Open the copied `C:\Program File\Informix Client-SDK\etc\sqlhosts.%informixserver%` file.
  - Copy the existing line and replace the machine name with its IP address.
- Set the loader configuration properties in Campaign.
- Select **Settings > Configuration** and then select **Campaign|partitions|partition1|dataSources|<datasourcename>**.
  - **LoaderCommand**: The path to the script or executable to invoke the database load utility. The script must be `CAMPAIGN_HOME/partition/partition[n]`.  
 Sample value: `<CAMPAIGN_HOME>\partitions\partition1\informixLoad.bat <CONTROLFILE>`
  - **LoaderControlFileTemplate**: The control file template configured for Campaign. This file must also be in `CAMPAIGN_HOME/partition/partition[n]` format.  
 Sample value: `<CAMPAIGN_HOME>\partitions\partition1\informixLoad.ctr`
  - **LoaderDelimiter**: The delimiter used in the loader control file template.  
 Sample value: `|`
  - **LoaderDelimiterAtEnd**: It specifies whether to put delimiter after last column value in control file or not.  
 Sample value: `True`. For Informix, this must be `true`.

## Action Db - Loader configurations

**vwload** is the tool provided by Actian Vector for load / bulk insert task.

CASE 1: Actian DB server and client running on same machine:

- Changes in **Settings > Configurations > Campaign > partitions > partition1 > dataSources > [actian db data source name]**
  - **LoaderDelimiter** > `|`
  - **LoaderCommand** > `path\to\batchFile.bat <DBUSER> <TABLENAME> <DATAFILE>`. Contents of batchFile.bat file are:

- set DB\_USER\_NAME=%1
- set TABLE\_NAME=%2
- set DATA\_FILE=%3
- set DB\_NAME=db\_name
- vwload -u %DB\_USER\_NAME% -t %TABLE\_NAME% %DB\_NAME% %DATA\_FILE%

OR if its shellScriptFile.sh :

- DB\_USER\_NAME=\$1
- TABLE\_NAME=\$2
- DATA\_FILE=\$3
- DB\_NAME=db\_name
- vwload -u \$DB\_USER\_NAME -t \$TABLE\_NAME \$DB\_NAME \$DATA\_FILE

Where **db\_name** must be replaced with the actual action database name on which this loader command supposed to be execute.

CASE 2: Action DB server and client are running on different machine:

Suppose Action DB server is running on **Machine-1** and Action client running on **Machine -2**.

- Changes in **Settings > Configurations > Campaign > partitions > partition1 > dataSources > [action db data source name]**
  - **LoaderDelimiter > |**
  - **LoaderCommand > path\to\batchFile.bat <DBUSER> <PASSWORD> <TABLENAME> <DATAFILE>.**

Contents of batchFile.bat file are:

- set DB\_USER\_NAME=%1
- set DB\_PASSWORD=%2
- set TABLE\_NAME=%3
- set DATA\_FILE=%4
- set DB\_SERVER\_MACHINE\_NAME=Machine-1
- set DB\_SERVER\_COMMUNICATION\_PROTOCOL=tcp\_ip
- set DB\_SERVER\_INSTANCE\_NAME=VW
- set DB\_NAME=db\_name
- set VNODE\_STRING= "@%DB\_SERVER\_MACHINE\_NAME%, %DB\_SERVER\_COMMUNICATION\_PROTOCOL%,%DB\_SERVER\_INSTANCE\_NAME%; [%DB\_USER\_NAME%,%DB\_PASSWORD%]::%DB\_NAME%"
- vwload -t %TABLE\_NAME% %VNODE\_STRING% %DATA\_FILE%

OR if its shellScriptFile.sh



- DB\_USER\_NAME=\$1
- DB\_PASSWORD=\$2
- TABLE\_NAME=\$3
- DATA\_FILE=\$4
- DB\_SERVER\_MACHINE\_NAME=Machine-1
- DB\_SERVER\_COMMUNICATION\_PROTOCOL=tcp\_ip
- DB\_SERVER\_INSTANCE\_NAME=VW
- DB\_NAME=db\_name
- VNODE\_STRING= "@\${DB\_SERVER\_MACHINE\_NAME},  
\${DB\_SERVER\_COMMUNICATION\_PROTOCOL},\${DB\_SERVER\_INSTANCE\_NAME};  
[\${DB\_USER\_NAME},\${DB\_PASSWORD}]::\${DB\_NAME}"
- vload -t \$TABLE\_NAME \$VNODE\_STRING \$DATA\_FILE

Where **db\_name** must be replaced with the actual action database name on which this loader command supposed to be execute. Machine-1 is the name of machine name entry done in /etc/hosts. tcp\_ip is hardcoded communication protocol. This should not change. VW is the running instance name of db server running on Machine-1. Please check and update this accordingly. Please don't do any changes in VNODE\_STRING.

## Fast loader repeated tokens

When you create a `LoaderControlFileTemplate` or a `LoaderControlFileTemplateForAppend`, a list of special tokens is repeated once for each field in the outbound table.

The available tokens are described in the following table.

**Table 18. Fast loader repeated tokens**

Token	Description
<CONTROLFILE>	This token is replaced with the full path and file name to the temporary control file that Unica Campaign generates according to the template that is specified in the <code>LoaderControlFileTemplate</code> parameter.
<DSN>	This token is replaced with the value of the <code>DSN</code> property. If the <code>DSN</code> property is not set, the <code>&lt;DSN&gt;</code> token is replaced by the data source name used in the category name for this data source (the same value used to replace the <code>&lt;DATABASE&gt;</code> token).
<DATABASE>	This token is replaced with the name of the data source that Unica Campaign is loading data into. This is the same data source name used in the category name for this data source.

**Table 18. Fast loader repeated tokens (continued)**

Token	Description
<DATAFILE>	This token is replaced with the full path and file name to the temporary data file created by Unica Campaign during the loading process. This file is in the Unica Campaign Temp directory, UNICA_ACTMPDIR.
<NUMFIELDS>	This token is replaced with the number of fields in the table.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<TABLE>	This token is obsolete, but is supported for backward compatibility. See <TABLENAME>, which replaced <TABLE> as of version 4.6.3.
<TABLENAME>	This token is replaced with the database table name that Unica Campaign is loading data into. This is the target table from your Snapshot process or the name of the Temp Table being created by Unica Campaign.
<USER>	This token is replaced with the database user from the current flowchart connection to the data source.

In addition to these special tokens, every line includes other characters. To include a single character on every line except the last line, the character can be enclosed within angle brackets. You can enclose only a single character between the angle bracket (< >) characters for this functionality.

This is commonly used to separate the list of fields with a comma. For example, the following syntax generates a comma-separated list of field names:

```
<FIELDNAME>< , >
```

The angle bracket (< >) characters around the comma indicate that the comma should be present on every line, after every inserted field name, except the last.

If any sequence of characters does not fit this requirement, it is repeated every time, including the last. So for example, to generate a parenthesized, comma-separated list of field names in which each field name is preceded by a colon, you can use the following syntax:

```
(
  : <FIELDNAME>< , >
)
```

Because the colon is not enclosed within angle bracket (< >) characters, it repeats for every line; however, the comma appears on every line except the last. It might produce output as follows:

```
(
:FirstName,
:LastName,
:Address,
:City,
:State,
:ZIP
)
```

Note that the comma does not appear after the last field name (ZIP), yet the colon appears before every field name.

## Using database load utilities with DB2 on z/OS

A database load utility can improve Unica Campaign performance. Follow these steps to configure Campaign to use a database load utility for a DB2® user database on z/OS®.

This procedure applies specifically to DB2® on z/OS®. If you are using DB2® on a different operating system, see [Setting up Unica Campaign to use database load utilities \(on page 91\)](#).

1. Set up the z/OS® UNIX™ System Services (USS) pipes on z/OS®.
2. Write a stored procedure to invoke the DSNUUTILU and a script to invoke the stored procedure.
3. Go to `Campaign|partitions|partition1|dataSources|<datasourcename>` and set the properties that begin with the word **Loader**.



**Note:** `LoaderControlFileTemplate` and `LoaderControlFileTemplateForAppend` are not used for DB2® on z/OS®.

4. Go to `Campaign|partitions|partition1|dataSources|<datasourcename>` and set both **DB2NotLoggedInitially** and **DB2NotLoggedInitiallyUserTables** to FALSE.

## Troubleshooting database load utilities for Unica Campaign

Some known issues with database loader utilities are listed below, along with workarounds or solutions.

### Timeout and locking problems: DB2® database load utility

Use the following suggestions to help troubleshoot timeout and locking problems that might occur when you use a DB2® database load utility with Unica Campaign.

Multiple flowcharts are running concurrently and the flowcharts write to the same table. A flowchart run fails with the following errors:

- Unica Campaign UI: "Loader command exited with an error status 4" and
- Loader logs: "SQL0911N The current transaction has been rolled back because of a deadlock or timeout".

For example, you are using multiple flowcharts to insert records to the `UA_ContactHistory` table using the Mail List process box.

The load utility does not support loading data at the hierarchy level. When you concurrently run multiple flowcharts that load data into the same table, each individual load process locks the table. Each load process has to wait until the previous load finishes. If a process takes a longer time to finish, the next load process in the queue times out and produces the errors listed above.

Table locking during load operations: In most cases, the load utility uses table level locking to restrict access to tables. The level of locking depends on the stage of the load operation and whether it was specified to allow read access.

A load operation in `ALLOW NO ACCESS` mode uses a super exclusive lock (Z-lock) on the table for the duration of the load. Before a load operation in `ALLOW READ ACCESS` mode begins, the load utility waits for all applications that began before the load operation to release their locks on the target table. At the beginning of the load operation, the load utility acquires an update lock (U-lock) on the table. It holds this lock until the data is being committed. When the load utility acquires the U-lock on the table, it waits for all applications that hold locks on the table prior to the start of the load operation to release them, even if they have compatible locks. This is achieved by temporarily upgrading the U-lock to a Z-lock, which does not conflict with new table lock requests on the target table as long as the requested locks are compatible with the load operation's U-lock. When data is being committed, the load utility upgrades the lock to a Z-lock, so there can be some delay in commit time while the load utility waits for applications with conflicting locks to finish.



**Note:**

- `MinReqForLoaderCommand` and `MinReqForLoaderCommandForAppend` is not applicable for contact history table, detailed contact history table, and response history table.
- The load operation can time out while it waits for the applications to release their locks on the table prior to loading. However, the load operation does not time out while waiting for the Z-lock needed to commit the data.

Workaround: Unica Campaign uses a shell script (or an executable, for Windows™) specified in the **Loadercommand** configuration property to invoke the database load utility. You can add queuing logic to your shell script or executable to work around this issue. This logic checks to see if one loader is running and performing load operation on a table; if this is the case, it does not allow other loaders to start loading until the previous one completes.

## "Check pending" problems: DB2® database load utility

"Check pending" problems can occur when using a DB2® database load utility with Unica Campaign. Use the following suggestions to help troubleshoot these types of issues.

You get a SQL0668N error.

If a database loader is used to insert records in a table and the table has referential constraints, the table will be left in a "check pending" state after the load operation. Referential constraints include unique constraints, range constraints for partitioned tables, generated columns, and LBAC security rules. When a table is in this state, you get a SQL0668N error if any select query is performed on the table.

To get the table out of a "check pending" state, run the following command:

```
SET INTEGRITY FOR TABLE <TABLENAME> IMMEDIATE CHECKED
```

The following code can be used in the script:

```
load client from <DATAFILE> of del modified by coldel| insert into <TABLE>(
  <FIELDNAME><,>
)
nonrecoverable;
set integrity for <TABLE> immediate checked;
```

## Archiving campaigns and flowcharts

The Unica Campaign application does not provide an automated way to archive retired marketing campaigns or flowcharts. However, you can back up the required files, then use the Unica Campaign user interface to delete unneeded campaigns and flowcharts.


There are several tables in the Unica Campaign system database that contain data about the status of flowcharts. However, those tables do not provide complete information to determine if Unica Campaign projects or flowcharts can be archived and purged.

To develop an archiving solution that is tailored to your needs, consult with Professional Services. If that is not possible, you can perform the steps below.


The following procedure is a manual process, but it will keep the system clean and will remove all associated components in the file system and the system tables.

1. Use the following information to determine if the flowchart can be archived:
  - Examine the log files for each flowchart and flowchart session to determine the last run/modified date and time stamp.
  - Check to see whether there are any triggers related to the campaign that will execute the campaign depending on some activity.
  - Check to see whether there are any schedules related to the campaign flowchart. In the case of a response flowchart, confirm that the time for taking responders into account has passed.
2. When a determination is made to archive certain campaigns and flowcharts, take a snapshot of the database and the Unica Campaign file directory structure at `Campaign/partitions/partition[n]`. You do not need to back up the `tmp` folder within `Campaign/partitions/partition[n]`.

If you have clustered listeners, also back up all of the files and folders in the location specified by `Campaign | campaignClustering | campaignSharedHome`.

 **Important:** It is extremely important that you complete BOTH a file directory backup AND a database snapshot at the exact same time. Unica Campaign renders the GUI based on the database, but there must also be an OS object for the associated database object. For best results, contact Professional Services before attempting a backup.

3. Use the Unica Campaign user interface to manage campaigns and flowcharts in folders. The guidelines below use six months and twelve months as an example. Depending on your business rules and legal requirements, your timelines may vary:
  - a. Create an archive folder, then create month subfolders within it.
  - b. At six months, move the campaigns and flowcharts to a month subfolder in the archive directory.
  - c. At twelve months, delete the month folder and all campaigns in it.

 **Important:** To maintain the integrity of the file system, and because tables have entity relationships, we recommend that you use the Unica Campaign user interface to delete campaigns and flowcharts.

The key to restoring objects (as is the key to archiving) is that Unica Campaign needs both a database entry and a file on the OS to create valid objects in Unica Campaign. Professional Services can assist with a backup and recovery strategy.

# Chapter 4. Customizing campaigns

Administrators can customize campaigns by using custom campaign attributes, initiatives, and products.

## Custom campaign attributes

You can customize campaigns by adding custom campaign attributes to store metadata about each campaign.



**Note:** If your Unica Campaign installation is integrated with Unica Plan, you must use Unica Plan to create custom campaign attributes. For details, see the Unica Plan documentation.

Custom attributes can help you further define and categorize your campaigns. For example, you might define a custom campaign attribute `Department` to store the name of the department in your organization that is sponsoring the campaign. The custom attributes you define appear on the **Summary** tab for each campaign.

Custom campaign attributes apply to all campaigns in your system. If you add a custom campaign attribute when there are existing campaigns, the value of the attribute is NULL for these campaigns. You can later edit these campaigns to supply a value for the custom attribute.



**Note:** Names of custom attributes must be unique across campaign, offer, and cell custom attributes.

## Custom cell attributes

You can create customized cell attributes. For example, you can define a custom cell attribute of `Marketing Approach` to store values such as `Cross-sell`, `Up-sell`, `Defection`, or `Loyalty`. Custom cell attributes are included in the target cell spreadsheet (TCS) for every campaign, even campaigns that have already been created.

Custom cell attributes are the same across all campaigns. Users enter values for custom cell attributes in the target cell spreadsheet of a campaign. For example, if you created the custom cell attribute `Marketing Approach`, users would see a `Marketing Approach` field when editing a row in the target cell spreadsheet.

Output processes in flowcharts can also generate output values for custom cell attributes as Unica Campaign Generated Fields (UCGF). Users can then view reports based on the values of the cell attributes, if the reports are customized to support this. For more information, see the Unica Campaign User's Guide.



**Note:** If Unica Campaign is integrated with Unica Plan, you must use Unica Plan to create custom cell attributes. For details, see the Unica Plan documentation.

## Custom offer attributes

Unica Campaign is delivered with a standard set of offer attributes for use in offer templates. You can create custom offer attributes to store additional offer metadata for definition, output, or analysis.

For example, you might define a custom offer attribute Interest Rate to store the value of the interest rate being marketed with mortgage offers.

When defining offer templates, you can choose which standard and custom offer attributes will be visible for a particular type of offer. Users then provide values for those attributes when creating and/or using offers.

You can use a custom attribute in an offer template in one of three ways:

- As a static attribute
- As a hidden static attribute
- As a parameterized attribute

## What is a static attribute?

A static attribute is an offer field whose value is set once and does not change when the offer is used.

When you create an offer template, you supply values for all static attributes. When a user creates an offer based on that template, the values you entered are used by default; the user can override these default values as needed. However, users cannot override values for static attributes when using the offer in a flowchart process.

There are static attributes that are automatically included in all offer templates.

In both offer template and offer, static attributes will not be mandatory by default. Instead, the behaviour is controlled by the "isMandatory" flag. If the Static attribute is marked "Mandatory", then user must provide value in the offer template and offer. If the Static attribute is marked "Non-Mandatory" then user can leave the field blank and create offer template as well as offers.

## What is a hidden static attribute?

A hidden static attribute is an offer field that is not displayed to users when they create offers based on this template. For example, a hidden static attribute could be the cost to your organization of administering an offer.

Users creating offers cannot edit (or see) the value of hidden static attributes. However, you can track and generate reports on the values of hidden static attributes in the same way as you could other offer attributes.

When you are creating an offer template, the value you enter for a hidden static attribute applies to all offers based on the template.

## What is a parameterized attribute?

A parameterized attribute is a field that users can change for every instance in which the offer is associated with a cell in a flowchart.


When you create an offer template, you supply a default value for the parameterized attribute. Then, when users create offers based on this template, they can accept or change the default value you entered. Finally, when the offer containing the parameterized attribute is associated with a cell in a flowchart, the user can accept or change the default value entered for the offer.





## Creating or editing custom attributes


You can define custom attributes for use on campaigns, offers, or cells on a target cell spreadsheet. When you create the attribute, you specify whether it can be used in campaigns, offers, or cells. You cannot change this selection after you save the attribute.

You must have permissions to add or modify campaign, offer, and cell attributes.



1. Select **Settings > Campaign settings**.
2. In the **Templates and customization** section, click **Custom attribute definitions**.
3. Click the **Add a custom attribute** icon  or click the name of the attribute that you want to modify.
4. Define the attribute.



For this option:	Take this action:
<p><b>Attribute display name</b></p>	<p>Specify the label to identify the attribute for users. For example, "Interest Rate". Double quotes in attribute display names are not supported in target cell spreadsheets. The target cell spreadsheet escapes special decoration for attribute display names. For example, instead of a column name in the target cell spreadsheet appearing in bold red text, the column name would appear exactly as shown here: <code>&lt;strong style=\"color: red;\"&gt;Name&lt;/strong&gt;</code>.</p> <p> <b>Note:</b> You cannot change the name of the standard offer attributes that are supplied with Unica Campaign.</p>
<p><b>Internal name</b></p>	<p>Specify a name to identify this attribute when you write an Unica expression, such as a query or custom macro. Use the same name as the <b>Attribute display name</b>, but without any spaces (for example, "Interest-Rate").</p> <p>The internal name must be globally unique, must start with an alphabetic character, cannot include spaces, and are not case-sensitive.</p> <p>To avoid errors, do not change the internal name of attributes that are used in flowcharts.</p>

For this option:	Take this action:
<p><b>Attribute of</b></p>	<p>Indicate where the attribute can be used. You cannot change this option after you save the attribute.</p> <ul style="list-style-type: none"> <li>◦ A <b>Campaign</b> attribute is included on every campaign, even campaigns that already exist.</li> <li>◦ An <b>Offer</b> attribute is available for use on new offer templates. If the attribute is included on an offer template, then any offers that are based on that template include the attribute.</li> <li>◦ A <b>Cell</b> attribute is included in the target cell spreadsheet for every campaign, even campaigns that already exist.</li> </ul>
<p><b>Description</b></p>	<p>Optional.</p>
<p><b>Mandatory</b></p>	<p>If you want to require a value for this attribute, select <b>Mandatory</b>. This setting has the following result.</p> <ul style="list-style-type: none"> <li>◦ For campaigns, users are required to specify a value for the attribute (the field cannot be left blank).</li> <li>◦ For cells, users are required to specify a value in the target cell spreadsheet (the cell cannot be left blank).</li> <li>◦ For offers, administrators are required to specify a value when the attribute is added to an offer template. The specified value is used for any offers that are based on that template, unless the user specifies a different value when creating or editing the offer.</li> </ul> <p> <b>Note:</b> If you add an offer attribute to an offer template as Static, "Hidden static" or "Parameterized", a value is always required, even if the attribute is defined as non-mandatory.</p> <p>If you change this option after an attribute is in use:</p>

For this option:	Take this action:
	<ul style="list-style-type: none"> <li>◦ If you change from mandatory to non-mandatory, a value is no longer required when the attribute is used.</li> <li>◦ If you change from non-mandatory to mandatory, a value is required whenever this attribute is used in the future. This change has no effect on existing objects unless you edit them. For example, if you open a campaign, target cell spreadsheet, or offer in Edit mode, you are required to specify a value before saving.</li> </ul>
<b>Form element type</b>	<p>Specify the type of data to be stored in the attribute field for offers or cells.</p> <p> <b>Important:</b> After you add the custom attribute, you cannot change its data type.</p>

5. Specify more information, depending on the Form Element Type that you selected.

If you selected this Form element type:	Take this action:
<b>Text field - Numeric</b>	<p>Specify the number of decimal places to display to the right of the decimal point.</p> <p> <b>Note:</b> If you decrease the value for an existing attribute, the display is truncated in the user interface. However, the original value is retained in the database.</p>
<b>Text field - Currency</b>	<p>Specify the number of decimal places (see above).</p> <p> <b>Important:</b> The currency value reflects the number of decimal places that are customarily used in the local currency. If you specify a number of decimal places less than the number commonly used, the currency value is truncated.</p>

If you selected this Form element type:	Take this action:
<b>Text field - String</b>	<p>Specify a <b>Maximum string length</b> to indicate the maximum number of bytes to store for any value of this attribute. For example, if you enter 32, single-byte languages such as English store 32 characters but double-byte languages store only 16 characters.</p> <p> <b>Important:</b> Decreasing the length of an existing attribute truncates existing values, which, if the field is used for matching purposes, can adversely affect response tracking.</p>
<b>Select box - String</b>	<ul style="list-style-type: none"> <li>◦ Specify a <b>Maximum string length</b> (see above).</li> <li>◦ Optionally, check <b>Allow addition of list items from within edit forms</b> to allow any user to add new unique values to the list of available values when creating or editing a campaign, offer template, or offer that includes this attribute. (This option does not apply to cells.) For example, if a Select Box on an offer template contains the values Small, Medium, Large, a user could add the value Extra Large when creating an offer or editing the offer template.</li> </ul> <p> <b>Important:</b> After the campaign, offer template, or offer is saved, the user cannot remove the new list item. The value is saved in the custom attribute definition and is then available to all users. Only administrators can remove items from lists, by modifying the custom attribute.</p> <ul style="list-style-type: none"> <li>◦ Populate the <b>Source list of available values</b> to specify which items are available in the Select Box. Enter values in the <b>New or selected item</b> field and click <b>Accept</b>. To delete a val-</li> </ul>

If you selected this Form element type:	Take this action:
	<p>ue, select it in the Source list of available values and click <b>Remove</b>.</p> <ul style="list-style-type: none"> <li>◦ Optionally, specify a <b>Default value</b> for the Select Box. The default value is used on the campaign, offer, or target cell spreadsheet unless the user specifies a different value when creating or editing the campaign, offer, or cell.</li> <li>◦ Specify the <b>Sort order</b> to determine how values appear in the list.</li> </ul>

6. Click **Save changes**.

## Defining corporate initiatives for marketing campaigns

Unica Campaign is delivered with a built-in attribute called **Initiative**. The **Initiative** attribute is a drop-down list on the Unica campaign **Summary** tab. Initially, there are no values in the list; as an administrator, you define the initiatives for users to select.

Follow the procedure below to define the values that users can select from the **Initiative** list in the Unica campaign **Summary** tab. You add initiatives directly in the database table `UA_Initiativies`.

When users create a marketing campaign, they will select an initiative from the list that you define.

1. Access the Unica Campaign system table database using your database management system.
2. In the database table `UA_Initiativies`, add values in the `InitiativeName` column. Each value can be a maximum of 255 characters.
3. Save the changes to the `UA_Initiativies` table.

## Adding products

You can add products that users can associate with offers. You add products directly in the database table `UA_Product`.

Users can associate an offer with one or more relevant products. Product IDs are stored in the `UA_Product` table in the Unica Campaign system table database. Initially, there are no records in this table. As an administrator, you can populate this table.

1. Use your database management system to access the Unica Campaign system table database.
2. Locate the `UA_Product` table.

The table is delivered with two columns:

- `ProductID` (`bigint`, length 8)
- `UserDefinedFields` (`int`, length 4)

3. Optionally, modify the table to include additional columns. You may also delete the `UserDefinedFields` column.
4. Populate the table as necessary to include products that can be associated with offers.
5. Save the changes to the `UA_Product` table.

Users can now assign relevant products when creating or editing an offer.

# Chapter 5. Offer template administration

Before administrators perform offer template management tasks, there are several important concepts to understand.

Offers are always based on offer templates. Offer templates include standard attributes, such as Offer name and Channel. Administrators can create custom attributes and add them to offer templates. Any offers that are based on that template will include the custom attribute.

An example of a custom attribute is a drop-down list of Interest Rates, which users can select from when they create an offer.

## What is an offer?

An offer is the specific marketing communication that you send to particular groups of people, using one or more channels. An offer can be simple or complex, and typically contains a creative piece, a cost, a channel, and an expiration date.

For example, a simple offer from an online retailer could consist of free shipping on all online purchases made in the month of April. A more complex offer could consist of a credit card from a financial institution, with a personalized combination of artwork, introductory rate, and expiration date that varied based on the recipient's credit rating and region.

In Unica Campaign, offers are:

- Based on the offer templates that you administer.
- Used in campaigns, where they are associated with target cells.

Associated offers are then delivered to customers identified in these target cells.

You can also group offers into lists, and assign offer lists to target cells.



**Note:** Offer names and offer list names have specific character restrictions. For details, see [Special characters in Unica Campaign object names \(on page 542\)](#).

## What is an offer template?

An offer template defines the structure of an offer. Users create offers by basing them on offer templates.



**Important:** Offer templates are required. Users cannot create an offer without basing it on a template.

You can create as many offer templates as appropriate to manage the different types of offers within your business. When defining an offer template, you specify the relevant offer attributes and how they will be used.

Offer templates provide the following benefits:

- By creating offer templates, you simplify offer creation for users because only the relevant offer attributes for a particular type of offer are shown.
- By providing default values for offer attributes, you speed up the offer creation process.
- By specifying which offer attributes are parameterized in an offer template, you control when new offers are created and when an offer version can be used instead.
- By using custom attributes to capture specific data (for example, the discount percentage or bonus points associated with the offer), you enable better reporting and analysis of campaigns.

## Offer templates and security

The security policy you set for an offer template determines which users will be able to use the offer template.

The security policy of the offer template is independent of the security policy applied to the offers created by using this offer template; that is, the security policy is not propagated to offers based on the template.

When users create new offers, the security policy of the offer is based on the folder in which it resides. If the folder is created in the top-level offers folder, the user can select other valid security policies for that offer.

To work with offer templates, which includes tasks such as adding, editing, or retiring offer templates, you must have the appropriate permissions, which includes the view permission for offer templates. For example, to add offer templates, you must be granted both the **Add offer templates** and **View offer templates** permissions.

For more information about Unica Campaign security, see the Unica Platform Administrator's Guide.

## Planning offer templates and offers

When you plan offers, you should consider which template to use, which attributes are parameterized, and whether you will use holdout control groups for cells assigned this offer.

Offers can vary in the following ways:

- Different parameterized offer fields, including valid and expiry dates;
- Different offer codes (number of codes, length, format, custom code generator);
- Custom attributes (which are exposed for a specific type of offer; for example, credit card offers have initial APR% and go-to rates; mortgage offers have payment frequency and term).

As a best practice, keep parameterized values in offers to a minimum. Most offer attributes should not be parameterized. You should create parameters only for those attributes that will not change the "essence" of an offer, such as effective and expiration dates.

Carefully consider the design of your offers and offer templates, as this can significantly affect how you can analyze and report on campaign details.

For information on working with offers, see the Unica Campaign User's Guide.



## Using custom attributes in offer templates

You can create offer attributes and use them on offer templates and offers.

After you create a custom offer attribute, it can be added to any new offer template. Any offers that are created from that template will include the custom attribute.



**Note:** You must reassign the offers, which include newly added offer attributes.

- [Standard offer attributes in Unica Campaign \(on page 121\)](#)
- [Creating or editing custom attributes \(on page 113\)](#)
- [Using drop-down lists in offer templates \(on page 129\)](#)

## Standard offer attributes in Unica Campaign

The following table lists the offer attributes that are delivered with Unica Campaign available in Unica Plan systems that enable offer integration.

**Table 19. Standard offer attributes**


Attribute Display Name	Attribute Internal Name*	Form Element TypeAttribute Type
Average Avg. Response Revenue	AverageResponseRevenue	Text Field - CurrencyMoney
Channel	Channel	Select Box - StringSingle-Select
Channel Type	ChannelType	Select Box - StringSingle-Select
Cost Per Offer	CostPerOffer	Text Field - CurrencyMoney
Creative URL	CreativeURL	Text Field - StringCreative URL
Effective Date	EffectiveDate	Text Field - DateDate Select
Expiration Date	ExpirationDate	Text Field - DateDate Select
Expiration Duration	ExpirationDuration	Text Field - NumericFloat
Fulfillment Cost	FulfillmentCost	Text Field - CurrencyMoney
Interaction Point ID	UACInteractionPointID	Text Field - NumericFloat
Interaction Point	UACInteractionPointName	Text Field - StringText - Single-Line
Offer Fixed Cost	OfferFixedCost	Text Field - CurrencyMoney


\*In Unica Plan systems that enable offer integration, attribute internal names are stored as all lowercase text.


## Creating or editing custom attributes


You can define custom attributes for use on campaigns, offers, or cells on a target cell spreadsheet. When you create the attribute, you specify whether it can be used in campaigns, offers, or cells. You cannot change this selection after you save the attribute.

You must have permissions to add or modify campaign, offer, and cell attributes.



1. Select **Settings > Campaign settings**.
2. In the **Templates and customization** section, click **Custom attribute definitions**.
3. Click the **Add a custom attribute** icon  or click the name of the attribute that you want to modify.
4. Define the attribute.



For this option:	Take this action:
<p><b>Attribute display name</b></p>	<p>Specify the label to identify the attribute for users. For example, "Interest Rate". Double quotes in attribute display names are not supported in target cell spreadsheets. The target cell spreadsheet escapes special decoration for attribute display names. For example, instead of a column name in the target cell spreadsheet appearing in bold red text, the column name would appear exactly as shown here: <code>&lt;strong style="color: red;"&gt;Name&lt;/strong&gt;</code>.</p> <p> <b>Note:</b> You cannot change the name of the standard offer attributes that are supplied with Unica Campaign.</p>
<p><b>Internal name</b></p>	<p>Specify a name to identify this attribute when you write an Unica expression, such as a query or custom macro. Use the same name as the <b>Attribute display name</b>, but without any spaces (for example, "Interest-Rate").</p> <p>The internal name must be globally unique, must start with an alphabetic character, cannot include spaces, and are not case-sensitive.</p> <p>To avoid errors, do not change the internal name of attributes that are used in flowcharts.</p>

For this option:	Take this action:
<p><b>Attribute of</b></p>	<p>Indicate where the attribute can be used. You cannot change this option after you save the attribute.</p> <ul style="list-style-type: none"> <li>◦ A <b>Campaign</b> attribute is included on every campaign, even campaigns that already exist.</li> <li>◦ An <b>Offer</b> attribute is available for use on new offer templates. If the attribute is included on an offer template, then any offers that are based on that template include the attribute.</li> <li>◦ A <b>Cell</b> attribute is included in the target cell spreadsheet for every campaign, even campaigns that already exist.</li> </ul>
<p><b>Description</b></p>	<p>Optional.</p>
<p><b>Mandatory</b></p>	<p>If you want to require a value for this attribute, select <b>Mandatory</b>. This setting has the following result.</p> <ul style="list-style-type: none"> <li>◦ For campaigns, users are required to specify a value for the attribute (the field cannot be left blank).</li> <li>◦ For cells, users are required to specify a value in the target cell spreadsheet (the cell cannot be left blank).</li> <li>◦ For offers, administrators are required to specify a value when the attribute is added to an offer template. The specified value is used for any offers that are based on that template, unless the user specifies a different value when creating or editing the offer.</li> </ul> <p> <b>Note:</b> If you add an offer attribute to an offer template as Static, "Hidden static" or "Parameterized", a value is always required, even if the attribute is defined as non-mandatory.</p> <p>If you change this option after an attribute is in use:</p>

For this option:	Take this action:
	<ul style="list-style-type: none"> <li>◦ If you change from mandatory to non-mandatory, a value is no longer required when the attribute is used.</li> <li>◦ If you change from non-mandatory to mandatory, a value is required whenever this attribute is used in the future. This change has no effect on existing objects unless you edit them. For example, if you open a campaign, target cell spreadsheet, or offer in Edit mode, you are required to specify a value before saving.</li> </ul>
<b>Form element type</b>	<p>Specify the type of data to be stored in the attribute field for offers or cells.</p> <p> <b>Important:</b> After you add the custom attribute, you cannot change its data type.</p>

5. Specify more information, depending on the Form Element Type that you selected.

If you selected this Form element type:	Take this action:
<b>Text field - Numeric</b>	<p>Specify the number of decimal places to display to the right of the decimal point.</p> <p> <b>Note:</b> If you decrease the value for an existing attribute, the display is truncated in the user interface. However, the original value is retained in the database.</p>
<b>Text field - Currency</b>	<p>Specify the number of decimal places (see above).</p> <p> <b>Important:</b> The currency value reflects the number of decimal places that are customarily used in the local currency. If you specify a number of decimal places less than the number commonly used, the currency value is truncated.</p>

If you selected this Form element type:	Take this action:
<p style="text-align: center;"><b>Text field - String</b></p>	<p>Specify a <b>Maximum string length</b> to indicate the maximum number of bytes to store for any value of this attribute. For example, if you enter 32, single-byte languages such as English store 32 characters but double-byte languages store only 16 characters.</p> <p> <b>Important:</b> Decreasing the length of an existing attribute truncates existing values, which, if the field is used for matching purposes, can adversely affect response tracking.</p>
<p style="text-align: center;"><b>Select box - String</b></p>	<ul style="list-style-type: none"> <li>◦ Specify a <b>Maximum string length</b> (see above).</li> <li>◦ Optionally, check <b>Allow addition of list items from within edit forms</b> to allow any user to add new unique values to the list of available values when creating or editing a campaign, offer template, or offer that includes this attribute. (This option does not apply to cells.) For example, if a Select Box on an offer template contains the values Small, Medium, Large, a user could add the value Extra Large when creating an offer or editing the offer template.</li> </ul> <p> <b>Important:</b> After the campaign, offer template, or offer is saved, the user cannot remove the new list item. The value is saved in the custom attribute definition and is then available to all users. Only administrators can remove items from lists, by modifying the custom attribute.</p> <ul style="list-style-type: none"> <li>◦ Populate the <b>Source list of available values</b> to specify which items are available in the Select Box. Enter values in the <b>New or selected item</b> field and click <b>Accept</b>. To delete a val-</li> </ul>

If you selected this Form element type:	Take this action:
	<p>ue, select it in the Source list of available values and click <b>Remove</b>.</p> <ul style="list-style-type: none"> <li>◦ Optionally, specify a <b>Default value</b> for the Select Box. The default value is used on the campaign, offer, or target cell spreadsheet unless the user specifies a different value when creating or editing the campaign, offer, or cell.</li> <li>◦ Specify the <b>Sort order</b> to determine how values appear in the list.</li> </ul>

6. Click **Save changes**.

## Working with offer templates

Every offer is based on an offer template. Therefore, administrators must create offer templates before users can create offers.

You can make limited modifications to a template that has offers based on it (you can change the basic options and default values of attributes). To make other changes, you should retire the original offer template and replace it by creating a new one with the desired changes.

Before you start working with offer templates, you should create any custom offer attributes you might need. For example, you could create a drop-down list consisting of several choices, which users will be able to select from when creating offers.



**Note:** Working with offer templates requires the appropriate permissions. For example, to add offer templates, you must have both the **Add offer templates** and **View offer templates** permissions. For more information, see [Security in Unica Campaign \(on page 21\)](#).

## Creating offer templates

Unica Campaign administrators must create offer templates before users can create offers. Follow these instructions to create offer templates.

1. Select **Settings > Campaign settings** to navigate to Campaign settings.

The Campaign settings page opens, presenting links for various administration tasks.

2. In the **Templates and customizations** section, click **Offer template definitions**.

The Offer template definitions window opens.

3. At the bottom of the list of offer templates, click **Add an offer template**.

The New offer template Step 1 of 3: Metadata window opens.

4. Enter offer template metadata as follows:

- a. Enter data for the basic options: **Template name**, **Security policy**, **Description**, **Suggested uses**, and **Template icon**.
- b. To use this offer template with Unica Interact, select **Allow offers created from this template to be used in real-time interactions**. (This option is not available if **Plan - Offer integration** is enabled in the configuration properties.)
- c. Accept the defaults, or modify data for the offer and treatment code formats and generators: **Offer code format**, **Offer code generator**, **Treatment code format**, and **Treatment code generator**.



**Important:** You cannot use the space character in an offer code format.

If you leave the **Treatment code generator** field empty, the default treatment code generator is used.

5. Click **Next >>**.

The New offer template Step 2 of 3: Offer attributes window opens.

6. Add standard and custom attributes to the offer template as needed. Use the arrow buttons to move attributes into and out of the attribute list for the offer template, and to change the order and type (static, hidden, or parameterized) of included attributes.



**Note:** For an offer to be available in a flowchart, it must have at least one standard or custom attribute.

7. Click **Next >>**.

The New offer template Step 3 of 3: Default values window opens.

8. For attributes that you added to the offer template, supply a default value that will be used when users create offers using this template. When creating offers, users can change the default value of static and parameterized attributes, but cannot change the value you entered for hidden static attributes in the offer template.
9. For parameterized attributes with values supplied in a drop-down list, you can also add list items here, as you create the offer template. You can remove any new list items you add here, but cannot remove any list items that existed previously. Any additions to the list items made here are saved back to the offer custom attribute.



**Important:** If you added the `Offer Valid/Expiration Dates` attribute to the template as a parameterized attribute, on this screen you are presented with a **Flowchart run date** option. If you select this option instead of entering a default Offer valid date, Unica Campaign uses the date on which the process using offer runs, not the run date of the entire flowchart.

10. If you selected **Allow offers created from this template to be used in real-time interactions**, enter an **Interaction point ID** and an **Interaction point name**.

You can enter any integer for the Interaction point ID default value, and any string for Interaction point name.

The values are automatically populated with the correct data by the runtime environment, however, the design environment requires a default value.

11. Click **Finish**.

You have created the offer template. It can now be used for creating offers.

## Modifying offer templates

If an offer template has offers based on it, you can change the basic options and default values of attributes in the template. However, you cannot modify template data about offer codes or offer custom attributes. To make these changes, retire the original offer template and replace it by creating a new one with the desired changes.

1. Select **Settings > Campaign settings**.
2. Click **Offer template definitions**.
3. Click the name of an offer template.

The **New offer template Step 1 of 3: Metadata** window opens.

If the offer template is currently used by an offer, you can only edit the basic options. If the offer template is not being used, you can also edit the offer and treatment code data.

4. Click **Next >>**.

The **New offer template Step 2 of 3: Offer attributes** window opens.

5. Modify the attribute settings as needed.



**Note:** If the offer template is currently used by an offer, you cannot change settings for offer attributes. If the template is not being used, you can modify the attributes in the offer template as needed. Use the arrow buttons to move attributes into and out of the attribute list for the offer template, and to change the order and type (static, hidden, or parameterized) of included attributes.

6. Click **Next >>**.

The **New offer template Step 3 of 3: Default values** window opens.

7. Supply a default value for attributes in the offer template.

When creating offers, users can change the default value of static and parameterized attributes. However, users cannot change the value that you enter for hidden static attributes.



**Important:** If you added the **Offer valid/expiration dates** attribute to the template as a parameterized attribute, on this screen you are presented with the option **Flowchart run date**. If you select this option





instead of entering a default Offer Valid date, Unica Campaign uses the date on which the process that uses the offer runs, not the entire flowchart.

8. Click **Finish**.

## Using drop-down lists in offer templates

A drop-down list, also known as a Select Box, is a list of values from which users can select a single item when they define an offer.

Follow this procedure to make a drop-down list available in offer templates (and therefore in offers).

1. Define a custom offer attribute of the type **Select Box - String**. Specify the list of available values when you define the custom offer attribute. See [Creating or editing custom attributes \(on page 113\)](#).
2. Add the attribute to an offer template. See [Creating offer templates \(on page 126\)](#).
3. To determine whether users can specify additional values when they configure a contact process: Choose **Settings > Configuration** and adjust the global property `Unica Campaign | partitions | partition[n] | server | flowchartConfig | disallowAdditionalValForOfferParam`.

Any offers that are based on the offer template will include the drop-down list. Users can select values from the drop-down list when they define the offer.

## Defining a list of outbound communication channels

Unica Campaign includes the **Channel** attribute for use in offer templates. Modify the **Channel** attribute to define a list of available outbound communication channels, such as e-mail or phone, for offers.

As delivered, the **Channel** attribute does not include any available values. To make use of the **Channel** attribute, you must modify it to provide the values from which users can select. To modify the attribute and define the available values, see [Creating or editing custom attributes \(on page 113\)](#).

1. Select **Settings > Campaign settings** to navigate to Campaign settings, or in case of Unica Centralized Offer Management, select **Settings > Offer settings**.
2. Click **Custom attribute definitions**.
3. Click the **Channel** attribute.
4. The **Channel** attribute is defined as a **Select Box - String**. Modify the attribute to specify the list of available values.

For more information, see [Creating or editing custom attributes \(on page 113\)](#).

5. Add the attribute to an offer template: Select **Settings > Campaign settings** and click **Offer template definitions**.

For more information, see [Modifying offer templates \(on page 128\)](#).

6. Choose **Settings > Configuration** and adjust the global property `Unica Campaign | partitions | partition[n] | server | flowchartConfig | disallowAdditionalValForOfferParam` to determine whether users can specify additional values when they configure a Mail list, Call list, or Optimize process.

## Changing the order in which offer templates appear

You can adjust the order in which offer templates appear for users when they create a new offer. By default, offer templates are listed in the order in which you created them.

Users see only the specific offer templates allowed by the security policy of the offer template and the roles of the user. Therefore, each user might see a different set of offer templates. The order that you specify is the order in which those templates appear.

1. Select **Settings > Campaign settings**.
2. Click **Offer template definitions** in the **Templates and customizations** section.
3. At the top or bottom of the list of offer templates, click **Reorder**.
4. Select one template at a time and click the up or down icon to move the template up or down in the list.
5. Click **Save changes**.

## Retiring offer templates

Offer templates cannot be deleted, but they can be retired by administrators to prevent further use. Retired templates are greyed out in the list of offer templates and cannot be used for creating new offers.

Retire an offer template when you no longer want users to be able to create new offers based on it. Previously created offers based on the template are not affected.





**Note:** After you retire an offer template, you cannot un-retire it. You would have to create a new offer template with the same characteristics.

1. Select **Settings > Campaign settings**.
2. Click **Offer template definitions**.
3. To the right of the offer template, click **Retire**.
4. Click **OK** to confirm.

## Template icons

Select a template icon as part of the **Basic options** when you create or modify an offer template. The template icon provides a visual cue to users when they create new offers based on this template.

For example, an offer template that is intended for telephone offers might use this icon:  The default icon looks

like this: 

To see the available icons, choose one from the **Template icon** list when creating or modifying an offer template.

## Default offer attributes

When you create an offer template, you can add template attributes as needed.

By default, the following static attributes are included in all offer templates:

- **Name**
- **Description**
- **Offer code**
- **Relevant product(s)**

You cannot remove these static attributes from a template.

## Introduction to using Unica Plan assets in Unica Campaign offers

If both Unica Plan and Unica Campaign are installed, and you licensed the Marketing Asset Management add-on for Unica Plan, your campaigns can include digital assets from your Unica Plan asset libraries. Unica Campaign does not need to be integrated with Unica Plan, although it may be.

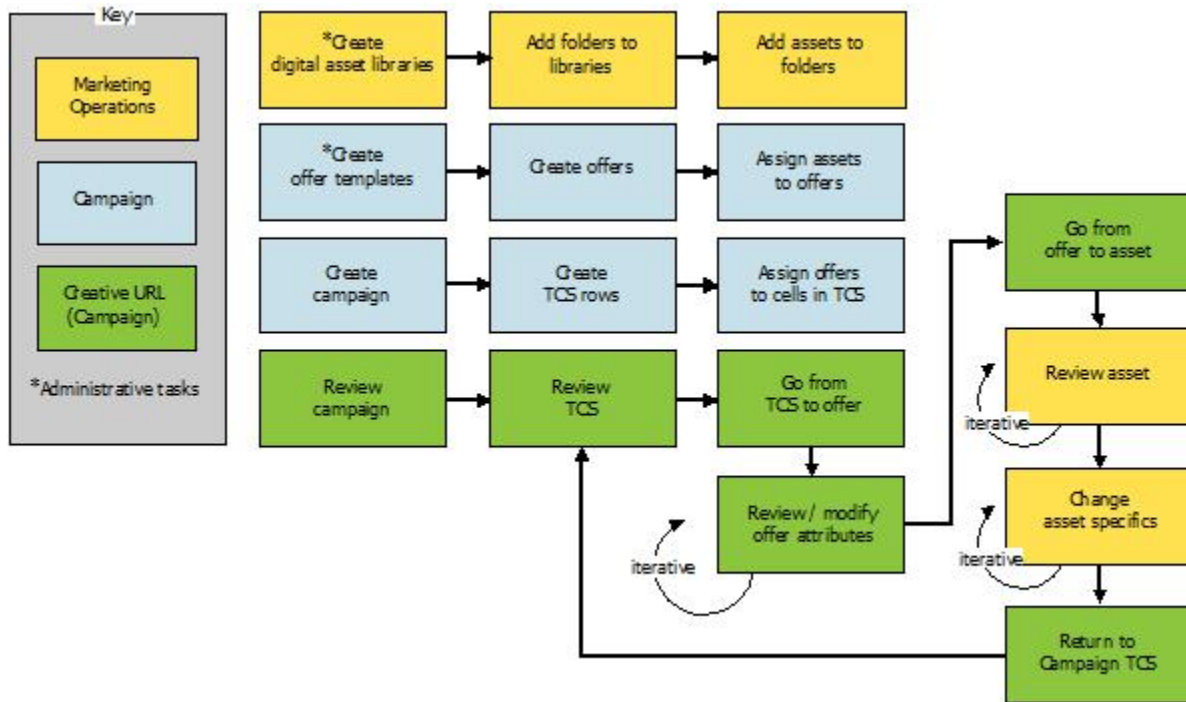
An example of this functionality is to create an offer that includes a product logo stored in a Unica Plan asset library.

To include a Unica Plan asset in an offer, a user creates an offer based on a template that includes the **CreativeURL** attribute. A "Creative URL" is a pointer that indicates the location of an asset in Unica Plan. The asset that the **CreativeURL** attribute points to is included in the offer.

The **CreativeURL** attribute allows users to move seamlessly from Unica Campaign to Unica Plan when configuring offers, offer templates, or campaigns.

For example, when creating or editing a campaign, you can go from a target cell spreadsheet (TCS) cell to the offer related to that cell. From the offer, you can go to the related asset in Unica Plan, where you can view or modify it. You can also upload a new asset to the library for immediate use in the campaign.

The following example shows one possible workflow for a system. This example is for a system that is not integrated. Your workflow might differ.



## Guidelines for using Unica Plan assets in Unica Campaign offers

This topic lists the prerequisites and requirements for using Unica Plan assets in Unica Campaign offers. This functionality depends on the **CreativeURL** offer attribute.

- Both Unica Plan and Unica Campaign must be installed. (The **CreativeURL** attribute is installed with Unica Campaign. However, you cannot use the feature unless Unica Plan is also installed.)
- You must license the Marketing Asset Management add-on for Unica Plan.
- Unica Campaign may or may not be integrated with Unica Plan. Even if UMO-UC integration is off, users can assign an asset to an offer.
- **CreativeURL** is a standard Unica Campaign offer attribute but it is not required. You can create offer templates with or without the attribute.
- If the **CreativeURL** attribute is included in a template, each offer based on that template must include an asset from a Unica Plan asset library.
- An offer template, and any offers based on it, can include only one **CreativeURL**. Therefore, each offer can include only one asset from Unica Plan.



**Note:** An offer can be related to only one asset. However, one asset can be related to multiple offers.

# Chapter 6. Audience level administration

Unica Campaign is delivered with a single audience level called Customer. You can define any additional audience levels that you need. Audience levels let flowchart designers target specific groups, such as Households, in marketing campaigns.

Unica Campaign administrators can perform the following tasks:

- Create the audience levels needed for your company's campaigns.
- Create database tables in the Unica Campaign system database to support the new audience levels.
- Map system tables to the supporting database tables for the new audience levels in the Unica Campaign system database.
- Specify audience levels and associated database fields when mapping user tables.
- Create global suppression segments for one or more audience levels.

## About audience levels

An audience level is a collection of identifiers that can be targeted by a campaign.

For example, a set of campaigns could use the audience levels "Household," "Prospect," "Customer," and "Account." Each of these levels represents a certain view of the marketing data available for a campaign.

Audience levels are typically organized hierarchically. Using the examples above:

- Household is at the top of the hierarchy, and each household can contain multiple `customers` as well as one or more `prospects`.
- Customer is next in the hierarchy, and each customer can have multiple accounts.
- Account is at the bottom of the hierarchy.

Other, more complex examples of audience hierarchies exist in business-to-business environments, where audience levels may need to exist for businesses, companies, divisions, groups, individuals, accounts, and so on.

These audience levels may have different relationships with each other, for example one-to-one, many-to-one, or many-to-many. By defining audience levels, you allow these concepts to be represented within Unica Campaign so that users can manage the relationships among these different audiences for targeting purposes. For example, although there might be multiple prospects per household, you might want to limit mailings to one prospect per household.

An audience level is composed of a fixed number of keys or database table fields, which together uniquely identify a member of that audience level.

For example, the audience level "Customer" might be identified by a single `IndivID` field or by a combination of a `HouseholdID` field and a `MemberNum` field.

For more details about audience levels, see the section on the Audience process in the Unica Campaign User's Guide.

## Why different audience levels are needed in Unica Campaign

Different audience levels enable flowchart designers to target and switch between specific identifiable groups in their campaigns, or to scope one audience level by another (for example, to target one individual per household).

For example, audience levels enable developers to do such things as:

- Select the customer with the highest account balance in each household.
- Select all accounts with a negative balance belonging to a particular set of customers.
- Select all households with at least one individual holding a checking account.

For more details about audience levels, see the section on the Audience process in the Unica Campaign User's Guide.

## The default Customer audience level

Unica Campaign is delivered with a single audience level called `Customer`. You can define additional audience levels as required for your user tables and campaign needs.

By default, the Unica Campaign system database contains the tables needed to support the `Customer` audience level. After installing Unica Campaign, you must map these tables.

## About additional audience levels and system tables

If you require additional audience levels, you must create and map the equivalent set of system tables to support them as you did for the default Customer audience level.

You must define audience levels before you map user tables, so that you can specify audience levels during the user table mapping process. Base tables mapped at a specific audience level, when queried, will return IDs at that audience level.

Before you create an additional audience level, you must create four tables in the Unica Campaign system table database.

Each audience level you create requires the following associated system tables:

- A contact history table
- A detailed contact history table
- A response history table
- A segment membership table

When you create the audience level, system table entries are automatically created.

After you create the audience level, you map these system tables to the database tables.



**Note:** It is recommended that you map the segment membership table only if you are using strategic segments with Unica Campaign flowcharts or Optimize sessions in Unica Optimize.

## Default Customer audience level system tables

Unica Campaign is delivered with system table ddl scripts to create the tables that support the default `Customer` audience level.

After installing Unica Campaign, you must map these system tables to tables in the Unica Campaign system database as follows:

**Table 20. Default audience level system tables**

Unica Campaign System Table	Database Table Name
Customer Contact History	UA_ContactHistory
Customer Response History	UA_ResponseHistory
Customer Detailed Contact History Table	UA_DtlContactHist
Customer Segment Membership	UA_SegMembership

If these tables are mapped as listed above, then the sample reports provided with Unica Campaign will work with a minimum number of changes.

The SQL statements used to create these tables and their related indices can be used as templates to create tables for other audience levels.

## About audience levels and strategic segments

For each audience included in a flowchart or Optimize session that uses strategic segments, you map the segment membership system table to a physical table that defines segment members.

For example, to use the default `Customer` audience in an optimization session that includes strategic segments, you must map the audience system table `Customer Segment Membership` to the `UA_SegMembership` segment database table. You populate the database table using the `Create Seg` process.



**Note:** recommends that you map an audience's segment membership table only if you plan to use the audience in flowcharts or Optimize sessions that use strategic segments.

Using strategic segments in Unica Campaign flowcharts or Unica Optimize sessions is optional. If you map the segment membership table, each time you run the flowchart or Optimize session, Unica Campaign or Unica Optimize updates the table. This is unnecessary processing overhead if you are not using strategic segments.

## Audience level unique identifiers

When you create a new audience level, you must specify at least one field to be used as the unique identifier for members of that audience level. To uniquely identify each member of the audience, you might need to use multiple fields.

For example:

- `Household` could be identified by the field `HHold_ID`
- `Customer` could be identified by the fields `HHold_ID` and `MemberNum`.
- `Prospect` could be identified by the field `Prospect_ID`.
- `Account` could be identified by the field `Acct_ID`.

The field names (and in particular any unique identifier field names) in the new audience level must exactly match the field name(s) in the database table for mapping. This allows Unica Campaign to automatically match the database fields to the appropriate system table fields when you create the audience level.



**Note:** Audience level field names have specific character restrictions. For details, see [Special characters in Unica Campaign object names \(on page 542\)](#)

## Required fields in audience-level-specific tables

This section provides lists of the required fields in the system tables required for each audience level.

- [Required fields in the contact history table \(on page 136\)](#)
- [Required fields in the detailed contact history table \(on page 137\)](#)
- [Required fields in the response history table \(on page 138\)](#)
- [Required fields in the segment membership table \(on page 139\)](#)

### Required fields in the contact history table

The contact history table for each audience level in the Unica Campaign system database must minimally contain the fields described in this section.

**Table 21. Required fields in the contact history table**

Key	Column Name	Data Type	Length	Allow Nulls
Yes	Identifier for audience level	numeric or text		No
Yes	CellID	bigint	8	No
Yes	PackageID	bigint	8	No



**Table 21. Required fields in the contact history table (continued)**

Key	Column Name	Data Type	Length	Allow Nulls
No	ContactDateTime	datetime	8	Yes
No	UpdateDateTime	datetime	8	Yes
No	ContactStatusID	bigint	8	Yes
No	DatelD	bigint	8	Yes
No	TimeID	bigint	8	Yes



**Note:** Unica Campaign is shipped with additional fields (`valueBefore` and `usageBefore`) in the `UA_ContactHistory` table for the Customer audience level to support the example reports. You can define your own "additionally tracked fields" for contact history and customize reports as needed.

## Required fields in the detailed contact history table

The detailed contact history table for each audience level in the Unica Campaign system database must minimally contain the fields described in this section.

**Table 22. Required fields in the detailed contact history table**

Key	Column Name	Data Type	Length	Allow Nulls
Yes	Identifier for audience level	numeric or text		No
No	TreatmentInstID	bigint	8	No
No	ContactStatusID	bigint	8	Yes
No	ContactDateTime	datetime	8	Yes
No	UpdateDateTime	datetime	8	Yes
No	DatelD	bigint	8	No

**Table 22. Required fields in the detailed contact history table (continued)**

Key	Column Name	Data Type	Length	Allow Nulls
No	TimeID	bigint	8	No

### Required fields in the response history table

The response history table for each audience level in the Unica Campaign system database must minimally contain the fields described in this section.

**Table 23. Required fields in the response history table**

Key	Column Name	Data Type	Length	Allow Nulls
Yes	Identifier for audience level.	numeric or text		No
Yes	TreatmentInstID	bigint	8	No
Yes	ResponsePackID	bigint	8	No
No	ResponseDateTime	datetime	8	No
No	WithinDateRangeFlg	int	4	Yes
No	OrigContactedFlg	int	4	Yes
No	BestAttrib	int	4	Yes
No	FractionalAttrib	float	8	Yes
No	CustomAttrib	float	8	Yes
No	ResponseTypeID	bigint	8	Yes
No	DateID	bigint	8	Yes
No	TimeID	bigint	8	Yes

**Table 23. Required fields in the response history table (continued)**

Key	Column Name	Data Type	Length	Allow Nulls
No	DirectResponse	int	4	Yes

Each response history table you create for a new audience level should have a foreign key constraint on the `UA_Treatment` table's `TreatmentInstID` field.

## Required fields in the segment membership table

If you are using strategic segments in Unica Campaign or Unica Optimize, you must create the segment membership table for each audience level that you use with strategic segments. At a minimum, the table must contain the fields described in this section.

**Table 24. Required fields in the segment membership table**

Key	Column Name	Data Type	Length	Allow Nulls
Yes	SegmentID	bigint	8	No
Yes	Identifier for audience level	numeric or text		No

## About audience levels and user tables

A user table can be associated with a single audience level or with multiple audience levels.

This section contains the following information:

- [User tables with a single audience level \(on page 139\)](#)
- [User tables with multiple audience levels \(on page 140\)](#)

### User tables with a single audience level

When you map a user table, you must specify at least one audience level to be the primary audience of that table.

During this step, Unica Campaign associates the field(s) you specified when creating the audience level with the identifier field(s) of the same name in the user table. By doing this, you have specified that by default, when Unica Campaign makes selections from this user table, IDs are returned from the primary audience level.

For example, if you create an audience level named `Account` with the field `Acct_ID`, and select this audience level as the primary audience when mapping the user table `Accounts`, you have associated the `Acct_ID` audience level field with the field in the user table that is the unique identifier (primary key) for the `Accounts` database table.

## User tables with multiple audience levels

A user table can be associated with multiple audience levels, with one of the audience levels designated as the primary audience level, and the remaining audience levels as alternate audience levels.



**Note:** To enable flowchart designers to switch from one audience level to another or to scope one audience level by another, you must define at least one user table with all required audience levels. Unica Campaign can then use this table to "translate" between one audience level and another as needed.

For example, a user table that contains data about customer accounts could have the following columns:

- Acct\_ID
- Indiv\_ID
- HHold\_ID

In this table, the `Acct_ID` might be unique for each record. Because an individual could have multiple accounts, and a household could contain multiple individuals, multiple records could have the same values for the `Indiv_ID` and `HHold_ID` fields.

Assuming you have three audience levels, `Account`, `Customer`, and `Household`, when you map this user table, you could specify all three of these audience levels and associate them with the corresponding fields listed above in the user table. This enables flowchart designers to switch target audiences, or scope one audience level by another (for example, accounts per customer, customers per household, or accounts per household) when using this table.

## Workflow for setting up a new audience level

The listed tasks provide the workflow for setting up a new audience level.

See each task for the specific procedures.

- [Task 1: Create the required database tables for each new audience level \(on page 140\)](#)
- [Task 2: Create the new audience levels in Unica Campaign \(on page 141\)](#)
- [Task 3: Map the Unica Campaign system tables to the database tables \(on page 142\)](#)
- [Task 4: Map the user tables containing relevant data to the appropriate audience level\(s\) \(on page 143\)](#)
- [Task 5: Save the mapped tables in a table catalog \(on page 143\)](#)

### Task 1: Create the required database tables for each new audience level

This task is part of the workflow for setting up a new audience level.

You must create physical database tables in the Unica Campaign system database to support each new audience level(s) you create. The required tables for each audience level are:

- A contact history table
- A detailed contact history table

- A response history table
- A segment membership table

Each required table has a set of required fields. You can create additional custom fields in your audience tables.



**Note:** You should create indices on the tables you create. For example, if you create the `INDIV_ContactHistory` table for a new Individual audience level, you could create an index as follows: `CREATE INDEX`

```
XIE1INDIV_ContactHistory ON INDIV_ContactHistory ( IndivID ).
```

To create the tables for other audience levels, you may want to use the SQL statements used to create the Unica Campaign default audience level tables and their related indices as templates. For example, you could use `UA_ContactHistory` as a template for `Acct_ContactHistory` (for an audience level `Account`). To see the available SQL statements, look in the `/Campaign/ddl` directory for the script that creates system tables for your database management system.



**Note:** You have the flexibility to map multiple system tables for a new audience level to the same underlying physical database table (containing sufficient audience fields to represent all necessary audience levels), or you can create separate database tables for each audience level. Consulting or your implementation partner can help you decide how to best implement contact and response history tables for your environment.

## Task 2: Create the new audience levels in Unica Campaign

This task is part of the workflow for setting up a new audience level.

1. Select **Settings > Campaign settings**.
2. On the Campaign Settings page, under **Data source operations**, click **Manage audience levels**.

The **Audience levels** dialog opens, displaying existing audience levels.

3. Click **Create new**.
4. Enter a unique **Audience level name**, which should reflect the group of identifiers at that audience level.



**Note:** Audience level names have specific character restrictions.

5. In the **Field list**, enter a name and select the type (numeric or text) for each field that will be used to uniquely identify each member of the audience level.



**Note:** Audience level field names have specific character restrictions.

You must specify exactly the same names as the field names in the database tables for this audience level. You will not be able to map a field in the next step unless Unica Campaign finds an exact match for the field name.

For example, if you are creating the audience level "Household" and you specify one field for unique audience level identifier called "HouseholdID," you must make sure that the ID field in your audience level-specific database tables matches this exactly; in other words, that it is also named "HouseholdID."

6. Click **OK**.

In the **Audience levels** dialog, when you select the new audience level, you see the required tables listed as "Not mapped." Your next step is to map the Unica Campaign system tables to the database tables.

### Task 3: Map the Unica Campaign system tables to the database tables

This task is part of the workflow for setting up a new audience level.

After creating the physical database tables for each new audience level and the audience levels in Unica Campaign, you must map the Unica Campaign system tables to these database tables.

You can map user tables to your created audience levels without mapping the Unica Campaign system tables to database tables, but you will not be able to log contact and response history without mapping the Contact History, Detailed Contact History, and Response History tables.

HCL recommends that you map segment membership system tables to physical database tables only for audiences used in Unica Campaign flowcharts or Unica Optimize sessions that contain strategic segments. Using strategic segments in Unica Campaign and Unica Optimize is optional.

1. Select **Settings > Campaign settings**.
2. On the Campaign Settings page, under **Data source operations**, click **Manage audience levels**.

The Audience levels dialog opens, displaying existing audience levels.

3. Select the audience level for which you are mapping database tables, and click **History tables**.
4. In the **Table mappings** dialog, select each Unica Campaign system table and click **Map table**.
5. In the table mapping dialog, select the database table that corresponds to the Unica Campaign system table for that audience level. The **Source table fields** list is populated with the fields from the database table that you selected. The **Required fields** list is populated with selected fields (from your source database table) and the corresponding required fields (from the Unica Campaign system table).



**Important:** Fields can be mapped only if Unica Campaign finds an exact match for the field names.

6. Click **Next** to specify mapping for any custom fields in your database tables.
7. Click **Next** to specify display names for custom fields. This option is not available for all tables.
8. Click **Finish** to complete the mapping. Repeat this procedure for each required Unica Campaign system table for the audience level.



**Note:** You can also perform this task from the **Manage table mappings** link on the Campaign Settings page.

## Task 4: Map the user tables containing relevant data to the appropriate audience level(s)

This task is part of the workflow for setting up a new audience level.

When you map a user table, you must specify one primary audience level. You can also specify one or more alternate audience levels.

For each audience level, map to the user table containing the identifier for the entity at that audience level.

## Task 5: Save the mapped tables in a table catalog

This is the final task in the workflow for setting up a new audience level.

(Optional). Save your mapped tables in a table catalog so that you can reload the catalog without remapping individual tables.

## Removing an audience level

When you remove an audience level, the system tables are removed, but the underlying database tables remain.

Therefore, if you remove an audience level, any processes and flowcharts depending on that audience level (that is, that attempt to write to the audience level tables) will generate errors.



**Important:** Do not remove an audience level that has been used within Unica Campaign, as this will result in significant system problems, as described below.



**Important:** Before removing an audience level, recommends backing up the entire Unica Campaign system so that you can recover the current system state if you encounter problems following the removal.

It is possible to restore a deleted audience level by creating a "new" audience level of the same name, with tables containing the same required fields, and remapping the audience level tables.

## To remove an audience level

Use caution when you remove an audience level. Do not remove an audience level that has been used within Unica Campaign, as this causes significant system problems.

1. Select **Settings > Campaign settings**.

The Campaign Settings page appears.

2. Under **Data source operations**, click **Manage audience levels**.

The Audience Levels window opens, displaying previously defined audience levels.

3. Select the audience level to remove.
4. Click **Remove**.

You are prompted to confirm the removal.

5. Click **OK**.

## About global suppressions and global suppression segments

Use the global suppression feature to specify a list of IDs (in a single audience level) that are automatically excluded from all cells in flowcharts in Unica Campaign.



**Note:** Specifying and managing global suppression segments requires the "Manage Global Suppressions" permission in Unica Campaign.

You do this by creating this list of unique IDs as a strategic segment, then by specifying that segment as a global suppression segment for a particular audience level. Only one global suppression segment can be configured for each audience level.

If a global suppression segment has been configured for an audience level, all top-level Select, Extract, or Audience processes associated with that audience level automatically exclude the IDs in the global suppression segment from their output results, unless the global suppression is explicitly disabled for a specific flowchart. By default, flowcharts have global suppression enabled so that no action needs to be taken for any configured global suppression to be applied.

For details about disabling global suppressions, see the Unica Campaign User's Guide.

An exception to the default of global suppression being enabled is the flowchart containing the CreateSeg process that created the global strategic segment itself. In this case, the global suppression is always disabled (only for the audience level for which the global suppression segment is created).

## Switching audiences with global suppressions

If you are switching from Audience 1 to Audience 2 in a flowchart, and have one global suppression defined for each of these audience levels, the global suppression segment for Audience 1 is applied to the input table, and the global suppression segment for Audience 2 is applied to the output table.

## About creating global suppression segments

You create global suppression segments by performing the following tasks:

- [To create the global suppression segment in a flowchart \(on page 144\)](#)
- [To specify the segment as a global suppression segment \(on page 145\)](#)

To create the global suppression segment in a flowchart

The best practice for creating or updating global suppression segments is to do this when no flowcharts at the same audience level are running (and thus potentially using the segment(s)). The consistency of suppression lists cannot be guaranteed if global suppression segments are created or updated while flowcharts are using them.



1. Create a strategic segment in a flowchart in the usual way, naming it in such a way that you can identify it easily for selection from a list. For details about creating strategic segments, see the *Unica Campaign User's Guide*.
2. In the CreateSeg process configuration dialog, on the Define Segments tab, click **Edit...**
3. In the Edit Segment window, in the **Temp Table Data Source** field, select one or more data sources.

You should specify all data sources where the global strategic segment is commonly used. If the strategic segment is not persisted in a data source, then the suppression is done on the Unica Campaign server using the binary file. If the CreateSeg process is unable to create the strategic segment or write the segment to any of the specified data sources, it will be unconfigured or fail when run.

Changes to the temp table data sources are made at the time you save the process configuration, not when the flowchart is saved or run.

4. Click **OK**.

On the Define Segments tab, you see that the data source you selected is displayed in the Temp Table DS column for the current segment.

## To specify the segment as a global suppression segment

Use this procedure to specify a segment as a global suppression segment.

1. After creating the segment that you want to use as a global suppression segment, in Unica Campaign, select **Settings > Campaign settings**.

The Campaign Settings page appears.

2. On the Campaign Settings page, click **Manage audience levels**.
3. In the Audience Levels window, select the audience level for which you are specifying a global suppression segment.
4. Click **Global suppressions...**

In the Global Suppression Segment window, the drop-down list displays the list of segments matching the current audience level.

5. Select the segment to use as the global suppression segment for the current audience level, then click **OK**.
6. Click **Close**.

The selected strategic segment is specified as the global suppression segment for the audience level.

On the Unica Platform Configuration page, the global suppression segment, when defined, is displayed in audience level properties in the following path:

```
partitions > partition[n] > audienceLevels > audienceLevelN > globalSuppressionSegmentID.
```

## Updating global suppression segments

You update global suppression segments in the same way as you would update strategic segments. For details about editing strategic segments, see the Unica Campaign User's Guide.



**Important:** The best practice for creating or updating global suppression segments is to do this when no flowcharts at the same audience level are running (and thus potentially using the segment(s)). The consistency of suppression lists cannot be guaranteed if global suppression segments are created or updated while flowcharts are using them.

## Deleting global suppression segments

You delete global suppression segments in the same way as you would delete strategic segments. For details about deleting strategic segments, see the Unica Campaign User's Guide.

When the flowchart that created the global suppression segment is deleted, the segment is also deleted.

## Logging for global suppressions

Information relating to global suppressions is included in the flowchart log.

The following information is included:

- The global suppressions segment name (and path) for processes in which it is applied
- The number of IDs before suppression
- The number of IDs after suppression

# Chapter 7. Contact history administration

Contact history is stored in the Unica Campaign system database in separate tables by audience level. Therefore, you need to set up audience levels before you begin working with contact history.

Before you begin working with contact history, you should read all audience level administration topics and set up the required audience levels.

In addition, you can find basic concepts about contact history, and information about setting up flowcharts to record contact history, in the Unica Campaign User's Guide.

## Contact history concepts

Contact history is maintained in the base contact history and detailed contact history tables in the Unica Campaign system database. Contact history is maintained separately for each audience level. Offer history and treatment history are used together with contact history to form a complete historical record of offers that are sent.

The following topics provide conceptual information about contact history.

### What is contact history?

Contact history is the historical record of your direct marketing efforts or communications, including detailed information about who was contacted, when, with what message or offer, and on what channel.

Contact history typically includes both targets contacted through campaigns, as well as your hold-out controls who do not receive a communication, but are measured for comparison purposes against the target group.

In Unica Campaign, contact history includes the record of the exact version offer given to each ID, including the values of personalized offer attributes, providing a complete historical view of your marketing communications.

For example, a campaign may produce a list of targeted customers, output through a Call List or Mail List process. That list of customers is written to the contact history table for the example `Customer` audience level, `UA_ContactHistory`, in the Unica Campaign system database.

Contact history is recorded and stored in the Unica Campaign system database. There is a separate entry for a base contact history system table for each audience level you create. The base contact history stores the audience membership within each target and control cell used in your marketing campaigns, when all audience entities within the same cell receive exactly the same offer(s). Data from the base contact history table is used in conjunction with the `UA_Treatment` system table to resolve exactly who received which offers.



**Note:** If users turn off contact history logging in the Call List or Mail List process, contact history made in that process is not written to the database.

Contact history is written to the database for production runs only, not test runs.

## What is detailed contact history?

Detailed contact history is populated only when data-driven offer personalization is used (where individuals in the same cell receive different offer versions: that is, offers with different values for personalized offer attributes). These details are written to the detailed contact history table (for example, `UA_Dt1ContactHist`) for each audience level.

There is a separate entry for a detailed contact history system table for each audience level you create. The detailed contact history stores the exact treatment(s) each audience entity received.

Detailed contact history records one row per audience ID-offer version pair. For example, if an individual receives three different offer versions, three rows are written to detailed contact history for that individual, and the three treatments will appear in the `UA_Treatment` table.



**Note:** If users turn off contact history logging in the Call List or Mail List process, detailed contact history made in that process is not written to the database.

Detailed contact history is written to the database for production runs only, not test runs.

## What is contact status?

Contact status is the indicator of the type of contact made.

Unica Campaign users specify the contact status to use when configuring a Call List or Mail List process.



**Note:** Control cells automatically receive the contact status with a value of 2 in the Defaults column. By default, the Name of that row is Contact.

Unica Campaign is delivered with a set of default contact status codes. As an administrator, you can add additional status codes.

## About updating contact status

You use the Track process to update contact statuses, as well as other tracked fields in the contact history.

For example, a Mail List process may record customer contacts in the `UA_ContactHistory` table. The contacts would have temporary contact status with a value in the `CountsAsContact` field of 0. The campaign manager then sends this list of contacts to a mail house. The mail house performs post-processing on the list, eliminating addresses that are no longer valid, and returns a list of customers who were actually contacted. A different flowchart then selects the customers from the returned list and uses the Track process to update the contact status to one with a `CountsAsContact` field of 1.

## How does contact history relate to audience levels?

Unica Campaign can record and maintain a separate contact history and detailed contact history for each audience level you have defined.

Each audience level should have its own related contact history and detailed contact history tables in the Unica Campaign system database.

## How does contact history relate to database and system tables?

Contact history tables, which must exist in the Unica Campaign system database, store historical contacts for each audience level.

The `Customer` audience level is delivered as an example, and the history of contacts targeting customers can be stored in the `UA_ContactHistory` in the Unica Campaign system database. Detailed history for the `Customer` audience level can be stored in the `UA_DtlContactHist` table.

If you create an additional audience level, you must create the contact history and detailed contact history tables and related indexes for it in the Unica Campaign system database. You can use the tables for the example `Customer` audience level as templates.

After you create the tables in the Unica Campaign system database for a new audience level, you must map the new tables for the audience level's contact history and detailed contact history.

## What is offer history?

Offer history is the historical record of offers made through campaigns. It is part of the overall historical record of contacts made through your campaigns.

Offer history is stored across multiple tables in the Unica Campaign system table database:

- `UA_OfferHistory` table
- `UA_OfferHistAttrib` table (for parameterized offer attributes),
- `UA_OfferAttribute` table (for static offer attributes)

For example, a typical flowchart results in a list of targeted customers, output through a Call List or Mail List process. A record of the offer(s) made in that flowchart are written to the `UA_OfferHistory` table offer history.



**Note:** If users turn off contact history logging in the Call List or Mail List process, offer history made in that process is not written to the database.

Offer history is written to the database for production runs only, not test runs.

Offer history is not stored in separate tables by audience level; all offer history is stored across the same set of system tables.

## What is treatment history?

Treatment history is the record of treatments generated through campaigns, including both target and control treatments. A treatment is a unique combination of a cell, an offer, and time (a specific flowchart run). If you run the same flowchart multiple times, a new treatment is generated each time.

Treatment history is stored in the `UA_Treatment` table in the Unica Campaign system table database, and is used together with contact history to form a complete historical record of offers sent to IDs in cells, and the specific details of each sent offer's attributes.

Cell membership is recorded in the `UA_ContactHistory` table for the appropriate audience level and the treatment(s) given to each cell are recorded in the `UA_Treatment` table. This is a highly compressed and efficient way to store complete historical information. For example, if all 10,000 people in a cell receive the same three offers, rather than writing  $3 * 10,000 = 30,000$  records to contact history, 10,000 rows are written to contact history recording the individuals within the cell, and 3 rows are written to the `UA_Treatment` table representing the treatments.



**Note:** If users turn off contact history logging in the Call List or Mail List process, the treatment history made in that process is not written to the database.

Offer history is written to the database for production runs only, not test runs.

Treatment history is not stored in separate tables by audience level; all treatment history is stored in the `UA_Treatment` table.

## Creating contact history tables for new audience levels

When you create a new audience level, you might need to create a table in the Unica Campaign system table database to store contact history and detailed contact history for targets and controls in that audience level.

When you create these tables, you must create indexes on them. For example, if you create the `INDIV_ContactHistory` table for a new `Individual` audience level, you could create an index as follows:

```
CREATE INDEX XIE1INDIV_ContactHistory ON INDIV_ContactHistory ( IndivID )
```

When you create a new audience level, you must map the contact history and detailed contact history system tables for the new audience level.

## Adding contact status codes

You can add your own contact status codes to supplement the contact statuses that are delivered with Unica Campaign. Define new contact status codes in the `UA_ContactStatus` table in the Unica Campaign system database. A contact status indicates the type of contact made (for example, Delivered, Undelivered, Control).

If the contact statuses that are supplied with Unica Campaign do not meet your needs, use the following procedure to add contact statuses. Unica Campaign users specify a contact status when they configure a Call List or Mail List process. They configure a Track process to update contact statuses.

1. Log in to the database management system that contains the Unica Campaign system table database.
2. Open the `UA_ContactStatus` table.
3. Add rows for new contact statuses. For each new status:

- a. Enter a unique `ContactStatusID`.



**Note:** The `ContactStatusID` can be any unique positive integer within the `internalIdLowerLimit` and `internalIdUpperLimit` configuration parameter values that are defined on the **Configuration** page in Unica Platform.

- b. Enter a `Name`.
- c. Optionally, enter a `Description`.
- d. Enter a unique `ContactStatusCode`. You can use the values A-Z and 0-9.
- e. In the `CountsAsContact` column, enter `1` if the status represents a successful contact, or `0` if it does not.



**Note:** This column is used by Unica Optimize to manage contact fatigue. It also can be useful for queries against the contact history table to suppress individuals who received a specific number of contacts within some time period.

- f. In the `Defaults` column, enter `0` if that status is not to be the default, or `1` if it is. Enter `2` for the default status for control cells. Ensure that only one row has the value of `1` and one row has the value of `2` in this column.

4. Save the table changes.

If necessary, see the documentation for your database management system for detailed instructions on modifying data in tables.

## Deleting contact status codes

You can delete contact status codes that you do not intend to use. However, you should not delete a contact status that is in use.

A contact status indicates the type of contact that was made, such as `Delivered`, `Undelivered`, or `Control`. Unica Campaign users specify a contact status when they configure a Call List or Mail List process. They configure a Track process to update contact statuses. Use the following procedure to delete a contact status.

1. Log in to the database management system that contains the Unica Campaign system table database.
2. Open the `UA_ContactStatus` table.
3. Delete the contact status row for any status that is not in use.
4. Save the table changes.

If necessary, see the documentation for your database management system for detailed instructions on modifying data in tables.

## Writing to contact history

To record contact history, users configure one or more contact processes, such as a Call List or Mail List, then run a flowchart in production (not test) mode. Contact history is written to the tables related to the audience levels that are used in the flowchart.



**Note:** The settings discussed in this topic do not affect Unica Deliver and Unica Interact. Those products use their own ETL processes to extract, transform, and load data into the Unica Campaign contact and response history tables.

The ability to write to contact history depends on the contact history logging options, which administrators can use to either allow or prohibit logging. These global configuration settings affect the contact processes as well as the Track process:

- The **logToHistoryDefault** configuration setting determines whether the **Log to contact history tables** option is checked or cleared by default in a contact or Track process box. If **logToHistoryDefault** is enabled, **Log to contact history tables** is checked by default, meaning contact history updates are allowed.
- The **overrideLogToHistory** configuration setting controls whether users with the appropriate permissions can change the **Log to contact history tables** setting when they configure a contact or Track process.

To make all flowchart production runs always write to contact history, enable **logToHistoryDefault** and disable **overrideLogToHistory**.

When contact history is logged, offer and treatment history are written as well.



**Note:** If a process is configured to log contact history, but the process runs on a cell with no selected targets, history records are not written.

For more information, see the Unica Campaign User's Guide.

## Updating contact history

To update contact history that has already been recorded, users configure the Track process and run it in production mode. Updating contact history is necessary, for example, to update contact status or add additionally tracked fields.

Consider the case where an updated contact list is received from a mail house with a list of targets who could not be contacted. In this case, you would use the updated list as the input to a Track process. When the flowchart that contains the Track process runs in production mode, contact history is updated for the tables related to the audience levels that are used.

The configuration settings **logToHistoryDefault** and **overrideLogToHistory** determine whether contact history can be updated.

Depending on the configuration settings, users may be able to check or clear the **Log to contact history and tracking tables** option when they configure the Track process.



## Clearing contact history

Users can clear the contact history generated by a contact process while configuring it. They are also prompted to select run history options when re-running processes or branches that have existing contact history, as these types of runs do not increment flowchart Run IDs.

Users can clear all contact history generated by that particular process, a particular run instance (identified by run date and time), or all contacts made within a specified contact date range. The appropriate records are then permanently deleted from the contact history table for the audience level. The next time the flowchart is run, the contact history is replaced rather than appended in the contact history table.

For more information, see the Unica Campaign User's Guide.

## Default contact status codes

Unica Campaign is delivered with the following contact statuses, defined in the `UA_ContactStatus` table.

**Table 25. Default contact status codes**

Contact-StatusID	Name	Description	Contact-StatusCode	Counts-AsContact	Defaults
1	Campaign Send	<NULL>	CSD	1	0
2	Delivered	<NULL>	DLV	1	1
3	Undelivered	<NULL>	UNDLV	0	0
4	Control	<NULL>	CTRL	0	2

# Chapter 8. Response history administration

Before you begin working with response history, you should read the audience level administration topics and set up required audience levels.

Response history is stored in the Unica Campaign system database in separate tables by audience level. Therefore, you must set up audience levels before you work with response history.

For basic concepts about contact and response history and information about setting up flowcharts to use the Response process, see the Unica Campaign User's Guide.

## Response history and response types

Response history is the historical record of responses to campaigns, either by targeted respondents or by hold-outs (members of control groups who perform the desired action despite not having been contacted). Response types are the specific actions that you are tracking in a campaign.

For more information about response history and instructions on how to design flowcharts to record responses, see the Unica Campaign User's Guide.

### What are response types?

Response types are the actions that you are tracking, such as click-through, inquiry, purchase, activation, use, and so on. Each response type is represented by a unique response code. Response types and codes are defined globally in the `UA_UsrResponseType` table, and are available for all offers, although not all response types are relevant for all offers. For example, you would not expect to see a click-through response type for a direct mail offer.

Unica Campaign is delivered with a set of default response types. As an administrator, you can add more response types.

Adding response types is explained in the Unica Campaign Administrator's Guide. For information about using and tracking response types, see the Unica Campaign User's Guide.

### How does response history relate to audience levels?

Unica Campaign records and maintains a separate response history for each audience level that is defined. Each audience level has its own related response history table in the Unica Campaign system database, as well as a related Unica Campaign system table.

### How does response history relate to database tables?

Response history tables, which must exist in the Unica Campaign system database, store historical responses for each audience level.

The `Customer` audience level is delivered by default, and the history of responses from customers could be stored in the `UA_ResponseHistory` in the Unica Campaign system database.

If you create an additional audience level, you must create the response history table for it in the Unica Campaign system database.

After you create the table in the Unica Campaign system database for a new audience level, you must map the new table to the Unica Campaign system table for the audience level's response history, which is automatically created when you create the audience level.

### Foreign key constraints in response history tables

Each response history table that you create for a new audience level must have a foreign key constraint on the `UA_Treatment` table's `TreatmentInstID` field. For details on how to set up this constraint, see the DDL file that creates the system tables.

## Action tables

An action table is an optional database table or file containing response data that is collected after offers are presented to customers.

An action table is audience level-specific. Typically, you create one action table for each audience level in Unica Campaign.

The action table can then serve as the source data of the input cell for the Response process in a campaign flowchart. Unica Campaign reads from the action table, and if a match is found between the relevant attributes and/or response codes, Unica Campaign populates the response history tables.

Using an action table is a best practice to ensure that sufficient information about target responses is recorded.



**Important:** Administrators must ensure that any action table that is used for response tracking is locked during response processing. Administrators must also clear rows after each Response process run to ensure that responses are not credited multiple times. For example, you can use Unica Campaign to run SQL after the Response process to purge the action table.

### What does an action table contain?

An action table includes data such as customer identification, response codes, and attributes of interest. Depending on how responses are tracked in your organization, the responses could be directly related to transactional data, such as purchases or contracts and subscriptions.

Each row of the action table represents a single event, which must minimally contain the audience IDs, the response type, and the response date. An action table typically includes one or more response codes (campaign, cell, offer, or treatment codes) and one or more standard or custom offer attributes for inferred response tracking (for example, the purchased product or service). Any field that is populated in an event is used to match against possible treatments that have that offer attribute. Any fields that are NULL are ignored.

The best practice is to use an action table that combines all responders and response types.

## Where are action tables located?

The decision where to put action tables must be made on a case-by-case basis and is usually done as part of your initial implementation.

When action tables are located in the user data mart, you can easily populate the table from other data mart tables, make joins, and perform similar database operations. However, you must ensure that you have permission to purge the action tables after each Response process run.

If your response logic is fairly simple (for example, if the action table is already populated with ETL routines and you only need to read the data from the table), then you may choose to locate the action table with the Unica Campaign system tables.

The Unica Campaign system tables include a sample action table for the `Customer` audience level, called `UA_ActionCustomer`, which administrators can customize as needed. The sample table contains several columns which you can use in Response tracking, such as the `CustomerId`, `Response` and `Tracking` codes.

## Sample action table (UA\_ActionCustomer)

The Unica Campaign system tables include a sample action table for the `Customer` audience level, called `UA_ActionCustomer`. The fields in this table are intended as examples of fields that may be useful for generating response history. Administrators can customize the table as needed. Typically, each audience level in Unica Campaign has its own action table, which is used for response tracking.

**Table 26. Sample UA\_ActionCustomer table**

Column Name	Data Type	Length	Allow Nulls
CustomerId	bigint	8	No
ActionDateTime	datetime	8	No
ResponseChannel	varchar	16	Yes
CampaignCode	varchar	32	No
OfferCode	varchar	64	No
CellCode	varchar	64	No
TreatmentCode	varchar	64	No

**Table 26. Sample UA\_ActionCustomer table (continued)**

Column Name	Data Type	Length	Allow Nulls
ProductID	bigint	8	No
ResponseTypeCode	varchar	64	Yes

## Creating response history tables for new audience levels

When you create a new audience level, you must create a table in the Unica Campaign system database to store response history for targets in that audience level.

When you create this table, you should also create an index on it to improve performance. For example, if you create the `INDIV_ResponseHistory` table for a new `Individual` audience level, you could create an index as follows:

```
INDEX XIE1INDIV_ResponseHistory ON INDIV_ResponseHistory ( IndivID )
```

After you create the response history table for a new audience level, you must map it to the Unica Campaign system table for the audience level response history.

## Setting the number of days after an offer expires to record responses

The response history tables can record whether responses were received before or after the expiration date of the specific offer version. This functionality depends on the configuration property `allowResponseNDaysAfterExpiration`.

To perform this task, you must have appropriate permissions to use Unica Platform. For information, see the *Unica Platform Administrator's Guide*.

1. Choose **Settings > Configuration** .
2. Go to **Campaign | partitions | partition[n] | server | flowchartConfig**.
3. Set the value for **AllowResponseNDaysAfterExpiration** to the desired number of days. The default value is 90 days.

## Adding response types

Response types are defined in the `UA_UsrResponseType` table in the Unica Campaign system database.

Unica Campaign includes a set of default response types. If the default response types are not sufficient, administrators can define additional response types . For more information, see [Default response types \(on page 158\)](#).

1. Log in to the database management system that contains the Unica Campaign system database.

See the documentation for your database management system for detailed instructions on modifying data in tables.

2. Open the `UA_UsrResponseType` table.
3. Add one row for each response type that you want to add:

- a. Enter a unique `ResponseTypeID`.
- b. Enter a `Name`.
- c. Optionally, enter a `Description`.
- d. Enter a unique `ResponseTypeCode`.
- e. In the `CountsAsResponse` column, enter `1` if the type represents a successful response, `0` if it does not count as a response, or `2` if it represents a reject.

The `CountsAsResponse` values are mutually exclusive for each response type. In other words, the same response type cannot be counted both as a response and a reject.

- f. In the `IsDefault` column, enter `1` for the response type that you want to be the default. Ensure that only one row has the value `1` in this column. All other rows must have the value `0`.

4. Save the table changes.
5. Remap the `UA_UsrResponseType` system table.



**Note:** If Unica Deliver offer integration is enabled and the response type originates in Unica Deliver: To support ETL of Unica Deliver response types, the response types must be defined in the Unica Deliver `UACE_ResponseType` table as well as the Unica Campaign `UA_UsrResponseType` table. Then the response types must be mapped in the `UA_RespTypeMapping` table.

## Default response types

New installations of Unica Campaign include the following response types, which are defined in the `UA_UsrResponseType` table. Upgrades include all response types except 9, 10, 11, which must be added manually if you plan to use Unica Deliver offer integration.

The `ResponseTypeID` and `ResponseStatusCode` must be unique. Do not change the supplied values for the default response types.

For `IsDefault`, only one row can be set to `1`. All other rows must be `0`.

The `CountsAsResponse` values are mutually exclusive for each response type. In other words, the same response type cannot be counted both as a response and a reject. Valid values are:

- 0 - does not count as a response
- 1 - counts as a positive response
- 2 - counts as a negative response

**Table 27. Default response types**

Response-TypeID	Name	Description	Response-StatusCode	Counts- As-Response	IsDefault
1	Explore	<NULL>	EXP	0	0
2	Consider	<NULL>	CON	0	0
3	Commit	<NULL>	CMT	1	0
4	Fulfill	<NULL>	FFL	0	0
5	Use	<NULL>	USE	0	0
6	Unsubscribe	<NULL>	USB	0	0
7	Unknown	<NULL>	UKN	1	1
8	Reject	<NULL>	RJT	2	0
9	Link Click*	<NULL>	LCL	1	0
10	Landing Page*	<NULL>	LPA	1	0
11	SMS Reply Message*	<NULL>	SRE	1	0

\*Response Types 9, 10, and 11 are intended for Unica Deliver offer integration. For new installations, these response types are added by default. Upgrades must add these response types manually if Unica Deliver offer integration will be used, then map them in `UA_RespTypeMapping`. Note that Landing Page and SMS Reply Message are not populated by the ETL process at this time.

## Logging response history

To log response history, users configure the Response process. Then, when the flowchart is run, response history is written to the table(s) related to the audience-level(s) used in the flowchart.

For more information, see the Unica Campaign User's Guide.



## Chapter 9. Monitoring and controlling flowchart runs

Choose **Campaign > Monitoring** and use the **All monitored runs** page to view the status of all active flowcharts and suspend, resume, or stop flowchart runs.

Operational monitoring tracks Unica Campaign flowcharts that are run from both the GUI (manual and scheduled runs) and the `unica_svradm` command-line utility. It does not keep track of session flowchart runs.

### Configuring operational monitoring

You must configure monitoring appropriately for your environment, including setting parameters for how long monitoring information is stored and displayed for historical flowchart runs. Additionally, ensure that security permissions are set appropriately.

1. Choose **Settings > Configuration**.
2. Open the **Campaign | monitoring** category and set the properties.
3. Ensure that security permissions are set appropriately for users:
  - To view the **All monitored runs** page, users must have the `Access monitoring page` OR `Perform monitoring tasks` permission.
  - Only users with `Perform monitoring tasks` permission are allowed to suspend, resume, or stop flowchart runs. This permission allows users to control all displayed flowcharts, regardless of the normal access rights the user has for each individual flowchart. Do not give this permission to users unless you intend to give them the right to suspend, resume, and stop any running flowchart.

For more information, see the *Unica Platform Administrator's Guide*.

### Using the **All monitored runs** page to control flowchart runs

Use the **All monitored runs** page to view, stop, suspend, or resume running flowcharts.

The ability to access the **All monitored runs** page and use the action buttons are determined by security permissions. See [Configuring operational monitoring \(on page 161\)](#).



**Note:** You can also Pause, Continue or Stop a flowchart from the Run menu on a flowchart page. The Pause and Continue actions are only available from the flowchart Run menu. For details, see the *Unica Campaign User's Guide*.

1. Choose **Campaign > Monitoring**.

The **All monitored runs** page groups active flowcharts by the campaigns that they belong to. The status for each flowchart is indicated in the Status column, and by a colored status indicator.

The action buttons that are available for each flowchart depend on the flowchart status. Additionally, they require the appropriate security permissions.

2. Optional: Click the name of a campaign to view the Summary of the associated campaign.

3. Optional: Click the name of a flowchart to view it in Read Only mode.
4. See the following topics for instructions on how to stop, suspend, or resume a run.


## Refreshing the **All monitored runs** page display

Use **Refresh** to update the **All monitored runs** page to ensure that you are viewing current operational details.

1. Choose **Campaign > Monitoring**.
2. Click **Refresh** at the top right. The page refreshes with current data.

## Stopping a running flowchart

You can perform the Stop action on a running flowchart.


1. Choose **Campaign > Monitoring**.
2. On the All monitored runs page, locate the flowchart that you want to stop.
3. Click the Stop button  next to the flowchart status.

The flowchart stops. Its status changes to **Stopped** and the status indicator color changes to red.

## Suspending a running flowchart

You can perform the Suspend action only on a running flowchart.

When you Suspend a flowchart, the run process ends, and system resources are released. A placeholder remains so that you can resume running the flowchart at the point at which you suspended it. This is different from pausing a flowchart (from the flowchart Run menu). When you pause a flowchart, the process remains, and does not release system resources (such as memory).

1. Choose **Campaign > Monitoring**.
2. On the All monitored runs page, locate the flowchart that you want to suspend.
3. Click the Suspend button  next to the flowchart status.

The suspend process starts. The flowchart status changes to **Suspending** and the color of the status indicator changes to yellow. In **Suspending** status, you cannot perform any actions on a flowchart.




**Note:** It may take a while to successfully suspend a running flowchart, because it must wait until running process boxes reach a state where they can be safely saved and resumed.

When the suspend process completes, the flowchart status changes to **Suspended**; the color of the status indicator remains yellow.

## Resuming a suspended flowchart

You can resume a suspended flowchart run. The resumed flowchart run restarts and continues running from at the point at which it was suspended.

1. Choose **Campaign > Monitoring**.
2. On the All monitored runs page, locate the suspended flowchart.
3. Click the Resume button  next to the flowchart status.

The flowchart resumes running. Its status changes to **Running** and the color of the status indicator changes to green.

## Flowchart states and valid actions

The following table describes the valid flowchart states and the actions that are available for each status on the **All monitored runs** page (**Campaign > Monitoring**).

The flowchart status reflects the status of the last run.



**Note:** If a user runs a flowchart and one branch succeeds, but another process in that flowchart but not in that branch has failed, the flowchart status is **Failed**.

**Table 28. Flowchart states and actions**


This table describes flowchart states and actions.

Status (color)	Description	Valid actions
Running (green)	The flowchart is running.	<ul style="list-style-type: none"> <li>• Suspend</li> <li>• Stop</li> </ul>
Paused (yellow)	<p>The flowchart was paused during running from the flowchart Run menu. (You cannot pause a flowchart from the <b>All monitored runs</b> page.)</p> <p>When a flowchart is paused, the process remains intact but stops processing, guaranteeing that no work is lost when the flowchart run is continued. Note that with the Pause action, system resources are not released (CPU utilization stops, but memory is not freed).</p> <p>You can continue running a paused flowchart from the flowchart Run menu.</p>	None from the <b>All monitored runs</b> page (Run>Continue from the flowchart)

**Table 28. Flowchart states and actions**

This table describes flowchart states and actions.

(continued)

Status (color)	Description	Valid actions
	For details about pausing and continuing a flowchart run, see the Unica Campaign User's Guide.	
Suspending (yellow)	The flowchart "Suspend" action has been initiated from the <b>All monitored runs</b> page and the flowchart is transitioning to this status.	None
Suspended (yellow)	<p>The flowchart Suspend action has completed and the flowchart is in a suspended state. The process is shut down and system resources have been released; a placeholder remains to enable restarting the flowchart run at the point at which it was suspended.</p> <p>You can resume running a suspended flowchart using the Resume button on the <b>All monitored runs</b> page.</p> <p> <b>Note:</b> Running process boxes that can be rerun from the beginning, resulting in the same net behavior, are immediately stopped when the Suspend command is issued, and any partially completed work is lost. These process boxes will be re-run when the flowchart run is resumed.</p>	<ul style="list-style-type: none"> <li>• Resume</li> </ul>
Succeeded (light blue)	The flowchart run completed successfully, without any errors.	None
Stopped (red)	The flowchart run was stopped either by a user from the flowchart Run menu or due to an error (that is, one or more process boxes in the flowchart encountered an error). For details about stopping a flowchart from the flowchart Run menu, see the Unica Campaign User's Guide.	None

**Table 28. Flowchart states and actions**

This table describes flowchart states and actions.

(continued)

Status (color)	Description	Valid actions
Failed (red)	The run failed due an unhandled error or a server error (that is., the flowchart server process exited unexpectedly).	None

# Chapter 10. Dimension hierarchy administration

A dimension hierarchy is a data construct that groups data into bins based on value ranges. Dimension hierarchies are the basis for a variety of reports.



**Note:** When dimension hierarchies are used to create cubes, use the Cube process to create dynamic data cubes from a flowchart in the Sessions area of the application.

## What is a dimension hierarchy?

A dimension hierarchy is a data construct used to group data into bins based on value ranges. A dimension hierarchy can contain multiple levels, each of which has its own set of bins. The bins in each lower level must roll up neatly into bins in higher levels.

For example, the Age dimension hierarchy could have two levels, Lowest level and Rollups. Customers are grouped in bins at each level:

**Lowest level:** (21-25), (26-30), (31-35), (36-45), (45-59), (60+)

**Rollups:** Young (21-35), Middle (36-59), Older (60+)



**Note:** You cannot split a lower level bin (for example, bin 26-30 above) and divide individuals aged 26-27 into "young" and 28-30 into "middle", when rolling up to a higher level. Any single bin in a lower level must fall completely within a higher level bin. If you wanted to actually define "young" as those aged 21-27, you would need to create separate bins (for example, 26-27 and 28-30) in the lower level so they could roll up into "young" and "middle," respectively.

Other commonly specified dimension hierarchies are time, geography, product, department, and distribution channel. However, you can create any kind of dimension hierarchy that relates to your business or campaign.

## Why use dimension hierarchies?

As the building blocks of cubes, dimension hierarchies are the basis for a variety of reports that can be used for data exploration, quick counts, or as a basis for targeting campaigns.

Cubes can pre-aggregate counts or simple calculations (sum, min, max, mean, standard deviation) of numeric fields (for example, total sales across all products at increasing aggregation levels, cross-tabular analysis of expenses versus sales by geography, and so forth).

Dimension hierarchies are also available as a means of selecting directly from strategic segments (without requiring any cubes to be built or to work from a cross-tab report).

Unica Campaign supports:

- Dimensions that are comprised of an unlimited number of levels and elements
- Data points built as input to customer analytic reporting and visual selection
- Rollups into unlimited number of categories to support drill-down capability

## About dimension hierarchies and cubes

You use dimension hierarchies to create dynamic data cubes, precalculated two- or three-dimensional aggregations of customer data built on a strategic segment.

Cubes are used for data exploration or visual selection, as you have the ability to drill through the data and use the resulting set of customers as a new cell in a flowchart.

For details about cubes, see the Unica Campaign User's Guide.

## About dimension hierarchies and database tables

When you create a dimension hierarchy in Unica Campaign, you map it to a table in a database or a flat file.

The table must contain columns for:

- The dimension name
- Each level in the dimension hierarchy
- The raw SQL or Unica expression that defines the audience entities in the bin
- The data source

For example, the `Age` dimension hierarchy has three levels. The first level is `All ages`, followed by the two levels shown in the two levels of the following list:

- Under 30
  - Under 20
  - 20 to 25
  - 26 to 30
- 30 to 50
  - 30 to 40
  - 41 to 50
- Over 50
  - 51 to 60
  - Over 60

This dimension hierarchy is based on the following database table:

**Table 29. Dimension hierarchy database table**

<b>Dimension- Name</b>	<b>Dim1Name</b>	<b>Dim2Name</b>	<b>Dim3Name</b>	<b>Expression</b>	<b>Datasource</b>
MemberAge	All Ages	Under 30	< 20 Years Old	age < 20	Your datamart
MemberAge	All Ages	Under 30	20 - 25 Years Old	age between 20 and 25	Your datamart
MemberAge	All Ages	Under 30	26 - 30 Years Old	age between 26 and 30	Your datamart
MemberAge	All Ages	30 - 50 Years	30 - 40 Years Old	age between 31 and 40	Your datamart
MemberAge	All Ages	30 - 50 Years	41 - 50 Years Old	age between 41 and 50	Your datamart
MemberAge	All Ages	Over 50	51 - 60 Years Old	age between 51 and 60	Your datamart
MemberAge	All Ages	Over 50	Over 60	age > 60	Your datamart

## Dimension hierarchy design guidelines

You should consider the following things when designing dimension hierarchies.

- How dimensions relate to each other (for example, Age/Geography/ Timespan).
- Level of detail for each dimension and cube.
- Dimensions are not limited to a single cube; they can be used in many cubes.
- Dimensions must roll up cleanly across boundaries, so elements must be mutually exclusive and not overlap.

## Managing dimension hierarchies

A dimension hierarchy is a data construct that groups data into bins based on value ranges. Dimension hierarchies are the basis for a variety of reports. Administrators can create and edit dimension hierarchies.



## Creating dimension hierarchies

After you define a dimension hierarchy in an external table or flat file, you can create dimension hierarchies in Unica Campaign.

Before you can create a dimension hierarchy in Unica Campaign, you or an consulting team must create the dimension hierarchy definition in a database table in your data mart or in a delimited or fixed-width flat file.

This is an operation external to Unica Campaign.

The lowest level of the dimension hierarchy must use either raw SQL or a pure Unica expression (no custom macros, user variables, or derived fields) to define the individual audience ID membership for each bin.

Follow these steps to create a dimension hierarchy in Unica Campaign.

1. Use one of these methods to open the **Dimension hierarchies** dialog:

- When editing a flowchart, open the **Admin** menu  and select **Dimension hierarchies**.
- From the **Campaign Settings** page, click **Manage dimension hierarchies**.

2. In the **Dimension hierarchies** dialog, click **New dimension**.

3. Enter the details for the new dimension hierarchy:

- The **Dimension name**
- A **Description**
- The **Number of levels** in the dimension hierarchy. This should correspond to the hierarchical levels in the table to which you are mapping this dimension hierarchy.
- If you will use this dimension hierarchy as the basis for a cube, ensure that **Elements are mutually exclusive** is checked (by default, this option is checked). Otherwise, you will receive an error when you use this dimension hierarchy to create a cube because the elements cannot overlap in a cube.

If you are creating a dimension hierarchy purely for use in selecting from a strategic segment, then it is permissible to disable this option and create overlapping definitions. It is recommended, however, that you create non-overlapping bins so that dimension hierarchies you create can be freely used both for building cubes as well as with strategic segments.

4. Click **Map table**.

The **Edit table definition** dialog opens.


5. To map your dimension hierarchy table to either a table in the database or a flat file containing the dimension hierarchy definition, follow the instructions in [Mapping a base record table to an existing database table \(on page 68\)](#).

After you finish mapping your dimension hierarchy, you return to the **Edit dimension** dialog, which now includes details for the new dimension hierarchy.

6. Click **OK**.
7. (Optional but recommended) You can store a dimension hierarchy for future use in a table catalog by clicking **Save**. If you store a dimension hierarchy you can retrieve it later for another use or share it with other users rather than recreating it.


## Loading stored dimension hierarchies

Dimension hierarchies are stored in table catalogs with any other mapped tables in the flowchart.

1. Use one of these methods to open the Dimension Hierarchies window:
  - When editing a flowchart, open the **Admin** menu  and select **Dimension hierarchies**.
  - From the **Campaign Settings** page, click **Manage dimension hierarchies**.
2. Click **Load**.
3. Select the table catalog containing the dimension hierarchies that you want to load.
4. Click **Load catalog**.

## Editing dimension hierarchies

You can change a dimension hierarchy name, description, levels, and table mapping.

1. Use one of these methods to open the Dimension Hierarchies window:
  - When editing a flowchart, open the **Admin** menu  and select **Dimension hierarchies**.
  - From the **Campaign Settings** page, click **Manage dimension hierarchies**.
2. You may need to load the dimension hierarchy that you want to edit.
3. Select the dimension hierarchy that you want to edit.
4. Click **Edit**.
5. Modify the following details:
  - The **Dimension name**
  - A **Description**
  - The **Number of levels** in the dimension hierarchy. This should correspond to the hierarchical levels in the database table to which you are mapping this dimension hierarchy.
  - If you are using this dimension hierarchy as the basis for a cube, ensure that **Elements are mutually exclusive** is checked (by default, this option is checked). Otherwise, you will receive an error when you use this dimension hierarchy to create a cube because the elements cannot overlap in a cube.
6. To modify the table mapping, click **Map table**.

The Edit Table Definition window opens.

7. Follow the instructions in [Mapping a base record table to an existing database table \(on page 68\)](#).
8. After you map the dimension, you return to the Edit Dimension window, which now includes details for the new dimension hierarchy.
9. Click **OK**.

You return to the Dimensions window.

10. (Optional but recommended) You can store your changes to any dimension hierarchies for future use in a table catalog by clicking **Save**.

## Updating dimension hierarchies


If the underlying data changes, you must manually update dimension hierarchies.

Unica Campaign does not support automatic updates of dimension hierarchies. If the underlying data changes, you must manually update your dimensions.



**Note:** Cubes are comprised of dimension hierarchies which are based on strategic segments, so you must update cubes whenever you update strategic segments.

1. Use one of these methods to open the Dimension Hierarchies window:

- When editing a flowchart, open the **Admin** menu  and select **Dimension hierarchies**.
- From the **Campaign Settings** page, click **Manage dimension hierarchies**.


2. You may need to load the dimension hierarchy that you want to edit.
3. Select the table catalog containing the dimension hierarchies that you want to update.
4. Click **Update**.

## Removing dimension hierarchies

Removing a dimension hierarchy makes it no longer available to strategic segments. Any cubes based on the dimension hierarchy will become unconfigured if they use a deleted dimension hierarchy.

If you delete a dimension hierarchy from a table catalog, it does not affect any existing flowcharts because those flowcharts contain a copy of the dimension hierarchy definition.

1. Use one of these methods to open the Dimension Hierarchies window:

- When editing a flowchart, open the **Admin** menu  and select **Dimension Hierarchies**.
- From the **Campaign Settings** page, click **Manage dimension hierarchies**.

2. You may need to load the dimension hierarchy that you want to update.
3. Select the dimension hierarchies you want to remove.
4. Click **Remove**.

You are prompted to confirm the removal.

# Chapter 11. Trigger administration

Unica Campaign allows you to define inbound and outbound triggers that can be used in all flowcharts in a partition.

Best practice is to provide trigger-level rights, such as **Run triggers**, only to restricted privileged users. Trigger permissions are available under Global Policy or by creating a custom role and adding that permission to it.

The trigger runs in the context of the user that runs the Unica Campaign listener. Therefore, the user from whose login the Unica Campaign listener is executed must have:

- restricted access to system files/ directories
- restricted execution rights on system level commands

Stored procedures are run with the same privileges as the user. Therefore, the Unica Campaign administrator should judiciously provide the add/edit stored procedure to users who need to perform this activity.



**Note:** For performance advantages, use the Unica Scheduler to send triggers to Unica Campaign. To learn more about the Scheduler, see the *Unica Platform Administrator's Guide*.

## What is an inbound trigger?

An inbound trigger is a message that is broadcast to one or more campaigns. You can configure a flowchart to "listen" for a particular trigger to start the execution of one or more processes.

Third-party systems typically send the trigger based on the occurrence of some external event.

## Why use an inbound trigger?

You can use an inbound trigger with a variety of events to start a process in Unica Campaign.

Some examples are:

- A database update triggers the recomputation of all strategic segments (for example, high, medium, and low value customer classifications based on the most recent purchase activity).
- A predictive model updating its scores in the database triggers a run of an acquisition campaign that is waiting for the latest scores.
- A third-party scheduling tool is used to schedule and trigger the execution of flowcharts.
- The run completion of Optimization session triggers the execution of the participating campaigns to retrieve and process their optimized results.

## Inbound triggers and the Schedule process

When configured to do so, the Schedule process listens for inbound triggers and executes when one is broadcast.

## What is broadcasting?

Broadcasting is the process of notifying all flowcharts in Unica Campaign, specific campaigns, or specific flowcharts that an inbound trigger has executed. Schedule processes configured to listen for that inbound trigger then execute.

To send an inbound trigger to a campaign or flowchart, you must broadcast the triggers to Unica Campaign using the trigger utility, `CAMPAIGN_HOME/bin/unica_actrg.exe`.

## What is an outbound trigger?

An outbound trigger is the execution of a command, batch file, or script that takes place after a flowchart or process is run. You can define triggers to perform virtually any action, such as opening an application, sending an email, or running a program.

Unica Campaign can execute an outbound trigger when a Schedule, Call List, or Mail List process executes. For example, when a Call List process completes, an outbound trigger can send an automatic email informing a manager that a list of contacts is ready.



**Note:** Triggers execute on completion of test runs as well as of production runs.

Unica Campaign can also automatically execute an outbound trigger when a flowchart runs. You can configure different triggers for when the flowchart completes successfully, or fails.

An outbound trigger can be synchronous or asynchronous.

## Synchronous outbound triggers

When Unica Campaign runs an outbound trigger synchronously, the process that called it waits for the executed command to complete and return with a success or failure status.

In other words, the flowchart does not continue running until the results of the trigger are returned. If the trigger fails, as indicated by a non-zero return value, the process box does not continue processing and indicates an error (with a red X) and an appropriate error message.

Synchronous execution is useful when the flowchart is waiting for an external process to complete its work before continuing. For example, a synchronous outbound trigger could run third-party predictive model scores in real-time, and the flowchart would wait until it completed before selecting from the updated model scores.

To make an outbound trigger synchronous, put a question mark (?) after the trigger name when you specify the trigger in the process configuration. For example:

```
EmailUpdate ?
```

## Asynchronous outbound triggers

When an asynchronous outbound trigger executes, the flowchart processing continues immediately; the process that called the trigger does not wait for it to succeed or fail.

To make an outbound trigger asynchronous, you do not have to add a termination character. However, to ensure that it is explicitly understood that the trigger is asynchronous, you can put an ampersand (&) after the trigger name when you specify the trigger in the process configuration. For example:

```
EmailUpdate &
```

## Why use an outbound trigger?

Outbound triggers may be useful in a variety of cases where you want to execute an action related to, but external to, a campaign.

Some typical examples of useful outbound triggers include:

- Sending an email notification upon completion of a campaign flowchart;
- Sending an email notification or performing some other task if a flowchart fails;
- Running a third-party modeling tool, such as SAS, to generate real-time results inline with the flowchart logic;
- Running a Unix shell script to send an output file by FTP after the file has been created;
- Launching a customer database update;
- Launching or triggering another flowchart.

## Return values for outbound triggers

Programs executed by an outbound trigger should return 0 on success, and a non-zero value on failure.

## How are triggers defined?

You define triggers when editing a flowchart. A trigger that you define in one flowchart is available to all flowcharts in the same partition.

The executable file for a trigger must be stored in the `CAMPAIGN_HOME/partitions/partition_name` directory. You can create a subdirectory, `triggers`, in this location, or use other subfolders as desired.

## Creating and managing triggers

You can create inbound and outbound triggers and organize them in folders.

### Creating triggers

You can define inbound and outbound triggers that can be used in all flowcharts in a partition.

You must have permissions to create triggers.

1. When editing a flowchart, open the **Options** menu  and select **Stored triggers**.

The Stored Trigger Definitions window opens.

2. Click **New Item**.

The data fields for the new trigger appear on the right of the window.

- Optionally, select a folder to save the trigger to in the **Save under** list.



**Note:** The folder location governs which users can access the trigger, based on the folder's security policy.

- Enter a name for the trigger in the **Name** field.
  - You cannot use spaces in the string, but you can use underscores (\_).
  - This name must be unique within the folder where you save it.
- If you are creating a trigger in the top-level folder, select a security policy, or keep the default.
- Optionally, enter a description of the trigger in the **Note** field.

You can provide a free-form text description of the trigger for documentation purposes. You also may want to keep a modification history of who modified the trigger, when, and what changes were made.

- In the **Command** field, you can enter path to the trigger file. It can be the path relative to the current partition root and file-name of the executable file on the Unica Campaign server. Or you can enter full file path to the location of the trigger file. You can also click **Browse** to select an executable file from within the current partition.

If you are creating an outbound trigger, to make it synchronous, terminate the command with a question mark (?).

To make the trigger asynchronous, do not terminate the command with a special character, or use an ampersand (&).

- Click **Save** and **Close**

## Editing or moving triggers

You can change the name and descriptive note for a trigger or move it to another folder. If you change a trigger name, any processes referring to that trigger will become unconfigured and will be unable to run. You must edit each process to refer to the new trigger name.

You must have permissions to edit or move triggers.

- When editing a flowchart, open the **Options** menu  and select **Stored triggers**.

The Stored Trigger Definitions window opens, displaying all triggers defined in the current Unica Campaign partition.

- Locate and select the trigger to edit in the **Items list**.
- Click **Edit/Move**.

The data fields for the trigger appear on the right of the window.

4. Optionally, select a different folder in the **Save under** list.



**Note:** The folder location governs which users can access the trigger, based on the folder's security policy.

5. Optionally, change the trigger name in the **Name** field.
  - You cannot use spaces in the string, but you can use underscores (\_).
  - This name must be unique within the folder where you save it.
6. If you are modifying a trigger in the top-level folder, or moving a trigger to the top-level folder, select a security policy, or keep the default.
7. Optionally, modify the description of the trigger in the **Note** field.
8. Optionally, in the **Command** field, modify the path relative to the current partition root and file-name of the executable file on the Unica Campaign server. You can click **Browse** to select an executable file from within the current partition.

If you are creating an outbound trigger, to make it synchronous, terminate the command with a question mark (?).

To make the trigger asynchronous, do not terminate the command with a special character, or use an ampersand (&).

9. Click **Save** and **Close**.


If you renamed a trigger, edit each process to refer to the new trigger name.

## Deleting triggers

If you delete a trigger, any processes that refer to that trigger will become unconfigured and unable to run. You must edit each process to remove the reference to the deleted trigger.



**Note:** You must have permissions to delete triggers.

1. When editing a flowchart, open the **Options** menu  and select **Stored triggers**.
2. Locate and select a trigger in the **Items list**. The list shows all triggers that are defined in the current partition.
3. Click **Remove**.
4. Click **OK** to confirm the deletion.
5. Click **Close**.


Edit each process to remove the reference to the deleted trigger.

## Organizing triggers in folders

You can use folders to organize triggers.



You must have permissions to create folders for triggers.


1. Open a flowchart for editing.
2. Open the **Options** menu  and select **Stored triggers**.
3. Click **New folder**.
4. Name the folder and enter a descriptive note.
5. In the **Create under** list, select the folder in which to create the new folder, or select **None** to create a top-level folder.
6. If you are creating a top-level folder, select a security policy.

A sub-folder automatically inherits its security policy from its parent folder.

7. Click **Save**.

## Moving trigger folders

You can move trigger folders. You must have permissions to move trigger folders.


1. When you edit a flowchart, open the **Options** menu  and select **Stored triggers**.
2. Select a folder in the left pane.
3. Click **Edit/Move**.
4. In the **Create under** list, select the folder under which you want to move the selected folder, or select **None** to make the folder a top-level folder.
5. If you are moving the folder to the top level, select a security policy.

A subfolder automatically inherits its security policy from its parent folder.

6. Click **Save**.

## Editing trigger folders


You can change the name and descriptive note for a trigger folder. You must have permissions to edit trigger folders.

1. When you edit a flowchart, open the **Options** menu  and select **Stored triggers**.
2. Select a folder in the left pane.
3. Click **Edit/Move**.
4. Change the **Name** and **Note** for the folder.
5. Click **Save**.

## Deleting trigger folders

You can delete trigger folders.

You must have permissions to delete trigger folders.

1. When editing a flowchart, open the **Options** menu  and select **Stored triggers**.
2. Select a folder in the left pane.
3. Click **Remove**.

You are prompted to confirm the deletion.

4. Click **OK**.

## Setting up outbound triggers

You must have permissions to use triggers in a flowchart.

### Setting up a process to execute an outbound trigger

Three processes can execute outbound triggers when they run.

These processes are:

- Schedule
- Call list
- Mail list


In the **Schedule** process, you specify the trigger(s) to execute in the **Schedule** tab.

In the Call List and Mail List processes, you specify the trigger(s) to execute in the **Fulfillment** tab.

For information on configuring these processes, see the Unica Campaign User's Guide.

### Setting up a flowchart to execute an outbound trigger on success

You can set up a flowchart to execute selected triggers when the flowchart runs successfully, on both production and test runs.

1. When you edit a flowchart, click the **Admin** menu  and select **Advanced settings**.

The **Advanced Settings** window opens.

2. Select the trigger to execute in the **Send trigger(s) on flowchart success**.

To use multiple triggers, enter the name of each trigger, separated by a comma and a space.

3. Click **OK**.

### Setting up a flowchart to execute an outbound trigger on failure

You can set up a flowchart to execute selected triggers when the flowchart encounters an error when running, on both production and test runs.

1. When you edit a flowchart, click the **Admin** icon  and select **Advanced settings**.

The **Advanced Settings** window opens.

2. Select the trigger to execute in the **Send trigger(s) on flowchart run error**.

To use multiple triggers, enter the name of each trigger, separated by a comma and a space.

3. Click **OK**.

## Setting up inbound triggers

You must have permissions to use triggers in a flowchart.

### To set up inbound triggers

Use this procedure to set up inbound triggers.

1. Create the triggers within a flowchart, as described in [Creating triggers \(on page 174\)](#).
2. Configure the Schedule process in any flowcharts you want to run when receiving the inbound trigger, as described in [Configuring the Schedule process to run with an inbound trigger \(on page 179\)](#).
3. Use the Unica Campaign Trigger Utility `unica_actrg` (in the folder `Campaign_home/bin`) to broadcast triggers, as described in:
  - [Broadcasting a trigger to all flowcharts in a campaign \(on page 180\)](#)
  - [Broadcasting a trigger to specific flowcharts \(on page 180\)](#)
  - [Broadcasting a trigger to all campaigns \(on page 180\)](#)

### Configuring the Schedule process to run with an inbound trigger

To use an inbound trigger to run a flowchart, that flowchart must start with a Schedule process that is configured as described here.

- In the **Schedule to run** list, select **Custom run**.
- Check **Run on trigger(s)**.
- In the **Run on trigger(s)** field, enter the names of triggers that when broadcast will run the flowchart. Separate multiple triggers with a comma and a space.

The Schedule process may also be configured to execute based on other conditions. Configuring the trigger condition will additionally execute the subsequent processes when the specified triggers are received.



**Important:** For a flowchart to run upon receipt of an inbound trigger, it must have a Schedule process configured as described above, and it must be running. Running the flowchart puts the flowchart into a "waiting," or "listening", state so that the flowchart is ready to execute when the trigger is received. A flowchart that is not running when the trigger is broadcast will not execute.

For more information about configuring the Schedule process, see the Unica Campaign User's Guide.

## Broadcasting a trigger to all flowcharts in a campaign

You can send an inbound trigger to all flowcharts in a campaign.

Run the Unica Campaign Trigger Utility with the following syntax:

```
unica_actrg campaign_code trigger_name
```

For example:

```
unica_actrg C003 web_hit
```

If a flowchart in the specified campaign starts with a Schedule process configured to run when receiving a broadcast based on the `web_hit` inbound trigger, that flowchart runs when the broadcast trigger is received.

## Broadcasting a trigger to specific flowcharts

You can send an inbound trigger to all running flowcharts with the specified name.

Run the Unica Campaign Trigger Utility with the following syntax:

```
unica_actrg -n flowchart_name trigger_name
```

For example:

```
unica_actrg -n account_inquiry_flowchart web_hit
```

If the flowchart of the specified name starts with a Schedule process configured to run when receiving a broadcast based on the `web_hit` inbound trigger, that flowchart runs when the broadcast trigger is received.

## Broadcasting a trigger to all campaigns

Use this procedure to send an inbound trigger to all campaigns.

Run the Unica Campaign Trigger Utility with the following syntax:

```
unica_actrg * trigger_name
```

For example:

```
unica_actrg * web_hit
```

The trigger is broadcast to all flowcharts in all campaigns. If any flowchart starts with a Schedule process configured to run when receiving a broadcast based on the `web_hit` inbound trigger, that flowchart runs when the broadcast trigger is received.



**Note:** On UNIX™ servers, the asterisk must either be escaped (\\*) or enclosed in double quotation marks ("\*").

## Setting up the trigger utility on a remote Windows machine

You can configure a Windows machine to send triggers to a Unica Campaign installation on UNIX or Windows. For Windows installation, ensure that the listener port is open for incoming traffic on the Campaign listener machine. Follow these steps to set up the `unica_actrg` utility and required files on a remote Windows machine.

### 1. Obtain the required files:

<CAMPAIGN_HOME>\bin	<ul style="list-style-type: none"> <li>◦ If you have existing Linux environment. You will need to perform a dummy windows installation to get the required ddl's from Campaign/bin directory.</li> <li>◦ On remote windows machine - copy content of bin directory from actual installation.</li> </ul>
<CAMPAIGN_HOME>\conf	config.xml

To obtain the files, you can copy them from another Unica Campaign installation on Windows or run the Unica Campaign installer. If you obtain the files by running the installer and you want to remove the unnecessary files, copy the files required for the trigger utility to another location, then uninstall Unica Campaign. For more information, see UnicaCampaign Installation Guide.

2. Open a command prompt on the remote Windows machine.
3. If it is not already set, set the `CAMPAIGN_HOME` environment variable on the remote Windows machine. For example:

```
set CAMPAIGN_HOME=C:\HCL\Unica\Campaign
```

When you run `unica_actrg` remotely, specify the port and server name of the machine where the Unica Campaign listener is installed. If you have a clustered listener configuration, best practice is to specify the server and port of the master listener.

## Tokens supported by triggers

Tokens can be used in the command line of an outbound trigger to pass specific information from the running flowchart.

The following table lists the tokens supported by triggers and the processes where specific tokens are available.

**Table 30. Tokens supported by triggers**

Token	Description	Where Used
<AMUSER>	The Unica user name of the user running the flowchart.	Processes that support outbound triggers.

**Table 30. Tokens supported by triggers (continued)**

Token	Description	Where Used
<CAMPCODE>	The campaign code associated with the current campaign.	Processes that support triggers, trigger on failure, trigger on success.
<CONTACTLIST>	<p>Contact list specified in a contact process.</p> <p>If the Contact List is written to a file, the appropriate full path name and file name replace the trigger token.</p> <p>If the Contact List is written to a database table, the token is simply removed.</p>	<b>Call list</b> and <b>Mail list</b> processes.
<CONTACTLOG>	<p>The log of the particular contact process.</p> <p>When the Log is written to a file, the appropriate full path name and file name replace the trigger token.</p>	<b>Call list</b> and <b>Mail list</b> processes.
<FLOWCHARTFILENAME>	Full path name of a flowchart's .ses file	Processes that support outbound triggers.
<IXUSER>	The user name of the Unica Collaborate user.	Processes that support triggers, trigger on failure, trigger on success.
<OUTPUTTEMPTABLE>	<p>A token for use in raw SQL in pre- and post-processing under the Advanced window to create a temp table. For example:</p> <pre>Create &lt;OUTPUTTEMPTABLE&gt; as SELECT CustIDs from CustomerTable WHERE ...</pre>	<b>Select</b> process.
<OWNER>	The Unica Platform security user name of the user who created the flowchart.	Processes that support triggers, trigger on failure, trigger on success.
<PROCESSNAME>	The name of the current process box.	Processes that support triggers.
<PROCESSID>	The ID of the current process box.	Processes that support triggers.

**Table 30. Tokens supported by triggers (continued)**

Token	Description	Where Used
<SESSIONID>	The ID of the current flowchart.	Processes that support triggers, trigger on failure, trigger on success.
<SESSIONNAME>	The name of the current flowchart.	Processes that support triggers, trigger on failure, trigger on success.
<UserVar .UserVarName>	Any user variable value. The user variable must be defined in current flowchart.	Processes that support triggers, trigger on failure, trigger on success.

## Unica Campaign trigger utility syntax and options

The trigger utility (`unica_actrg`) supports the following syntax and options.

```
[ -p <port> [-S]] [-s <server_name>] [-v] [<campaign_code> | -n "<flowchart_name>"] "<trigger1>" "<trigger2>"...
```

The `unica_actrg` utility supports the following options.

**Table 31. Unica Campaign trigger utility options**

Parameter	Use
<code>-p &lt;port&gt;</code>	<p>The port where the listener is running.</p> <p>For a single-node listener configuration: Port and server are optional unless you run the trigger from a remote machine.</p> <p>For a clustered listener configuration: Port and server are optional unless you run the trigger from a remote machine. When running locally, the trigger automatically goes to the master listener. If you are running the trigger utility from a remote machine, best practice is to specify the server and port of the master listener.</p>
<code>-s &lt;server_name&gt;</code>	<p>The name of the listener server.</p> <p>For a single-node listener configuration: Port and server are optional unless you run the trigger from a remote machine.</p> <p>For a clustered listener configuration: Port and server are optional unless you run the trigger from a remote machine. When running locally, the trigger automatically goes to the master listener. If you are running the trigger utility</p>

**Table 31. Unica Campaign trigger utility options (continued)**

Parameter	Use
	from a remote machine, best practice is to specify the server and port of the master listener.
-v	Reports the version of the Unica Campaign Trigger Utility.
-S	If you use -p to specify a port, you may also include -S to establish an SSL connection.
<campaign_code>	The identifier of the campaign that contains all flowcharts that you want to run. This parameter cannot be used with the -n "<flowchart_name>" parameter.
-n "<flowchart_name>"	The name of the flowchart that you want to run. Because flowchart names are not necessarily unique, all flowcharts with this name receive the broadcast trigger. This parameter cannot be used with the <campaign_code> parameter.
"<trigger1>" "<trigger2>" ...	The name of the trigger to use. You must specify at least one trigger. You can optionally specify multiple triggers, separated by spaces.



# Chapter 12. Unica Campaign log files

Unica Campaign records information in several different log files.

By default, most log files are in the following locations:

```
<Campaign_home>/logs  
<Campaign_home>/partitions/partition[n]/logs
```

If you have a clustered listener configuration, additional log files are in the following locations:

```
<campaignSharedHome>/logs  
<campaignSharedHome>/partitions/partition[n]/logs
```

## Unica Campaign log file names and locations

Log files record information about the Unica Campaign web application, listeners, utilities, flowcharts, and operations.



**Note:** The `<campaignSharedHome>` that is mentioned in the following table is a shared location that was specified at installation time. It is configurable at `Campaign|campaignClustering|campaignSharedHome`. `<Campaign_home>` is the location where Unica Campaign is installed.

**Table 32. List of Unica Campaign log files**

Log file	Description	Default name and location
Flowchart logs	Each flowchart has its own log file, named <code>CampaignName_-CampaignCode_ Flowchart-Name.log</code> .	Single-node listener: <code>&lt;Campaign_home&gt;/partitions/partition [n]/logs/&lt;flowchart&gt;.log</code> Clustered listeners: <code>&lt;campaignSharedHome&gt;/partitions/partition [n]/logs/&lt;flowchart&gt;.log</code>
Web application log	Events generated by the Unica Campaign web application.	On the web application server: <code>&lt;Campaign_home&gt;/logs/campaignweb.log</code>
Unica Deliver ETL log	Events generated by the ETL process that coordinates Unica Deliver offer integration with Unica Campaign.	<code>&lt;Campaign_home&gt;/logs/ETL.log</code>
Import Error log	Generated if errors occur when a call is made from Unica Campaign to the Engage ImportAPI.	<code>&lt;Campaign_home&gt;/logs/ImportError.log</code>
Engage ETL log	Processed email events and the ETL to DtlcontactHist and Response history table in Campaign system Schema	<code>&lt;Campaign_home&gt;/logs/EngageETL.log</code>

**Table 32. List of Unica Campaign log files (continued)**

Log file	Description	Default name and location
Listener log	Events generated by the Unica Campaign listener (unica_aclsnr). In a clustered configuration, each listener has its own log file.	On the listener server: <code>&lt;Campaign_home&gt;/logs/unica_aclsnr.log</code>
Master listener log	Cluster-related events for activities related to load balancing, heartbeat, node selection, and failover. (Clustered listener configurations only.)	<code>&lt;campaignSharedHome&gt;/logs/masterlistener.log</code>
Campaign Server Manager log	Generated if errors occur when running the Unica Campaign Server Manager utility (unica_svradm).	On the listener server where the utility is run: <code>&lt;Campaign_home&gt;/logs/unica_svradm.log</code>
Cleanup utility log	Generated if errors occur when running the cleanup utility (unica_acclean).	On the listener server where the utility is run: <code>&lt;Campaign_home&gt;/logs/unica_acclean.log</code>
Session utility log	Generated if errors occur when running the Unica Campaign session utility (unica_acsesutil).	On the listener server where the utility is run: <code>&lt;Campaign_home&gt;/partitions/partition [n]/logs/unica_acsesutil.log</code>
Sessions log	Information about server connections when flowcharts are opened.	Single-node listener: <code>&lt;Campaign_home&gt;/partitions/partition [n]/logs/ac_sess.log</code>  Clustered listeners: <code>&lt;campaignSharedHome&gt;/partitions/partition [n]/logs/ac_sess.log</code>
UBX log	Events downloaded from UBX in event tables of Unica Campaign in Campaign System schema.	<code>&lt;Campaign_home&gt;/logs/UBX.log</code>
Web connections log	Information about user connections to the Unica Campaign system database. When a user logs into Unica Campaign, information is logged to the ac_web.log file.	Single-node listener: <code>&lt;Campaign_home&gt;/partitions/partition [n]/logs/ac_web.log</code>  Clustered listeners: <code>&lt;campaignSharedHome&gt;/partitions/partition [n]/logs/ac_web.log</code>
UBX tools log	Generated when RegisterEndPoint utility executes to register Unica Campaign as a UBX endpoint.	<code>&lt;Campaign_home&gt;/tools/UBXTools/ubx_tools.log</code>

## Flowchart logs

Each flowchart can write to its own log file whenever the flowchart is edited or run. Flowchart log files are useful for analyzing flowchart performance and database interactions.

The default file name for a flowchart log file is `<CampaignName>_<CampaignCode>_<FlowchartName>.log`.

The default location is `partitions/partition_name/logs` under `<Campaign_home>` (for a single listener node configuration) or `<campaignSharedHome>` (for a clustered configuration).

### Temporary file listing in the flowchart log

The flowchart log file contains all the information about temporary files that are created or deleted by the flowchart processes during flowchart execution. For example, while downloading data, while calculating derived fields, during bulk insert and so on. You can use this information to identify the flowchart that is creating temporary files of a large size and modify the flowchart logic if required.



**Prerequisite:** The `Campaign|unicaACListener|loggingLevels` property must be set to `High`.

To enable temporary file listing in the flowchart log, complete the following steps:


1. Open the flowchart in edit mode and click **Log options > Logging options**.
2. In the **Severity level** section, select **Debug**.
3. In the **Events** section, select **File operations (open, read, write, etc.)**

## Configuring flowchart logging

Administrators can configure logging for all flowcharts in a partition and optionally allow users to override the settings for individual flowcharts.


To perform this procedure, you must have **Administer configuration** page permission in Unica Platform.

Task	Instructions
Set the global configuration properties to determine how logging occurs for all flowcharts in a partition.	<ol style="list-style-type: none"> <li>1. Choose <b>Settings &gt; Configuration</b>.</li> <li>2. Set the properties under <code>Campaign partitions partition[n] server logging</code>.</li> </ol> <p>For example, you can enable or disable logging, set the logging level, specify which events are logged, and allow users to change the log file path.</p>
Set the administrative privileges to allow users to adjust logging options for individual flowcharts.	<ol style="list-style-type: none"> <li>1. Choose <b>Settings &gt; User roles and permissions</b>.</li> <li>2. Select a partition under the Campaign node.</li> <li>3. Click <b>Add roles and assign permissions</b>.</li> </ol>

Task	Instructions
	<ol style="list-style-type: none"> <li>4. On the <b>Properties for Administrative Roles</b> page, click <b>Save and edit permissions</b>.</li> <li>5. For <b>Logging</b>, check <b>Override flowchart log options</b>.</li> </ol>
(Optional) For troubleshooting purposes only, enable trace logging	<p>Trace logging is useful when working with Support. When you enable trace logging, trace events are written to both the listener log <code>unica_aclsnr.log</code> and the flowchart log <code>&lt;flowchart&gt;.log</code>.</p> <p>Trace events are identified by [T] in the log file.</p> <p>To enable trace logging:</p> <ol style="list-style-type: none"> <li>1. Add the following entry to the setenv.sh or setenv.bat file: <code>UNICA_AC-TRACE=Trace</code></li> <li>2. Restart the listener.</li> </ol> <p> <b>Note:</b> Trace logging can degrade performance so be sure to disable it (comment out the line and restart the listener) after you are done.</p>

All flowcharts in the partition now use the configured properties for logging.

However, users who are allowed to **Override flowchart log options** can change the logging options when editing

a flowchart. They can open a flowchart for editing and use the **Log options** menu  to adjust the logging options, including the Severity level and Events that are logged. The selected options apply only to the flowchart that is being edited. The selected options do not persist beyond the current session. The next time a user edits the flowchart, the logging options revert to the default settings.

If **AllowCustomLogPath** is enabled in the global configuration and users have the appropriate permissions, users can change the log file location when editing a flowchart by choosing **Log options > Change log path**.

If **enableLogging** is enabled in the global configuration and users have the appropriate permissions, users can turn logging on or off for individual flowcharts by checking or clearing **Enable logging** on the **Log options** menu.

## Viewing and analyzing flowchart log files

Each flowchart has its own log file, which records events during each flowchart and process run. You can analyze the log file to determine how a flowchart is performing and troubleshoot errors. The events and logging levels that are recorded are determined by the flowchart Logging options.

1. Open a flowchart for editing.

2. Open the **Log options** menu  and select **View log**.

The log file opens in a separate window.

3. To interpret the log file, see the examples in [Flowchart log file structure \(on page 189\)](#).


4. If the log file contains too much information (or not enough), use **Logging options** to adjust the severity level (Information, Warning, Error, Debug) and categories of events that get logged, then do a test run of a process and reexamine the log file. When you are done, revert to the default logging level to avoid performance issues.

5. If a log file gets too long, use **Clear log** to remove all existing entries. To back up the log file before clearing it, open it for viewing and copy the contents to another file.

## Flowchart log file structure

To analyze flowchart log files, it helps to understand the log file structure.

The following example illustrates the log file structure. To view or set logging options, open a flowchart for editing

and use the **Log options** menu  menu. The **Logging options** choice lets you adjust the logging level (Information, Warning, Error, Debug), specify which event categories to log, and include the Process ID in log entries.


Timestamp	PID	Level (I. W. E)	Category	Process name	Message body
04/20/2005 17:14:20.667	(1752)	[I]	[PROCESS]		SESSION_RUN_START
04/20/2005 17:14:20.797	(1752)	[I]	[PROCESS]	[Active]	Select PROCESS_RUN_START
04/20/2005 17:14:20.907	(1752)	[I]	[DB QUERY]	[Active]	Northwind (thread 000004B8): SELECT
04/20/2005 17:14:20.957	(1752)	[I]	[TABLE ACC]	[Active]	Northwind (thread 000004B8): Query completed;
04/20/2005 17:14:22.069	(1752)	[I]	[TABLE ACC]	[Active]	Northwind (thread 000004B8): Data retrieval
04/20/2005 17:14:22.089	(1752)	[I]	[PROCESS]	[Active]	Select N_RECORDS = 89
04/20/2005 17:14:22.099	(1752)	[I]	[PROCESS]	[Active]	Select PROCESS_RUN_DONE

The following example shows a portion of a flowchart log file. When you analyze a log file, it is helpful to identify where each process run starts and ends and to see the SQL that generated the database queries. Depending on your flowchart, you can also look for information about derived fields or other entities that you are interested in analyzing.



## Clearing a flowchart log file

If a flowchart log file gets too long, you can clear it to delete all entries in the log file. Users must have the appropriate Logging permissions to clear a log file. New entries will be written to the log the next time you run a process or flowchart.

1. Open a flowchart for editing.
2. Optional: Back up the log file before you clear its contents. The easiest way to back it up is to choose **Log options > View log**, copy the contents, and save it to another file.
3. Open the **Log options** menu  and select **Clear log**.
4. When prompted, confirm that you want to delete the contents of the log file.

## The Unica Campaign web application log

The web application log file (`campaignweb.log`) records events that are generated by the Unica Campaign web application.

The `campaignweb.log` file is located on the Unica Campaign web application server. The default file name and location is `Campaign_home/logs/campaignweb.log`.

Depending on the logging settings, there may be multiple historical Unica Campaign web application logs, each ending with an extension number, for example `campaignweb.log.1`, `campaignweb.log.2`, and so on.

To adjust the logging properties for `campaignweb.log`, modify the `campaign_log4j.properties` file, located by default in `Campaign_home/conf`.

## Configuring Unica Campaign web application logging

To adjust the logging settings for the Unica Campaign web application log file (`campaignweb.log`), modify the `campaign_log4j.xml` file.

1. Open the `campaign_log4j.xml` file in a text editor.

By default, the file is located at `Campaign_home/conf/campaign_log4j.xml`. If the file is not in the default location, you can find it in the location specified in the configuration property `Campaign | logging | log4jconfig`.

2. Use the comments in the `campaign_log4j.xml` file to determine how to adjust the logging settings for `campaignweb.log`.

For Example:

- You can adjust the logging level. Options include ALL (equivalent to Debug), HIGH (Information), MEDIUM (Warnings), or LOW (Errors).
  - You can specify whether to generate one or multiple web log files (`campaignweb.log.1`, `campaignweb.log.2`, `campaignweb.log.3`).
  - You can change the `campaignweb.log` path and file name. By default, the log file is located on the Unica Campaign web application server, at `Campaign_home/logs/campaignweb.log`.
3. Save the `campaign_log4j.xml` file.
  4. Restart the Unica Campaign web application.

### Important note to users who want to enable debug level logging to a specific user or a group of users:

Customers can set debug level logging to a specific user or a set of users. With this feature, the analysis of the debug log is handled with ease as it creates a separate log file for the particular user and it does not impact the system performance of the remaining users.

To enable this feature, users may have to adjust the logging settings of the Unica Campaign web application log file, by modifying the `campaign_log4j.xml` file. By default, the logging properties are loaded from the `campaign_log4j.xml` file. However, with this feature users can also configure the Unica Campaign web application logging properties by providing them in an XML file format.

By default, Unica Campaign installation would generate `campaign_log4j.xml` file at `./Affinium/Campaign/conf` location which would have the Unica Campaign logging properties in an XML format.

Loading of the XML file or the properties file is configurable and can be changed in the Unica Campaign web application under the **Configuration** settings. To access these properties, choose **Settings > Configuration**

#### log4jConfig

Campaign | logging

#### Description

The `log4jConfig` property specifies the location of the Unica Campaign log properties file, `campaign_log4j.xml`. Specify the path relative to the Unica Campaign home directory, including the file name. Use forward slashes (/) for UNIX™ and backslashes (\) for Windows™.

#### Default value

`./conf/campaign_log4j.xml`

Perform the following steps to configure the `Campaign_log4j.xml` file:

- Log in to Unica Campaign web application with an administrator user role.
- Navigate to **Configuration**.
- Navigate to **Affinium>Campaign> logging location**.
- Edit `log4jConfig` property which saves location of logging configuration file and provide correct path of XML format logging properties file.
- Click **Save Changes**.
- Restart the Unica Campaign web application.

Perform the following steps to configure the Unica Campaign web application in the `Campaign_log4j.xml` file:

- Locate the file specified by the **applications>Campaign>logging>** property. By default, the file would be at the following location at `Campaign_home/conf/campaign_log4j.xml`.
- Use the comments that are specified within this file to change the web application logging settings.
- Save the file and restart the Unica Campaign web application to change the file name and the location of the Unica Campaign web application log.



**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the Unica Platform Administrator's Guide.

- If customer needs user specific logging, then remove the following comments from the XML configuration file and restart the server.

```
<!--<appender name="USER_FILE" class="org.apache.log4j.RollingFileAppender">
  <param name="Encoding" value="UTF-8"/>
  <param name="File" value="user_campaignweb.log"/>
  <param name="MaxBackupIndex" value="50"/>
  <param name="MaxFileSize" value="1000KB"/>
  <param name="Threshold" value="DEBUG"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d{MM/dd/yyyy HH:mm:ss} %-5p %c{2} [%L] - %m%n"/>
  </layout>
  <filter class="com.unica.manager.logger.UserMatchFilter">
    <param name="stringToMatch" value="asm_admin"/>
  </filter>
</appender-->
```



```

</filter>
</appender> -->

<!-- <appender-ref ref="USER_FILE" /> -->

```

You can add multiple tags in the `Campaign_HOME/conf/campaign_log4j.xml` file to create per user log files. For each new entry, a new appender must also be added. By default, with this appender tag, the campaign web log is created in the application profile home location with name `user_campaignweb.log`. You can specify any valid name and path for the campaign web log file. For filtered and non-filtered users, the default logs like `campaignweb.log`, `Engage_ETL.log`, `UBX.log`, are also created in the application profile home location. You must specify the absolute/complete paths to generate all the log files in respective folders.

## The Unica Campaign and Unica Deliver ETL log file

The `ETL.log` file records events generated by the ETL process that coordinates Unica Deliver offer integration with Unica Campaign. The default file location is `Campaign_home/logs/ETL.log`

The Unica Campaign ETL process extracts, transforms, and loads offer response data from the Unica Deliver tracking tables into the Unica Campaign contact and response history tables. The ETL log file records the success, failure, and other statuses of events related to envelopes, treatments, and responses.

To adjust ETL logging behavior, modify the logging properties in the `campaign_log4j.xml` file. This is the same properties file that is used to configure the Unica Campaign web application log file. The location of the properties file is specified by **Settings > Configuration > Campaign > Logging**. The default location is `Campaign_home/conf`.

When the ETL log file size grows beyond 10MB, the ETL log file is rotated in the same way as the Unica Campaign web application log file. A number is appended to each successive log file, such as `ETL.log.1`, `ETL.log.2`, and so on. To adjust this behavior, modify the log4j properties file.

## Using log4j to configure web application and Unica Deliver ETL logging

The Unica Campaign web application and the Unica Deliver ETL process use the Apache `log4j` utility for logging configuration, debugging, and error information. Apache `log4j` is an open source Java™-based logging utility.

To configure logging for the Unica Campaign web application and the Unica Deliver ETL process, edit the `campaign_log4j.xml` file.

1. Open the `<Campaign_home>/conf/campaign_log4j.xml` file.

If the properties file is not in the `/conf` directory, look in the location that is specified in `Campaign|logging|log4jconfig`.

2. Adjust the property values in the properties file.

For information about changing property values, see the following sources:

- The comments in the `campaign_log4j.xml` file.
- The log4j documentation on the Apache web site: <http://logging.apache.org/log4j/1.2/manual.html>

3. Restart the Unica Campaign web application.

## Viewing and configuring Unica Campaign listener and master listener logs


The listener allows clients such as the Unica Campaign web application to connect to back-end analytical server processes. Each listener records events in its own log file. Additionally, if you have a clustered configuration, there is a master listener log file.


For single-node configurations:

The listener log file is on the listener server machine, at `<Campaign_Home>/logs/unica_aclsnr.log`.

For clustered configurations:

- Each listener generates its own log file on its own server machine, at `<Campaign_Home>/logs/unica_aclsnr.log`.
- Additionally, cluster-related events related to load balancing, heartbeat, listener node selection, and failover are recorded in a master listener log file: `<campaignSharedHome>/logs/masterlistener.log`. The `<campaignSharedHome>` is a shared location that was specified at installation time. It is configurable at `Campaign | campaignClustering | campaignSharedHome`.

Task	Action	Notes®
To view a listener log file	<p>On the Unica Campaign server, choose <b>Settings &gt; Campaign Settings &gt; View System Log</b>.</p> <p> <b>Note:</b> If there are multiple partitions, this option is not available for security reasons.</p> <p>You can also go to any machine where a listener is installed and open <code>&lt;Campaign_Home&gt;/logs/unica_aclsnr.log</code> in a text editor.</p>	The log opens in a new browser window. Events that occur after you open the log file are not listed.
To view the master listener log (clustered configuration only)	On the master listener server, open <code>&lt;campaignSharedHome&gt;/logs/</code>	If you are not sure which machine is the master listener, look at <b>Campaign</b>

Task	Action	Notes®
	<code>masterlistener.log</code> with a text editor.	<code>unicaACLlistener</code>   <code>node [n]</code>   <code>masterListenerPriority</code> .
To configure logging for each listener node	<ol style="list-style-type: none"> <li><b>Settings &gt; Configuration.</b></li> <li>Go to <b>Campaign   unicaACLlistener</b> and adjust the settings that begin with "log".</li> </ol>	Depending on how you configure logging, each listener may generate one log file or multiple sequentially-named log files, such as <code>unica_aclsnr.log.1</code> , <code>unica_aclsnr.log.2</code> , and so on.
To configure master listener logging (clustered configuration only)	<ol style="list-style-type: none"> <li><b>Settings &gt; Configuration.</b></li> <li><b>Campaign   campaignClustering</b></li> </ol>	To perform this task, you must have <b>Administer Configuration page</b> permission in Unica Platform.
To enable trace logging for diagnostic purposes	<ol style="list-style-type: none"> <li>Add the following entry to the <code>setenv.sh</code> or <code>setenv.bat</code> file: <code>UNICA_ACTRACE=Trace</code></li> <li>Restart the listener.</li> </ol>	<p>Trace logging is useful when working with Support. When you enable trace logging, trace events are written to <code>unica_aclsnr.log</code> and also to the flowchart log file <code>&lt;flowchart&gt;.log</code>.</p> <p>Trace events are identified by [T] in the log file.</p> <p> <b>Note:</b> Trace logging can degrade performance so be sure to disable it (comment out the line and restart the listener) after you are done.</p>

## The Unica Campaign Server Manager log

The Unica Campaign Server Manager log file (`unica_svradm.log`) is generated if an error occurs when running the `unica_svradm` utility.

This log file is located on the listener server where the utility is run: `<Campaign_home>/logs/unica_svradm.log`.

## The session utility log

The Unica Campaign session utility log file is generated if an error occurs when running the `unica_acsesutil` utility.

This log file is located on the listener server where the utility is run: `<Campaign_home>/partitions/partition [n]/logs/unica_acsesutil.log`.

## The sessions log

The `ac_sess.log` file records information about server connections when flowcharts are opened.

When a user views a flowchart before editing it, session information for that flowchart is logged in the `ac_sess.log` file. The location of the log file depends on whether you have a clustered or single-node listener configuration:

Single-listener configuration: `<Campaign_home>/partitions/partition [n]/logs/ac_sess.log` on the listener server

Clustered configuration: `<campaignSharedHome>/partitions/partition [n]/logs/ac_sess.log`

## The web connections log

The `ac_web.log` file records information about the user's connections to the Unica Campaign system database.

When a user logs into Unica Campaign, information is logged in the `ac_web.log` file. The location of the log file depends on whether you have a clustered or single-node listener configuration:

For a single-listener configuration: `<Campaign_home>/partitions/partition [n]/logs/ac_web.log` on the listener server

For a clustered configuration: `<campaignSharedHome>/partitions/partition [n]/logs/ac_web.log`

## The cleanup utility log

The cleanup utility log file is generated if an error occurs when running the `unica_acclean` utility.

The log file is generated on the listener server where the utility is run: `<Campaign_home>/logs/unica_acclean.log`. The default name is `unica_acclean.log` but a different name can be assigned at run time.

## Windows™ event logs

When Unica Campaign is installed on Microsoft™ Windows™, you can optionally log events to the Windows™ event log for troubleshooting purposes.



**Attention:** Windows™ Event logging can cause issues with flowchart runs. Avoid enabling this feature unless advised by Technical Support.

Windows™ event logging of listener events is controlled by the configuration properties at `Unica Campaign | unicaACLlistener`.

Windows™ event logging of flowchart events is controlled by the configuration properties at `Unica Campaign | partitions | partition[n] | server | logging`.

To adjust these properties, you must have **Administer configuration page** permission in Unica Platform.

# Chapter 13. JVM Parameters

The following section provides the details of the JVM parameters for Campaign run time.

## Run the installer executable with this option:

```
-DUNICA_GOTO_CREATEEARFILE=TRUE
```

On UNIX type systems, run the .bin file rather than the .sh file.

The installer wizard runs.

## While deploying unica campaign on websphere application.

Add JVM argument -Dhttps.protocols=TLSv1.2 in Websphere for Unica Deliver to work.

```
-Dcollaborate.https.port=<https port configured for collaborate>
```



**Note:** This system property is required only if Campaign communicates with Collaborate using https protocol.

## Deploying Campaign on WAS from a WAR file

Users can add the following JVM arguments to enable unicode support with this JVM.

- -Dfile.encoding=UTF-8
- -Dclient.encoding.override=UTF-8
- -Dcollaborate.https.port=<https port configured for collaborate>

## Deploying Campaign on WAS from an EAR file

Users can add the following JVM arguments to enable unicode support with this JVM.

- -Dfile.encoding=UTF-8
- -Dclient.encoding.override=UTF-8

```
-Dhttps.protocols=TLSv1.2 -Djdk.tls.client.protocols=TLSv1.2
```

Set the protocol to TLSv1.2 from WAS admin console

```
-Dcollaborate.https.port=<https port configured for collaborate>
```

## Deploying Unica Campaign on WebLogic

Add the following parameters in the JAVA\_OPTIONS section of startWeblogic.cmd or startWeblogic.sh,

- -Dfile.encoding=UTF-8
- -Dclient.encoding.override=UTF-8
- -Dcollaborate.https.port=<https port configured for collaborate>

If you are deploying in a production environment, set the JVM memory heap size parameters to at least 1024 by adding the following line to the setDomainEnv script: Set MEM\_ARGS=-Xms1024m -Xmx1024m -XX:MaxPermSize=256m

### Configuring WebLogic to display reports (UNIX),

Modify the JAVA\_OPTIONS parameter to add the following value:

-Djava.awt.headless=true for your weblogic server (startWeblogic.sh)

-Dcollaborate.https.port=<https port configured for collaborate>

### Deploying Unica Campaign on JBoss Application Server

Modify the JAVA\_OPTIONS parameter to add the following value in setenv.bat/sh.

set "JAVA\_OPTS=%JAVA\_OPTS%

- -DUNICA\_PLATFORM\_CACHE\_ENABLED=true
- -Dclient.encoding.override=UTF-8
- -Djboss.as.management.blocking.timeout=3600
- -Dcollaborate.https.port=<https port configured for collaborate>

### Deploying Unica Campaign on Tomcat Application Server

Modify the JAVA\_OPTIONS parameter to add the following value in setenv.bat/sh.

set JAVA\_OPTS=%JAVA\_OPTS% -DUNICA\_PLATFORM\_CACHE\_ENABLED=true -Dclient.encoding.override=UTF-8.

-Dcollaborate.https.port=<https port configured for collaborate>

### While using datafilteringScriptTool with SSL, datafilteringScriptTool script (.bat or .sh)

SET SSL\_OPTIONS=-Djavax.net.ssl.keyStoreType="JKS"

-Djavax.net.ssl.trustStore="C:\security\myTrustStore.jks"

-Djavax.net.ssl.trustStorePassword=myPassword

"%JAVA\_HOME%\bin\java" -DUNICA\_PLATFORM\_HOME="%UNICA\_PLATFORM\_HOME%"

%SSL\_OPTIONS%

com.unica.management.client.datafiltering.tool.DataFilteringScriptTool %\*

### Configuring the ACOOptAdmin tool (ACOOptAdmin.sh (UNIX) or ACOOptAdmin.bat (Windows) file)

- -Djavax.net.ssl.keyStoreType=keyStoreTypeValue
- -Djavax.net.ssl.keyStore=<keyStoreValue>
- -Djavax.net.ssl.keyStorePassword=<keyStorePasswordValue>

- -Djavax.net.ssl.trustStore=<trustStoreValue>
- -Djavax.net.ssl.trustStorePassword=<trustStorePasswordValue>+

### **Configuring WebLogic to display reports (UNIX) (inside startWeblogic.sh)**

Modify the JAVA\_OPTIONS parameter to add the following value:

-Djava.awt.headless=true

-Dcollaborate.https.port=<https port configured for collaborate>



# Chapter 14. Unique code administration

Each campaign, cell, offer, and treatment in Unica Campaign has an identifying code that is generated by code generators, and conforms to a specified format.

Unica Campaign administrators can:

- Set configuration parameters to control how each type of code is generated, and valid formats for codes.
- Create a custom code generator if the default generators do not meet your needs.

All properties for configuring campaign and cell codes, code generators, and certain attributes of offer codes are set on the Unica Platform Configuration page.

Offer code formats are defined in offer templates rather than configured using parameters.

## About campaign codes

A campaign code is the globally unique identifier for a campaign. Each campaign must have a code, and no two campaign codes in the same Unica Campaign partition can be the same.



**Note:** Note that although campaign codes must be unique within each partition, campaign names do not need to be unique.

When users create a campaign, the **Campaign code** field is automatically populated with a unique value from the code generator.

Users can click **Regenerate code** to have the code generator supply a new identifier, or they can enter a code manually. If users manually enter a code, it must be unique and in the specified format.

## Changing the campaign code format

When you change campaign code format, the new format applies to all new campaigns. Existing campaigns can continue to use their current codes in the previous format. However, if a user edits the campaign code, the new code must adhere to the current campaign code format.

To complete this task, you must have appropriate permissions to use Unica Platform.

1. Choose **Settings > Configuration**.
2. Choose **Campaign > partitions > partition[n] > server > systemCodes**.
3. Set the `campCodeFormat` property. Be sure to follow the code format requirements.

## About cell codes


A cell code is an identifier for each cell in a flowchart or in the target cell spreadsheet.

In the flowchart processes that create new output cells (for example, Select, Merge, Segment, Sample, Audience and Extract processes), cell code(s) for the output of the process are configured in the **General** tab.

By default, the cell code is generated automatically; users can manually override the generated cell code by clearing the **Auto Generate** checkbox and entering a code in the valid format.

Whether cell codes must be unique within a flowchart depends on the setting of the `AllowDuplicateCellCodes` configuration parameter (described in Code generation references). If the value of `AllowDuplicateCellCodes` is `FALSE`, cell codes must be unique within a flowchart; the same cell code can still exist in different flowcharts and campaigns. If the value of `AllowDuplicateCellCodes` is `TRUE`, cell codes within a single flowchart do not have to be unique.

If duplicate cell codes are not allowed and a user enters a cell code that is already used elsewhere in the same flowchart, no error is immediately generated. However, users can validate flowcharts and detect duplicate cell codes using the flowchart validation tool if duplicate cell codes are not allowed. For information on flowchart validation, see the section on Validating Flowcharts in the Unica Campaign User's Guide.

 **Important:** Automatically generated cell codes are guaranteed to be unique only if no user ever overrides any cell code. For more information about working with cells, see the *Unica Campaign User's Guide*.

## Changing the cell code format

Do not change the cell code format after users have created flowcharts. Doing so will invalidate existing flowcharts.

To complete this task, you must have appropriate permissions to use Unica Platform.

1. Choose **Settings > Configuration**.
2. Choose **Campaign > partitions > partition[n] > server > systemCodes**.
3. Set the `cellCodeFormat` property. Be sure to follow the code format requirements.


## About offer and treatment codes

An offer code is the globally unique identifier for an offer. A treatment code is the globally unique identifier for a combination of a cell (a list of IDs) and an offer.

Each offer in Unica Campaign must have a code, and no two offer codes in the same Unica Campaign partition should be the same. An offer code can have one to five parts, which you specify when you create the offer template.

When users create an offer, the Offer Code fields are automatically populated with a unique value from the code generator.

Users can click **Regenerate code** to have the code generator supply a new identifier, or they can enter a code manually. To override offer codes, users must have the appropriate permission.

 **Important:** Automatically generated offer codes are guaranteed to be globally unique only if no user ever overrides any offer code.

The unique combination of a cell and an offer used at a particular point in time is a treatment. Each treatment is uniquely identified with a treatment code.

Separate treatments, and treatment codes, are generated each time a flowchart is run. If users run a flowchart on January 1, and again on January 15, two separate treatments are created. This allows you to track responses to offers in the most granular way possible.



**Note:** After treatment codes are generated, they cannot be overridden.

## Changing the offer or treatment code format in an existing offer template

You can change the offer and treatment code formats in existing offer templates only if the template has not yet been used to create offers.

You define the offer and treatment code formats for each offer template you create. You set the offer or treatment code formats at the time that you create each offer template. You can also change the offer and treatment code formats for existing offer templates by editing the template, but only if the template has not yet been used to create offers.

1. Choose **Settings > Campaign Settings**.
2. Click **Offer Template Definitions**.
3. Click the link for the offer template whose offer or treatment code format you want to change.
4. On the offer template definition page, modify the Offer Code Format or Treatment Code Format. Be sure to follow the code format requirements.



**Important:** Do not use the space character in an offer code format.

5. Click **Finish**.

## Code format requirements

The default and valid format for each type of generated code uses a series of characters to represent the character types. You can override the default formats for the codes generated by the Campaign built-in code generators.

Unique campaign, cell, treatment, and offer codes must be 32 characters or less. This restriction applies to codes generated by default and custom code generators as well as any manually entered codes. Offer codes must not include the space character.

The following table lists the characters that you can use to control code formats.

**Table 33. Control code formats**

Characters	Treated As
A-Z, any symbol, b-z (except c, n, x)	A constant value in the generated code
a	Any uppercase letter A-Z
c or x	Any uppercase letter A-Z, or any number, 0-9
x	Any uppercase letter A-Z, any number 0-9. However, users can replace the generated character with any ASCII character.  To specify variable length codes, the code format must end with one or more "x" characters, and the <code>allowVariableLengthCodes</code> property must be set to <code>TRUE</code> .
n	Any number 0-9

**Example:** The format definition `CAMP_aaannn` generates the following code: `CAMP_DWP839` (`CAMP_`, followed by three randomly generated capital letters, then three randomly generated numeric digits)

## Default code formats

The following table shows the default formats for campaign, cell, offer and treatment codes that are generated by the Unica Campaign built-in code generators.

**Table 34. Default code formats**

Code type	Default Value	Where Defined
Campaign	Cnnnnnnnnn	campCodeFormat parameter on the Unica Platform Configuration page
Cell	Annnnnnnnn	cellCodeFormat parameter on the Unica Platform Configuration page
Offer	nnnnnnnnn	In each offer template defined in Unica Campaign

**Table 34. Default code formats (continued)**

Code type	Default Value	Where Defined
Treatment	nnnnnnnnnn	In each offer template defined in Unica Campaign

## About code generators

Code generators are the programs used to automatically generate campaign, cell, offer, and treatment codes of the required format in Unica Campaign.

In addition to its built-in code generators, Unica Campaign supports custom code generators that you develop.

### Default code generators in Unica Campaign

Unica Campaign provides code generators that automatically generate campaign, cell, offer, and treatment codes matching the default specified format for each type of code.

The following table shows the name of each type of code's built-in code generator and its location:

**Table 35. Default code generators**

Code Type	Default Generator	Location
Campaign	uaccampcodegen	<install_dir>/Campaign/bin
Cell	uaccampcodegen	<install_dir>/Campaign/bin
Offer	uacoffercodegen	<install_dir>/Campaign/bin
Treatment	uaccampcodegen	<install_dir>/Campaign/bin

Replace <install\_dir> with the actual directory in which Unica Campaign is installed.

If the code generators built-in to Unica Campaign do not meet your company's needs, you can develop and use custom code generators.

## About custom code generators

If the Unica Campaign default code generators do not meet your needs, you can develop and use your own code generators.

A custom code generator is any program you develop to output unique campaign, offer, or cell codes (or all three). You can develop a custom code generator in any programming language that can be compiled into an executable file for the operating system on which the Unica Campaign web application is deployed.



**Important:** If the Unica Campaign web and analytical servers are deployed on separate machines, make sure you deploy the code generators on all machines.

The most common reason for creating a custom code generator is to generate codes that are useful for your company's business needs. For example, your custom code generator could be set up to create campaign codes containing the campaign owner's initials and the current date.

## Requirements for custom code generators

Custom code generators must meet several requirements.

- The executable name must be a single word with no spaces;
- The unique codes generated must match the specified code format, which is passed as an input to the custom code generator;
- The custom code generators must output the unique codes, or an error, to the standard output stream (`stdout`);
- Custom campaign and cell code generators must be placed in the `/Campaign/bin` directory. Custom offer code generators can be placed in a location of your choice which you then must specify in the offer code generator configuration properties on the Unica Platform Configuration page.

## About configuring Unica Campaign to use custom code generators

You specify campaign and cell code formats and generators using a property on the Unica Platform Configuration page.



**Note:** To complete this task, you must have the appropriate permissions in Unica. For information, see the *Unica Platform Administrator's Guide*.

You specify the offer and treatment code generators for each offer template you create. Each offer created based on a template then uses the program(s) you specify to generate unique offer and treatment codes.

### To specify the campaign code generator

On the configuration page, set the value of the `campCodeGenProgFile` property in the `Campaign > partitions > partition[n] > server > systemCodes` category to the executable name of the custom campaign code generator.



**Note:** To complete this task, you must have appropriate permissions to use Unica Platform. For information, see the *Unica Platform Administrator's Guide*.

## To specify the cell code generator

On the Configuration page, set the value of the `cellCodeGenProgFile` property in the `Campaign > partitions > partition[n] > server > systemCodes` category to the executable name of the custom campaign code generator.



**Note:** To complete this task, you must have appropriate permissions to use Unica Platform. For information, see the *Unica Platform Administrator's Guide*.

## To specify the offer code generator

You specify the offer code generator on the **Settings > Campaign Settings** page.

1. Log in to Unica Campaign and click **Settings > Campaign Settings**.
2. On the Campaign Settings page, click **Offer Template Definitions**.
3. Click the link for the offer template whose offer code generator you want to specify.
4. On **Step 1** of the new offer template definition page, enter the executable name of the custom offer code generator as the value of the **Offer Code Generator** field.
5. Click **Finish**.

## To specify the treatment code generator

You specify the treatment code generator on the **Settings > Campaign Settings** page.

1. Log in to Unica Campaign and click **Settings > Campaign Settings**.
2. On the Campaign Settings page, click **Offer Template Definitions**.
3. Click the link for the offer template whose offer code generator you want to specify.
4. On the **Step 1** offer template definition page, enter the executable name of the custom treatment code generator as the value of the **Treatment Code Generator** field. If you leave this field empty, the default treatment code generator is used.
5. Click **Finish**.

## About creating custom code generators

You can create custom code generators in any language that can be compiled into an executable file for the operating system on which you are running Unica Campaign.

## About outputting unique codes

The custom code generator must output unique codes, of no more than 32 characters, to the standard output stream (`stdout`).



**Important:** When Unica Campaign saves offer and cell codes, it does not check their uniqueness. You must ensure that any custom code generators you use can generate globally unique codes (assuming no users override generated codes).

The output line must:

- Start with `1`,
- Followed by one or more blank spaces,
- Followed by the unique code, within double quotation marks.

### Example

The following example shows the correct code output format:

```
1 "unique_code"
```

## About outputting errors

The custom code generator must output an error to the standard output stream (`stdout`) when it is not able to properly generate a unique code of the proper format.

The output line for the error must:

- Start with `0`,
- Followed by one or more blank spaces,
- Followed by the error message, within double quotation marks.

### Example

The following example shows the correct code output format:

```
0 "error_message"
```



**Note:** The error message generated by the custom code generator is displayed to the user and written to the log.

## About placing the custom code generators

You must place the application that generates campaign or cell codes in the `bin` directory of your Unica Campaign installation.

You can place custom offer code generators in a location of your choice, then specify the location using Unica.

## To specify the location of the custom offer code generator

On the Configuration page, change the value of the `offerCodeGeneratorConfigString` property in the `Campaign | partitions | partition_N | offerCodeGenerator` category to the location of the custom offer code generator executable. The location is relative to the Unica Campaign web application home.





**Note:** To complete this task, you must have appropriate permissions to use Unica Platform. For information, see the *Unica Platform Administrator's Guide*.

## Properties related to code generation

On the **Settings > Configuration** page, you can modify configuration properties to customize code formats and generators.

For information about these properties, see the context help or the Unica Platform Administrator's Guide.

**Table 36. Properties to customize code formats and generators**

Property	Path
allowVariableLengthCodes	Campaign   partitions   partition[n]   server   systemCodes
campCodeFormat	Campaign   partitions   partition[n]   server   systemCodes
campCodeGenProgFile	Campaign   partitions   partition[n]   server   systemCodes
cellCodeFormat	Campaign   partitions   partition[n]   server   systemCodes
cellCodeGenProgFile	Campaign   partitions   partition[n]   server   systemCodes
displayOfferCodes	Campaign   partitions   partition[n]   server   systemCodes
offerCodeDelimiter	Campaign   partitions   partition[n]   server   systemCodes
allowDuplicateCellcodes	Campaign   partitions   partition[n]   server   flowchartConfig
defaultGenerator	Campaign   partitions   partition[n]   offerCodeGenerator
offerCodeGeneratorClass	Campaign   partitions   partition[n]   offerCodeGenerator
offerCodeGeneratorClasspath	Campaign   partitions   partition[n]   offerCodeGenerator
offerCodeGeneratorConfigString	Campaign   partitions   partition[n]   offerCodeGenerator

## Parameters for the default campaign and cell code generators

The `uaccampcodegen` program supports the parameters described in this section. The `uaccampcodegen` program is located in the `bin` directory under your Unica Campaign installation directory.

**Table 37. Parameters for default campaign and cell code generators**

Parameter	Use
-c	Passes in the cell name.
-d	Passes in the day. Can accept one or two integers, not to exceed the value of 31.
-f	Passes in the code format, used to override the default format.
-i	Passes in an additional integer to be used to generate a unique code.
-m	Passes in the month. Can accept one or two integers, from 1 - 12.
-n	Passes in the campaign name.
-o	Passes in the campaign owner.
-s	Passes in an additional string to be used to generate a unique code.
-u	Passes in the campaign ID, to use instead of the system generated ID.
-v	Prints the first argument to the standard out stream (STOUT).
-y	Passes in the year. Accepts four integers.

## Parameters for the default offer code generator

The `uacoffercodegen` program supports the parameters described in this section. The `uacoffercodegen` program is located in the `bin` directory under your Unica Campaign installation directory.

**Table 38. Parameters for default offer code generator**

Parameter	Use
-a	Passes in the number of offer code parts, from 1 - 5.
-d	Passes in the day. Can accept one or two integers, not to exceed the value of 31.
-f	Passes in the code format, used to override the default format.
-i	Passes in an additional integer to be used to generate a unique code.
-m	Passes in the month. Can accept one or two integers, from 1 - 12.
-n	Passes in the campaign name.
-s	Passes in an additional string to be used to generate a unique code.
-u	Passes in the campaign ID, to use instead of the system generated ID.
-v	Prints the first argument to the standard out stream (STOUT).
-y	Passes in the year. Accepts four integers.

**Example**

```
uacoffercodegen -f "nnnnnnnnn nnnnnnnnn nnnnnnnnn nnnnnnnnn nnnnnnnnn"
-a 5 -u 3 -y 2008 -m 1 -d 14
```

## Parameters for custom code generators

Unica Campaign supports custom parameters as input to any custom code generator you configure for use in Unica Campaign.


No validation is performed for these parameters; however, the following restrictions apply:

- You cannot reuse the flags for the default Unica Campaign code generators as flags for parameters in the custom code generators.
- Do not use spaces in custom code generator executable names.
- Do not use double quotation marks around parameters or around the executable name.
- Spaces are read as separators between the code generator executable name, and between parameters. The first space is interpreted as marking the end of the executable name; subsequent spaces are interpreted as separating multiple parameters.
- The code generator fields for the Configuration Manager and the offer template interface are limited to 200 characters.

# Chapter 15. Advanced settings for individual flowcharts


When a flowchart is open for editing, administrators can choose the **Advanced settings** option on the **Admin** menu to make administrative changes that affect only the current flowchart.

Many of the **Advanced settings** options allow you to override global configuration settings for individual flowcharts. For example, if the Auto-save feature is set to 1 minute in the configuration settings but 2 minutes for an individual flowchart, then that flowchart will be recovered every 2 minutes. If no value is provided at the global level, then the value that is set at the flowchart level is used.

1. Open a flowchart in Edit mode.
2. Open the **Admin** menu  and select **Advanced settings**.
3. Use the available controls on the tabs of the **Advanced settings** dialog:
  - **General**: Save flowchart run results, use in-database optimization, disable global suppression, and other settings for the current flowchart. Also, send triggers on flowchart run errors or success.
  - **Server optimization**: Control the use of virtual memory and temporary tables for this flowchart.
  - **Test run settings**: Specify whether to write test run results to the database for this flowchart.


## Adjusting General settings for individual flowcharts

Use the **General** tab under **Admin > Advanced settings** to adjust administrative settings for an individual flowchart. For example, you can override global configuration settings for the current flowchart.

1. Open a flowchart in Edit mode.
2. Open the **Admin** menu  and select **Advanced settings**.

The **General** tab is selected by default. Use the controls to adjust administrative settings for the current flowchart.

## Save flowchart run results

The **Save flowchart run results** option under **Admin**  **> Advanced settings** lets you save the run results for an individual flowchart. Use this option to override the global configuration setting `Campaign|partitions|partition[n]|server|flowchartRun|saveRunResults`.

When you open a flowchart for editing, you can select **Save flowchart run results** to specify that all output cells from flowchart runs are saved when the run is finished. The next time you open the flowchart, you will be able to profile the results of any process that has finished running, or start a process or branch run from the middle of a flowchart. If you do not save results, each time you want to view results of a flowchart run you will have to re-run the entire flowchart from the beginning.

For flowcharts that create artifacts that you want to save, you must select **Save flowchart run results**. For example, if you have flowcharts that include CreateSeg processes, you must save run results. If you do not save run results, the strategic segments will not persist.


By default, this option is selected.

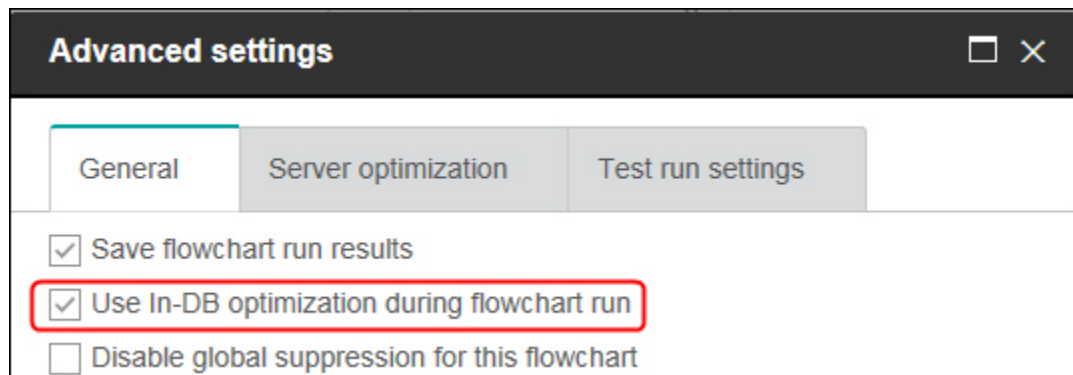
## Setting in-database optimization to improve flowchart performance

Using in-database optimization can improve flowchart performance. When in-database optimization is on, processing is done on the database server and output is stored in temporary tables on the database server whenever possible.


You can apply in-database optimization in two ways: globally and for individual flowcharts. The best practice is to turn off the global configuration setting and set the option at the flowchart level.


1. To adjust the option globally, at the partition level:
  - a. Choose **Settings > Configuration**.
  - b. Choose **Unica Campaign > partitions > partition[n] > server > optimization**.
  - c. Set **useInDbOptimization** to `TRUE` (on) or `FALSE` (off).
2. To override the option for an individual flowchart:

- a. Open a flowchart in **Edit** mode.
- b. Open the **Admin** menu  and select **Advanced settings**.
- c. Select or clear **Use In-DB optimization during flowchart run**.



When you save and run the flowchart, in-database processing will be used whenever possible, if you are using in-database optimization.

 **Note:** In-database processing cannot be done if you specify any limitations on the output cell size or if temporary tables are disabled for a process.

 **Note:** During a long running campaign flowchart, Campaign application loses connection to the DB2 database. Long running Campaign flowchart fails with SQL30081N error on Campaign server log



file. To run long running Campaign Flowchart, you must set the `STALE_CONN_TIMEOUT` (seconds time period) environment variable in Campaign listener (`rc.unica_ac/cmpServer.bat`). script files. Campaign server finds that if connection is idle for (`STALE_CONN_TIMEOUT`) seconds time period, it will re-open that connection.

## Details about in-database optimization

In-database optimization avoids copying IDs from the database to the Unica Campaign server for processing whenever possible. This option can improve flowchart performance.

In-database optimization determines:

- Whether operations are done on the database server or the local Unica Campaign server; and
- Where the results of operations are stored.

When in-database optimization is on:

- Processing tasks such as sorting, joining, and merging data are done on the database server whenever possible.
- Output cells of processes are stored in temporary tables on the database server.

In-database optimization affects CPU consumption:

- When in-database optimization is on, more CPU is consumed on the database server.
- When in-database optimization is off, more CPU is consumed on the Unica Campaign server.

You can apply in-database optimization globally and override the global setting for individual flowcharts. The best practice is to turn off the global configuration property (**Use in-DB optimization**) and set the option at the flowchart level (**Advanced Settings > Admin > Use in-DB optimization during flowchart run**).



**Important:** In-database processing cannot be done if you specify any limitations on the output cell size or if temporary tables are disabled for a process.

### Limitations of in-database optimization

- In-database optimization is not supported for all databases.
- Depending on the logic that is required, some functions are still performed on the Unica Campaign server, even with in-database processing turned on. Some examples are given below:

- The query uses tables from different data sources.

For example, if a Select process queries different data sources, Unica Campaign automatically stores the ID lists for those cases on the application server.

- The query contains non-SQL macros or derived fields.

For example, to calculate a derived field, Unica Campaign evaluates the derived field formula to see whether any part of the calculation can be performed with SQL. If simple SQL statements can be used, the calculation is done in-database. If not, temporary tables are created on the Unica Campaign server to handle the calculations and persist the results from process to process within a flowchart.

### Processing raw SQL in macros

Custom macros that consist of raw SQL statements can be processed in-database, within the following guidelines:

- All raw SQL custom macros must begin with `select` and contain exactly one `from` in the rest of the text.
- For databases that only support insert into `<TempTable>` syntax, you must map at least one base table to the same data source at the same audience level as the raw SQL custom macro. If the fields that are selected by the raw SQL custom macro are too large for the fields of the temp table, a runtime error occurs.
- If you use a raw SQL query in a Select process that has an input cell, you must use the `<TempTable>` token to obtain the correct list of audience IDs. Also use the `<OutputTempTable>` token to prevent audience IDs from being retrieved from the database back to the Unica Campaign server.
- If you use raw SQL with in-database optimization, you must code the raw SQL to join with the temp table from the upstream process. Otherwise, the results are not scoped by the results from the upstream process.

### Disable global suppression for this flowchart

Global suppression involves specifying a list of IDs (in a single audience level) that are automatically excluded from all cells in flowcharts in Unica Campaign.

If you have appropriate permissions, you can disable global suppression for this flowchart.



**Note:** If you do not have the appropriate permissions, you cannot change the setting and must run the flowchart with the existing setting. By default, new flowcharts are created with this setting cleared, and global suppressions applied.

### Y2K Threshold

The **Y2K Threshold** option under **Admin > Advanced Settings** determines how Unica Campaign interprets years that are represented with only two digits.



**Note:** We highly recommend that you store dates in the database with 4-digit years.

Valid values are 0 to 100; any values higher than 100 are set to 100. The default setting is 20.

Unica Campaign uses the threshold value to calculate a range of years whose lower limit is the threshold value + 1900 and whose higher limit is 99 years above that.



For example, if you set the threshold to 50, the range of years is from  $1900+50 = 1950$ , to 99 years above that, that is, 2049.

So, if you enter a two-digit year greater than or equal to your threshold (in this case, 50), the date is interpreted to be in the 1900s. If you enter a two-digit year less than your threshold, it is interpreted to be in the 2000s.

If you set the threshold to the maximum value of 100, the range of years will be  $1900+100 = 2000$ , to 2099. In this case, all two-digit years are interpreted to be in the 2000s.

You can change this threshold as needed.

## Auto-save (during user configuration)

The **Auto-save (during user configuration)** option under **Admin > Advanced settings** automatically saves an individual flowchart at the specified intervals. Use this option to override the global configuration setting `Campaign|partitions|partition[n]|server|flowchartSave|autosaveFrequency`.

You can set the auto-save feature to automatically save your work periodically for recovery purposes. If the Unica Campaign server process (unica\_acsvr) terminates while you are editing a flowchart, when you re-open the flowchart you will see the last auto-saved version of the flowchart.



**Note:** You must have previously saved the current flowchart (provided a file name) for this feature to work.

You can specify a number of minutes to control how often the flowchart will be saved. For example, if you enter 5, the flowchart is saved every 5 minutes. T

Unica Campaign stores auto-save files in a temporary directory (`CAMPAIGN_HOME\partitions\partitionN\tmp`), with the extension .asf, so that the original flowchart files are not changed. When you manually save and exit the flowchart, the .asf file is deleted and the flowchart is saved as a .ses file.

In non-recovery situations (for example, if you manually exit flowchart Edit mode without saving your flowchart), auto-saved versions are not retrieved. In this situation, when you re-open a flowchart that you manually left without saving, you will see the last manually saved version.

Auto-save never saves flowcharts that are in a paused state, even if Auto-save occurs while a selected process is running.

The default setting for Auto-save is **Never**.

## Checkpoint (during flowchart execution)

The **Checkpoint (during flowchart execution)** option under **Admin > Advanced settings** automatically saves a flowchart run at specified intervals. Use this option to override the global configuration setting `Campaign|partitions|partition[n]|server|flowchartSave|checkpointFrequency` for a specific flowchart.

The Checkpoint feature provides the ability to capture a "snapshot" of a running flowchart for recovery purposes. A checkpoint save has the same effect as if you selected **File > Save**. This feature allows you to recover a flowchart in the state of the most recent checkpoint save, in the event the server stops or goes down.

When you set a frequency interval for checkpoint, it controls a timer on the server for a running flowchart. Checkpoint saves are made at the specified intervals.

Checkpoint is active during a flowchart run and when you run a branch in the flowchart. When the running flowchart saves, Unica Campaign saves it in Paused mode. When you open the flowchart, you must either stop or resume the flowchart. When you resume, the currently executing processes run again from the beginning.

Unica Campaign stores checkpoint files in a temporary directory (`CAMPAIGN_HOME\partitions\partitionN\tmp`), with the extension .asf. The .asf files are deleted when the flowchart execution successfully completes.

If the server process (unica\_acsvr) goes down while the flowchart is running, then the flowchart run is recovered automatically from the .asf file. The execution flow can therefore resume from the last checkpoint that was saved before the process failed, so the flowchart run does not need to restart from the top process box.

The default setting for Checkpoint is **Never**.

## Maximum data errors allowed

The **Maximum data errors allowed** option under **Admin > Advanced settings** determines how many data errors are allowed during data export, for the current flowchart.

When Unica Campaign exports data to a file or a mapped table (for example, in a Snapshot or Optimize process), it occasionally encounters an error in format (for example, data does not fit into the table). The **Maximum data errors allowed** option lets Unica Campaign continue processing the file (if fewer than N number of errors occur) instead of failing on the first error.

The default is zero (0) errors.



**Note:** Set this value higher if you are debugging a problem with an export and you want to write the errors to a log file.

## Send trigger(s) on flowchart run errors

The **Send trigger(s) on flowchart run errors** option under **Admin > Advanced settings** lets you specify actions to occur if a campaign flowchart encounters errors during a run.

When you open a flowchart for editing, you can use this option to select one or more triggers from a list of outbound triggers. The selected triggers run if a campaign encounters errors during a flowchart run. Errors are indicated by a red X.

You most commonly use this option to trigger an email to alert an administrator of the problem. The selected triggers will run for each process run that fails.

## Send Trigger(s) on Flowchart Success


The **Send Trigger(s) on Flowchart Success** option under **Admin > Advanced settings** lets you specify actions to occur when a flowchart run completes successfully.

When you open a flowchart for editing, you can use this option to select one or more triggers from a list of outbound triggers.

You most commonly use this option to trigger an email to alert an administrator of the successful run. The selected triggers will run only if the entire flowchart run completes successfully.

## Adjusting server optimization settings for individual flowcharts

Use the **Server optimization** tab under **Admin > Advanced settings** to specify the virtual memory usage limit and override the use of temporary tables for a specific flowchart.

1. Open a flowchart in Edit mode.
2. Open the **Admin** menu  and select **Advanced settings**.

Select the **Server optimization** tab, then use the controls to adjust settings the current flowchart.

## Unica Campaign Virtual Memory Usage

The **Unica Campaign Virtual Memory Usage** option under **Admin > Advanced settings** allows you to specify the maximum amount of system virtual memory (MB) to use when running a specific flowchart.

You can increase the value to improve performance or decrease the value to limit the resources used by a single flowchart. The maximum value is 4095 MB. If you enter a larger value, Campaign automatically limits it to 4095 MB. The default value that appears is determined by the configuration setting **Campaign | partitions | partition[n] | server | optimization | maxVirtualMemory**.

## Disallow use of temporary tables for this flowchart

The **Disallow Use of Temp Tables for This Flowchart** option under **Admin > Advanced settings** allows you to specify that temporary tables should not be used for the current flowchart.


This option overrides the global configuration setting `Campaign|partitions|partition[n]|dataSources|dataSourcename|AllowTempTables`.

This option does not affect the system data source. Temporary tables are still created for the data source UA\_SYSTEM\_TABLES despite disallowing flowchart temporary tables. The flowchart temporary tables are independent of the system data source tables.

## Adjusting test run settings for individual flowcharts

Use the **Test run Settings** tab under **Admin > Advanced settings** to specify whether to write test run results to your database for a specific flowchart.

Typically, test run results are not written to your database. However, you might want to verify that the run results are being recorded properly. To do this, limit your cell size, then follow the procedure below. Limiting the cell size ensures that you are using a limited amount of data to test your flowchart run and its output.

1. Open a flowchart in Edit mode.
2. Open the **Admin** menu  and select **Advanced settings**.
3. Select the **Test run settings** tab.
4. Select **Enable output**.
5. Save the flowchart, then perform a test run.

# Chapter 16. Unica Campaign integration with other products

Unica Campaign optionally integrates with a number of other products.

For integration instructions, see the documentation that is included with each application, plus any documents mentioned below.

**Table 39. Integrate Unica Campaign with other HCL products**

Task	Documentation
Integrate with Acoustic	Campaign and Acoustic Integration Guide for IBM Marketing Cloud
Integrate with Coremetrics	<p><i>Unica Campaign Administrator's Guide</i>: How to configure the integration</p> <p>Unica Campaign User's Guide: Targeting IBM Digital Analytics segments in Campaigns</p>
Integrate with Unica Plan	Unica Plan and Unica Campaign Integration Guide
Integrate with Opportunity Detect	<p><i>Unica Campaign Administrator's Guide</i>: How to configure the integration</p> <p>Opportunity Detect Administrator's Guide and Opportunity Detect User's Guide: How to administer and use the product</p>
Integrate with Unica Deliver	<p><i>Unica Campaign Installation and Upgrade guides</i>: How to install and prepare Deliver components in the local environment.</p> <p>Unica Deliver Startup and Administrator's Guide: How to connect to the hosted messaging resources.</p> <p>Unica Campaign Administrator's Guide: How to configure offer integration.</p>
Integrate with IBM SPSS Modeler Advantage Enterprise Marketing Management Edition	<i>Unica Campaign and IBM SPSS Modeler Advantage Enterprise Marketing Management Edition Integration Guide</i>
Integrate with IBM WeatherFX.	You can integrate IBM WeatherFX with Campaign so that you can pull weather events into your marketing campaigns. Unica Campaign version 9 and version 10 users can integrate with WeatherFX by using the UBX toolkit.
Integrate with Unica Director.	<i>Unica Director Installation Guide</i> : You can integrate Unica Director with Campaign so that you can get status of all the run-

**Table 39. Integrate Unica Campaign with other HCL products (continued)**

Task	Documentation
	ning/completed/failed flowcharts, migrate Campaigns from one Unica Campaign environment to other, download/delete flow-chart logs etc.
Integrate with Journey	Unica Campaign can integrate with Journey so that a continuous communication with users can be established based on the inputs from Campaign.  Refer Campaign Administrators guide > Unica Campaign and Unica Journeys integration
Integrate with Link	Unica Campaign uses Link capabilities of connectors which include the configuration and provisioning to connect to any delivery channel.  Refer Campaign Administrators guide > Unica Campaign and Unica Link integration
Integrate with Segment Central	Unica Campaign can integrate with Segment Central so that a segments can be created and managed based on the inputs from Campaign.
Integrate with Audience Central	Unica Campaign can integrate with Audience Central so that a audiences can be identified and managed based on the inputs from Campaign.



**Note:** Additional integrations may be available that are not listed in this table.

## Overview of IBM Watson Marketing Assistant integration with Unica Campaign

Watson Marketing Assistant integration with Unica Campaign will enable real time interaction for Unica Campaign users, thus enabling businesses to serve their customers better. Call from Unica Campaign to EBA is secured and this is achieved using the JSON web token (JWT). Unica Campaign communicates with EBA-QUEST integrated application. Currently, the QUEST Cognos model is based on Unica Campaign system schema, Modelling on User database is not supported. For example, WMA in Unica Campaign will support queries like,

- 1.Which Unica Campaign has the most offers.
- 2.When was the particular Campaign created
- 3.When was 10 Percent Off offers used in Test Direct Marketing Campaign
- 4.How many people in control groups received 20 Percent Off offers; show offer start date

Though Watson integration is enabled, It is mandatory for a user to have Report System Role in order to access WMA.

Watson Assistant can be launched through the newly added menu item, under Analytics >“Watson Assistant?.

## Unica Campaign and Unica Centralized Offer Management integration

Unica Centralized Offer management when installed by default gets enabled. Unica Centralized Offer management makes all Campaign Offers information including (templates, attributes, offer folders, offers, offer lists) available in the Centralized Offer Management. Unica Centralized Offer Management integration is controlled based on the flag in Unica Platform configuration.

Settings for 'Unica Platform' (Affinium|suite)

```
set Enable Centralized offer management = Yes
```

Based on the the value of flag there are various changes related to Offers in Unica Campaign. Based on the Offer role assigned to Platform user - new Offers menu is available in the top menu navigation. Existing Offers menu under Campaign will not be available. If user disables the integration - new offers menu disappears and existing Campaign > Offers menu appears.

While Centralized Offer Management is enabled, users are required to perform the following tasks in Centralized Offer Management:

1. View Offers information
2. Add / Edit or other capabilities on Offers and Offer lists
3. Offer folder management
4. Offer template administration
5. Offer attributes management.

Offer management part from Campaign is not accessible while Centralized Offer Management is enabled.

Unica Campaign and Centralized Offer Management Offer and Unica Plan integration changes access to certain functionalities as mentioned in below table:

Centralized Offers	Campaign Integration	Plan-Offer Integration	Change in behavior to access offers functionality

En-abled	Dis-abled	Dis-abled	<ul style="list-style-type: none"> <li>• Campaign Offers menu will not be available. Users are required to use the Offers from Centralized Offers.</li> <li>• Campaign Offers templates will not be available. Users are required to use the Offer templates from Centralized Offers.</li> <li>• Campaign - custom attributes will be available for users, It will have access only to Campaign and Cell type attributes.</li> </ul>
En-abled	En-abled	Dis-abled	<ul style="list-style-type: none"> <li>• Campaign Offers menu will not be available. Users are required to use the Offers from Centralized Offers.</li> <li>• Campaign Offers Templates will not be available. Users are required to use the Offer templates from Centralized Offers.</li> <li>• Campaign - custom attributes are available. But it does not have Add attributes allowed. Only Campaign and Cells attributes are available for listing. Offers attributes creation is not allowed as COM is enabled. Campaign and Cell Attributes creation is not allowed as UMO UC is enabled.</li> </ul>
En-abled	En-abled	En-abled	<ul style="list-style-type: none"> <li>• Campaign Offers menu will not be available. Users are required to use the Offers from Centralized Offers.</li> <li>• Campaign Offers will not be available. Users are required to use the Offer templates from Centralized Offers.</li> <li>• Campaign - custom attributes are available. But it does not have Add attributes allowed. Only Campaign and cells attributes are available for listing. Offers attributes creation is not allowed as COM is enabled. Campaign and Cell Attributes creation is not allowed as UC UMO is enabled.</li> </ul>
En-abled	Dis-abled	En-abled	<ul style="list-style-type: none"> <li>• Campaign offers menu will not be available.</li> <li>• Campaign Offers templates will not be available. Users are required to use the Offer templates from Centralized Offers.</li> <li>• Campaign - custom attributes will be available for users, It will have access only to Campaign and Cell type attributes.</li> <li>• Only Campaign and Cell type attributes will be listed.</li> </ul>
Dis-abled	En-abled	Dis-abled	<ul style="list-style-type: none"> <li>• Campaign Offers menu will be available.</li> <li>• Campaign Offers Templates will be available.</li> <li>• Campaign - custom attributes are available. Listing of all attributes is available. Add attribute only allows Offer type attribute addition.</li> </ul>



Dis-abled	Dis-abled	En-abled	<ul style="list-style-type: none"> <li>• Campaign Offers menu will be available. (Plan Offers is enabled but UMO UC integration is disabled).</li> <li>• Campaign offers templates will be available. (Plan Offers is enabled, but MO UC integration is disabled)</li> <li>• Campaign - custom attributes are available. Listing of all attributes is available. Add attribute allows the addition of any type of attribute.</li> </ul>
Dis-abled	En-abled	En-abled	<ul style="list-style-type: none"> <li>• Campaign Offers menu will not be available. User are required to use Plan Offers.</li> <li>• Campaign Offers templates will not be available due to Plan and Campaign integration</li> <li>• Campaign - custom attributes are available. But it does not have Add attributes allowed. Listing of all types of attributes is available.</li> </ul>
Dis-abled	Dis-abled	Dis-abled	<ul style="list-style-type: none"> <li>• Campaign Offers menu will be available.</li> <li>• Campaign Offers templates link will be available.</li> <li>• Campaign - All custom attributes will be available for creation.</li> <li>• All attributes types will be listed.</li> </ul>

## Leveraging new capabilities from Centralized Offer Management

Support to new type of Offer Attributes:

The following are the attributes with new Centralized Offer Management.

- Checkbox - Boolean
- Picker URL
- Text Integer
- SSDB
- Rich Text



**Note:** From 12.1 FP 2 onwards, Centralized Offer Management supports two New Offer Attributes types, SSDB and RichText. For more detail please check product guide.

These newly introduced offer attributes are available in Maillist or Callist process boxes for personalization.



**Note:** The personalization of these attributes values does not have any validation when personalized from Maillist or Call list process boxes.

## Adding new attribute in existing Offers or Templates

Centralized Offer Management supports adding new attribute in the existing Offers or Offer template. Users can add new attribute in existing offers or template, which are already assigned in flowcharts. In this case, these attributes will not get reflected in flowchart run or process boxes, unless users explicitly remove and assign this offer to the cell.

## Setting up Unica Plan assets for use in Unica Campaign offers

This topic explains what administrators must do to allow Unica Campaign users to relate a digital asset from Unica Plan to a Unica Campaign offer.

An asset is an electronic file that is designed for use in a marketing program. Examples include logos, brand images, marketing research documents, reference materials, corporate collateral, or document templates. To add an asset to a Unica Campaign offer, you use the **CreativeURL** attribute. The **CreativeURL** attribute is a standard offer attribute that is installed with Unica Campaign. A "Creative URL" is a pointer to a file in a Unica Plan asset library.

**Table 40. Setting up Unica Plan assets for use in Unica Campaign offers**

### Setting up Unica Plan assets for use in Unica Campaign offers

Task	Details	For documentation
Create a library to hold digital assets.	This task is typically done by a Unica Plan administrator.  In Unica Plan, select <b>Settings &gt; plan settings</b> , click <b>Asset library definitions</b> , and add a library.	Unica Plan Administrator's Guide
Add assets to the library.	This task is typically done by Unica Plan users.  In Unica Plan, select <b>Operations &gt; Assets</b> . Open a library, go to a folder, and click the <b>Add asset</b> icon. Specify the asset name, description, and other information, and use <b>Upload</b> to select and upload files into the library.	Unica Plan User's Guide
Create an offer template that includes the <b>CreativeURL</b> attribute.	This task is typically done by a Unica Campaign administrator.  Offer attributes are the fields that define an offer. <b>Creative URL</b> is a standard attribute supplied with Unica Campaign. Adding the <b>Creative URL</b> attribute to a template makes the attribute available to any offers based on that template.	For systems that do not integrate Unica Plan and Unica Campaign, see the Unica Campaign Administrator's Guide: <a href="#">Creating offer templates (on page 126)</a> .  For systems that integrate Unica Plan and Unica Campaign, see the

**Table 40. Setting up Unica Plan assets for use in Unica Campaign offers****Setting up Unica Plan assets for use in Unica Campaign offers**

(continued)

Task	Details	For documentation
	<p>For example, in a system that does not integrate Unica Plan with Unica Campaign, you select <b>Settings &gt; Campaign Settings</b>, and click <b>Offer template definitions</b>. Click <b>Add</b>, then follow the prompts:</p> <ol style="list-style-type: none"> <li>1. In Step 1 of 3, define the template.</li> <li>2. In Step 2 of 3, move <b>Creative URL</b> into the Selected Attributes list.</li> <li>3. In Step 3 of 3, click <b>Browse library</b> in the <b>Creative URL</b> field. Go to a folder in an asset library and select an asset to use in this offer. Or, if you want to create an asset, click the name of a library, then click <b>Add asset</b> and provide the required information. In the <b>File</b> field, click <b>Upload</b>, then browse to a file. You can upload a File, a Preview File, and a Thumbnail. Follow the prompts to complete the action.</li> </ol> <p>The URL to the asset is now included in the <b>Creative URL</b> field.</p> <ol style="list-style-type: none"> <li>4. Save the offer template.</li> </ol>	Unica Plan and Unica Campaign Integration Guide.
Use Unica Campaign to create an offer that includes an asset from Unica Plan.	Unica Campaign users can now create offers based on a template that includes the <b>Creative URL</b> attribute. When defining the offer, users can go to the asset library and select or create an asset.	Unica Campaign User's Guide

## Configuring Unica Deliver offer integration with Unica Campaign

You can configure Unica Campaign to support Unica Deliver offer integration, so offer communications can be tracked over the email channel. Unica Campaign reports can then provide detailed response tracking for Unica Deliver.

Before you configure Unica Deliver offer integration, you must install and prepare the local installed Unica Deliver components, as explained in the Unica Campaign installation and upgrade guides. You must also connect to the hosted email resources, as explained in the *Unica Deliver Startup and Administrator's Guide*.

The following table lists the tasks that Unica Campaign administrators must complete to configure Unica Deliver offer integration.

**Table 41. Configuring Unica Deliver offer integration**

**Configuring Unica Deliver offer integration**

Task	Details	For more information
<p>1. Configure the contact and response history ETL process for this partition.</p>	<p>Unica Campaign uses its own ETL process to extract, transform, and load offer response data from the Unica Deliver tracking tables into the Unica Campaign contact and response history tables. The ETL process coordinates information across the necessary tables.</p> <p>To configure the ETL process:</p> <ol style="list-style-type: none"> <li>1. In Unica Campaign, select <b>Settings &gt; Configuration</b>.</li> <li>2. Adjust the properties for the following parameter:</li> </ol> <pre>Unica Campaign   partitions   partition[n]   Unica Deliver   contactAndResponseHistTracking</pre>	<p>See <a href="#">Campaign   partitions   partition[n]   Deliver   contactAndResponseHistTracking</a> (on page 361).</p>
<p>2. Optionally, adjust the ETL logging properties.</p>	<p>ETL logging is enabled by default. The default file location is <code>Campaign_home/logs/ETL.log</code>. To adjust logging behavior, modify the <code>campaign_log4j.xml</code> file, whose default location is <code>Campaign_home/conf</code>.</p>	<p>See <a href="#">The Unica Campaign and Unica Deliver ETL log file</a> (on page 193).</p>
<p>Optionally, allow monitoring of the ETL process so JMX-based clients can see the progress of the ETL.</p>	<p>In Unica Campaign, select <b>Settings &gt; Configuration</b>, then set the following parameter to <code>Yes</code>:</p> <pre>Campaign   monitoring   monitorEnabledForDeliver</pre>	<p>See <a href="#">Campaign   monitoring</a> (on page 498)</p>
<p>3. Upgrades only: Adjust Unica Campaign</p>	<p>New installations can skip this step because the tables were updated during the installation process.</p>	<p>See <a href="#">Adjusting Unica Campaign response tables for Unica Deliver offer integration</a> (on page 229).</p>

**Table 41. Configuring Unica Deliver offer integration****Configuring Unica Deliver offer integration**

(continued)

Task	Details	For more information
response tracking and mapping tables.	<p>Upgrade customers who want to use Unica Deliver offer integration must add the Link Click, Landing Page, and SMS Reply Message response types to the <code>UA_UsrResponseType</code> table, then update the <code>UA_RespTypeMapping</code> table to map the response types to Unica Deliver.*</p> <p>The <code>UA_RespTypeMapping</code> table is required to coordinate transfer of data across the necessary Unica Campaign and Unica Deliver tables.</p>	
4. Optionally, adjust Unica Campaign performance reports.	<p>The Unica Campaign Detailed Offer Response Breakout report includes Link Click, Landing Page, and SMS Reply Message email responses by default, so no adjustments are necessary for this report.*</p> <p>The other performance reports show the sum of all contacts or responders over all channels that were used, but they do not specifically break out the information by channel. You can customize these reports to include channel information, if desired.</p> <p>Changes affect both the object-specific reports on the campaign or offer <b>Analysis</b> tab and the reports on the <b>Analytics</b> menu.</p>	See the Unica Reports Installation and Configuration Guide.

\*Only the Link Click response type is currently used for Unica Deliver offer integration. The Landing Page and SMS Reply Message columns are not populated by the ETL process at this time.

## Adjusting Unica Campaign response tables for Unica Deliver offer integration

If you are using Unica Deliver offer integration, you must confirm that the Unica Campaign response tracking and mapping tables are set up properly.



**Note:** New installations can skip this task because the tables were updated during the installation process. Upgrade customers who want to use Unica Deliver offer integration must perform the following steps.

To support ETL of response types between Unica Campaign and Unica Deliver, response types must be defined in both the `UACE_ResponseType` table (for Unica Deliver) and the `UA_UsrResponseType` table (for Unica Campaign).

Then the response types must be mapped in the `UA_RespTypeMapping` table. The `UA_RespTypeMapping` table maps the **CampaignRespTypeID** in `UA_UsrResponseType` to the **DeliverRespTypeID** in `UACE_ResponseType`

1. Add the Link Click, Landing Page, and SMS Reply Message response types to the `UA_UsrResponseType` table using the values shown below:
  - insert into `ua_usrresponsetype` (`ResponseTypeId`, `Name`, `Description`, `ResponseTypeCode`, `CountsAsResponse`, `isDefault`) values (9, 'Link Click', NULL, 'LCL', 1, 0)
  - insert into `ua_usrresponsetype` (`ResponseTypeId`, `Name`, `Description`, `ResponseTypeCode`, `CountsAsResponse`, `isDefault`) values (10, 'Landing Page', NULL, 'LPA', 1, 0)
  - insert into `ua_usrresponsetype` (`ResponseTypeId`, `Name`, `Description`, `ResponseTypeCode`, `CountsAsResponse`, `isDefault`) values (11, 'SMS Reply Message', NULL, 'SRE', 1, 0)

For more information, see [Default response types \(on page 158\)](#).

2. Update the `UA_RespTypeMapping` table to include Link Click (9,1,3), Landing Page (10,14,3) and SMS Reply Message (11,18,3), as shown below:
  - insert into `ua_resptypemapping` (`campaignresptypeid`, `Deliverresptypeid`, `applicationtype`) values (9,1,3)
  - insert into `ua_resptypemapping` (`campaignresptypeid`, `Deliverresptypeid`, `applicationtype`) values (10,14,3)
  - insert into `ua_resptypemapping` (`campaignresptypeid`, `Deliverresptypeid`, `applicationtype`) values (11,18,3)

The `ApplicationType` of 3 indicates Unica Deliver and must not be changed.



**Note:** Only the Link Click response type is currently used for Unica Deliver offer integration. Landing Page and SMS Reply Message are not populated by the ETL process at this time.

Additional information about Unica Deliver offer integration is provided in the *Unica Deliver User's Guide* and the *Unica Campaign User's Guide*.

## Integrating IBM Digital Analytics and Unica Campaign

Online segments that are defined in Digital Analytics can be used in Unica Campaign to target customers based on their web activity and behavior. **Important!** This topic pertains to IBM Digital Analytics, not Digital Analytics for On Premises.


- Regardless of the Unica Campaign environment that you are integrating (Development, Test, Stage, Production), you must point the Unica Campaign installation to a production version (not a test version) of the IBM Digital Analytics environment, because that is the only environment that has the required "export" capability.
- The server machine where the Unica Campaign listener is running must be able to reach the IBM Digital Analytics data center in order to communicate with the hosted Coremetrics URL (<https://>

export.coremetrics.com/eb/segmentapi/1.0/api.do). Use port 443, which is the default port for https. Without direct access between the listener machine (which may sit behind a secure firewall) and the data center, the integration will not work.

The integration relies on several components:


- An integration service that acts as the integration point between the two products.
- A translation table that maps Digital Analytics keys to Unica Campaign audience IDs.
- Configuration settings in Unica Campaign that enable the integration on a per-partition basis.
- Permissions and accounts set in Digital Analytics, Unica Platform, and Unica Campaign.

1. Make the following adjustments in IBM Digital Analytics and Unica Campaign.

Task	Performed by	Details	For documentation
<b>A.</b> Configure a Digital Analytics client ID to support the integration.	Provisioning	Global User Authentication must be enabled and an ExportBuilderSegmentAPI role must be added to your Digital Analytics client ID.	This step is performed by IBM.
<b>B.</b> Configure Digital Analytics for integration with Unica Campaign.	Digital Analytics user with admin rights	Create a user group and a user account in the Digital Analytics production environment for all client IDs that you want to access in Unica Campaign. Assign the user to the group, then set Global User Authentication.Export   <b>Important:</b> After you perform this task, IBM Provisioning must enable the ExportBuilderSegmentAPI role for the new user group.	See <a href="#">Configuring Digital Analytics to enable Unica Campaign integration (on page 234)</a> .
<b>C.</b> Broadcast Digital Analytics segments for use in Unica Campaign.	Digital Analytics user with admin rights	In Digital Analytics, select <b>Manage &gt; Report options &gt; Report segments</b> . In the <b>Actions</b> menu, click the <b>Broadcast</b> icon next to the segment that you want to share with Unica Campaign. When opens, complete the segment form.	See the Digital Analytics and Export product documentation.
<b>D.</b> Create and populate a translation table.	IBM Consulting, IBM Business Partner, or customer IT team	A translation table maps the Digital Analytics registrationid (online key) to the Unica Campaign audience ID (offline key).	See <a href="#">Creating and populating the translation table (on page 235)</a> .

Task	Performed by	Details	For documentation
<b>E. Optional:</b> Configure SSO.	Unica Campaign user with admin rights	Single sign-on (SSO) lets Unica Campaign users access Digital Analytics without being prompted to log in, for easier navigation between products.	See the <i>Unica Platform Administrator's Guide</i> .
<b>F. Configure a Unica Platform account and data source to access the integration service.</b>	Unica Campaign user with admin rights	This step may be done before or after step G.  Choose <b>Settings &gt; Users</b> , select the same user that is defined for <b>ASMUserForCredentials</b> (see step G), click <b>Edit data sources</b> , and add a new data source: <ul style="list-style-type: none"> <li>◦ The Data Source name must exactly match the value that is defined in Unica Campaign for <b>ASMDatasourceForCredentials</b>.</li> <li>◦ The <b>Data source login</b> and <b>Password</b> are the Digital Analytics user account credentials that were defined in <i>step B, Configure Digital Analytics for integration with Unica Campaign</i>.</li> </ul>	See the <i>Unica Platform Administrator's Guide</i> .
<b>G. Set Unica Campaign configuration properties for each partition where you will enable integration.</b>	Unica Campaign user with admin rights	Choose <b>Settings &gt; Configuration</b> .  Go to <b>Unica Campaign   partitions   partition[n]   Coremetrics</b> and set the following properties: <ul style="list-style-type: none"> <li>◦ <b>ServiceURL</b>: <code>https://export.coremetrics-.com/eb/segmentapi/1.0/api.do</code></li> <li>◦ <b>CoremetricsKey</b>: <code>registrationid</code></li> <li>◦ <b>ClientID</b>: The Digital Analytics ID that is assigned to your company. If you have multiple IDs, use any client ID that was selected for the user account in <i>step B. Configure Digital Analytics for integration with Unica Campaign</i>.</li> <li>◦ <b>TranslationTableName</b>: The name of the translation table.</li> <li>◦ <b>ASMUserForCredentials</b>: The Unica Platform account that is allowed to access</li> </ul>	See <a href="#">Campaign   partitions   partition[n]   Coremetrics (on page 372)</a>



Task	Performed by	Details	For documentation
		<p>the integration service. The default is <code>as-admin</code>.</p> <ul style="list-style-type: none"> <li>◦ <b>ASMDatasourceForCredentials:</b> The data source assigned to the Unica Platform account. The default is <code>UC_CM_ACCESS</code>.</li> </ul>	
<b>H.</b> Map the translation table.	Unica Campaign user with admin rights	<p>Choose <b>Settings &gt; Unica Campaign Settings &gt; Manage table mappings</b>. When prompted, specify the data source that corresponds to the user database where the user tables reside (<i>not</i> the data source that you defined for <code>ASMDatasourceForCredentials</code>).</p> <p>Select the translation table, the table fields, and audience levels.</p> <p>Store the table mapping in the default catalog (<code>default.cat</code>) so it is available to all flowcharts.</p>	See <a href="#">Mapping the translation table (on page 240)</a> .
<b>I.</b> Give Unica Campaign users permission to use Digital Analytics segments in flowcharts.	Unica Campaign user with admin rights	<p>Choose <b>Settings &gt; User roles and permissions</b>.</p> <p>Go to <b>Unica Campaign   Partition[n]   Global Policy</b>.</p> <p>Click <b>Add roles and assign permissions</b>, then <b>Save and edit permissions</b>.</p> <p>Under <b>Campaigns</b>, grant permission to <b>Access digital analytics segments</b>.</p> <p> <b>Note:</b> Even if you are using single sign-on (SSO), you still have to set Global Policy to provide access to segments.</p>	See <a href="#">Implementing security policies (on page 30)</a>
<b>J.</b> Enable the integration for each Unica Campaign partition.	Unica Campaign user with admin rights	<p>Choose <b>Settings &gt; Configuration</b>. Go to <b>Unica Campaign   partitions   partition[n]   server   internal</b> and set <b>UC_CM_integration</b> to Yes.</p>	See <a href="#">Campaign   partitions   partition[n]   server   internal (on page 487)</a> .

2. Unica Campaign users can now select Digital Analytics segments in flowcharts:

- a. Add a Select process to a flowchart.
- b. Choose **Digital Analytics segments** as the **Input**.

- c. Select a **Client ID** to see the segments that were broadcast for that ID.
- d. Select a segment. The **Select segment** list shows only the segments that were created for the client ID that is selected. To see other segments, select a different client ID (assuming you have permission to see that client ID).
- e. Use the **Segment range** date and calendar controls at the bottom of the dialog to specify the date range from which you want to obtain data for the selected segment.

When the Select process runs:

- It pulls data from Digital Analytics via the integration service. The segment data is simply a list of registration IDs.
- Using the mapped translation table, registration IDs are converted to Unica Campaign audience IDs.
- The audience IDs are then available for use in downstream processes in the flowchart.

For more information about flowcharts, see the [Unica Campaign User's Guide](#).

## Configuring Digital Analytics to enable Unica Campaign integration

This task explains how to configure Digital Analytics for integration with Unica Campaign. In this task, you create a user group, create a user and assign it to the group, and set Global User Authentication. This task is typically performed by an IBM Digital Analytics user with administrator privileges.

1. As an Admin, log in to Digital Analytics under the Client ID that you want to enable and navigate to the Admin page.
2. Click **Manage groups - full list** or **Manage groups - by group**.
3. Click **New user group**.
4. In the **New user group** dialog, specify the following information:
  - **Group name:** For example: *MyCompany Campaign Integration Group*
  - **Client ID:** If you have multiple Client IDs, you must select all of the IDs that you want to use in Unica Campaign. Click the **Select** button. Then check the boxes for all IDs that you want to be able to access in the Unica Campaign UI, and click **OK**.
  - Select **Standard access**.
  - **Permission options:** Check **Digital Analytics**, **Digital Analytics explore**, and **Digital Analytics export**.
5. Click **Save**.
6. Click **Manage users – by user** or **Manage users – full list**.
7. Click **New user**.
8. In the **New user** dialog, specify the following information:
  - **Name:** For example: *Demo image campaign API*.
  - **User name** and **Password:** The User Name and Password that you specify here must match the Data Source Login and Password that are defined in the Unica Platform data source.
  - **Email:** Enter an email address.
  - **Client ID:** Check the boxes for each client ID that you want to be able to access in Unica Campaign. For **User group**, select the group name that you specified previously.



**Note:** Depending on the password settings for the Client ID, you may need to proactively set up a calendar entry for yourself to update the password before it expires. For best results, set to the same password; otherwise it will break the integration unless the passwords are also updated in Unica Campaign.

9. Click **Save**.

10. Click **Global user authentication** and modify the following information:

- **Shared secret:** Use the same password that you created previously.
- **Automatic user account creation:** Enabled
- **Select the group accounts will automatically be created in:** Select the group name that you specified previously.



**Important:** After you create the new user group, IBM Provisioning must enable the ExportBuilderSegmentAPI role for the new user group.

To complete the integration, perform the remaining steps that are described in the topic [Integrating IBM Digital Analytics and Unica Campaign \(on page 230\)](#).

## Creating and populating the translation table

A translation table is required to support the IBM Digital Analytics and Unica Campaign integration. The translation table is typically created and populated by IBM Consulting, an IBM Business Partner, or the customer IT team as an initial step in configuring the integration.

A translation table typically consists of two columns: One for the online key (registrationid) from IBM Digital Analytics, and one for the corresponding offline key (audience ID) that is used by Unica Campaign. You must create the table and then populate it.

1. Create a translation table that adheres to the following guidelines:

The table must be configured on the user data source (typically an Enterprise Data Warehouse or datamart) that will provide selections to Unica Campaign.

The data source must allow table creation permission for the user, because Unica Campaign needs to create a temporary table on the data source at run-time, to hold the list of IDs that meet the segment definition.

The first column must be called `registrationid`:

- You must use this exact name.
- This field holds the IBM Digital Analytics RegistrationID (online key).
- The data type must be the same data type defined for the registrationID in IBM Digital Analytics. For example, they must both be `VARCHAR`.
- The size of this field depends on what you are using for the registrationID. For example, if the registrationID includes the email address, 256 is a reasonable size.

The second column holds the primary audience level ID (offline key) for Unica Campaign:

- Use the audience name that is defined in Unica Campaign.
- The audience ID and its data type are determined by the customer whose system is being integrated. For example, it might be `CustomerID` or `AccountID`, and the data type might be `BIGINT`.

The table can include only *one* audience, but the audience can consist of multiple fields (columns):

- For performance and storage reasons, best practice is to use single-key audiences.
- If the primary audience consists of multiple physical keys (compound keys), the translation table must contain a column for each audience key plus a column for `registrationID`. For example, if your primary audience consists of the keys `CustomerID` and `AccountID`, then the translation table must include three columns: `registrationid`, `CustomerID`, `AccountID`. This requirement pertains only if you are mapping for a compound audience.

2. Populate the translation table. For guidelines, see [Translation table data sources \(on page 236\)](#).

The method for populating the translation table depends on each customer's requirements and configuration:

- Determine a common logic to identify which IBM Digital Analytics registration ID matches which Unica Campaign audience ID.
- After an initial complete load of the translation table with the `registrationid` from IBM Digital Analytics and the audience information from the customer data, deltas can be loaded into the translation table based on a predefined schedule. This is customer-specific and varies per implementation.



**Important:** If the translation table does not contain "registrationid to CustomerID" mapping information, then that particular record will be dropped from the selection during the flowchart run. Therefore, to avoid data loss, it is important to keep this table up-to-date.

Perform the additional required steps to configure the integration. For example, you must map the translation table in Unica Campaign. For a complete list of steps, see [Integrating IBM Digital Analytics and Unica Campaign \(on page 230\)](#).

## Translation table data sources

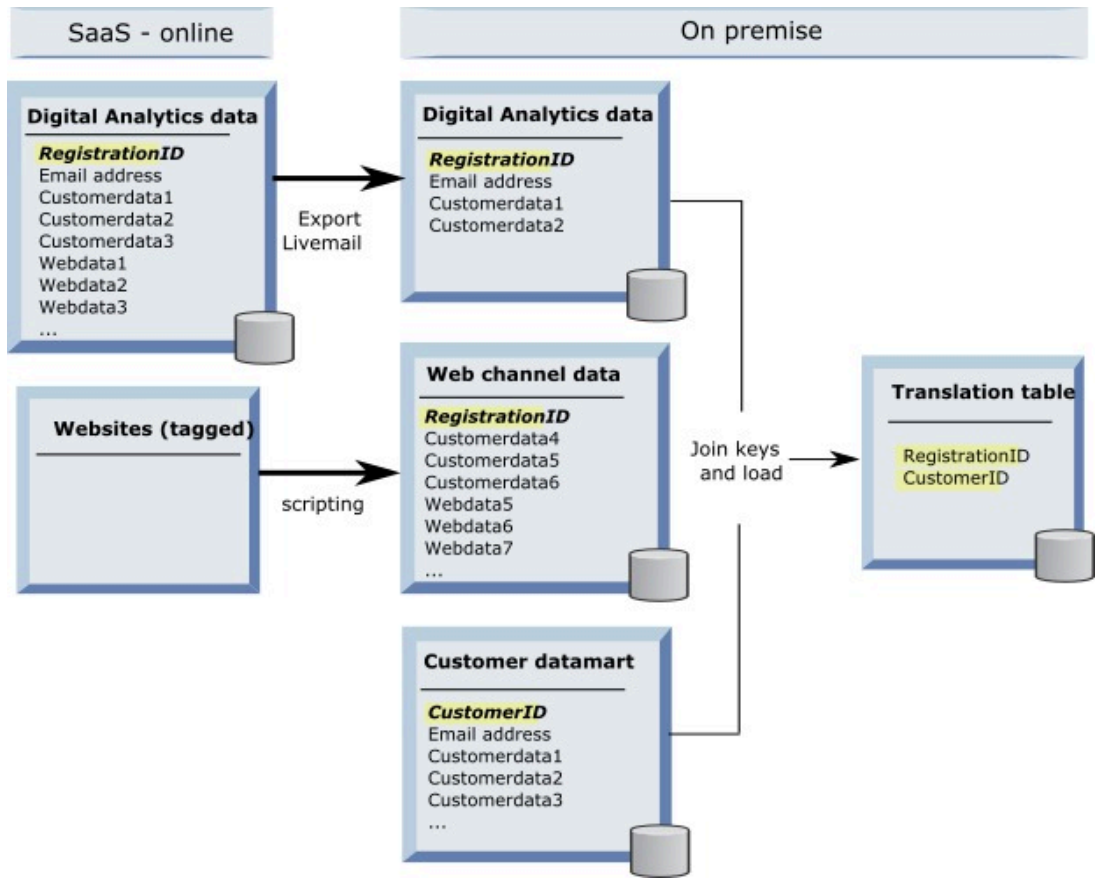
The following diagrams show different scenarios that you can consider when determining how to populate the translation table. A translation table is required to coordinate the transfer of data between IBM Digital Analytics and Unica Campaign.

A translation table contains one column for the IBM Digital Analytics `registrationID` and another column for the Unica Campaign audience identifier (such as `CustomerID` or `AccountID`). This mechanism matches IDs from one data source to another.

A typical integration can have access to both online (SaaS) and on-premise data sources:

- Web data is available in a web datamart, which contains information from web channel interfaces.
- Data can be exported from the SaaS IBM Digital Analytics solutions, using IBM Digital Analytics Export (`registrationid`) and Livemail (for other web-related data).
- Customer data sources, such as databases or flat files (on-premise).

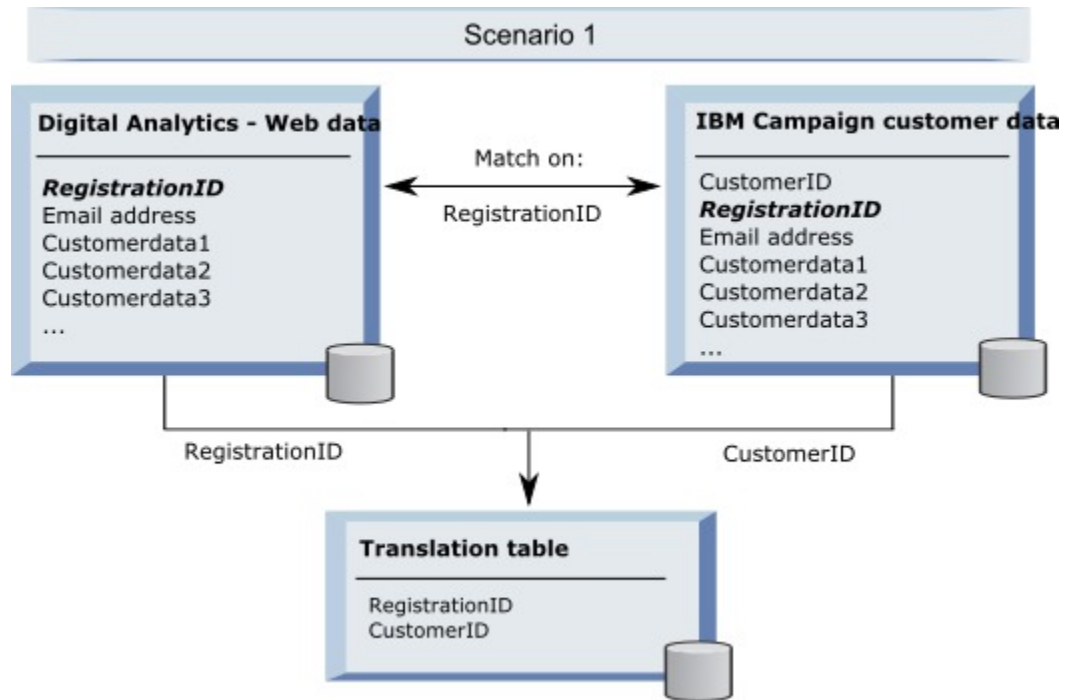
The following illustration shows how data sources feed into a translation table. The translation table associates records across the products, using the IBM Digital Analytics `registrationID` and the Unica Campaign audience ID (CustomerID in this example).



The following examples show different scenarios that you can consider when determining how to populate the translation table. These scenarios provide examples of how to use data matching to identify records that correspond to the same entities across multiple databases.

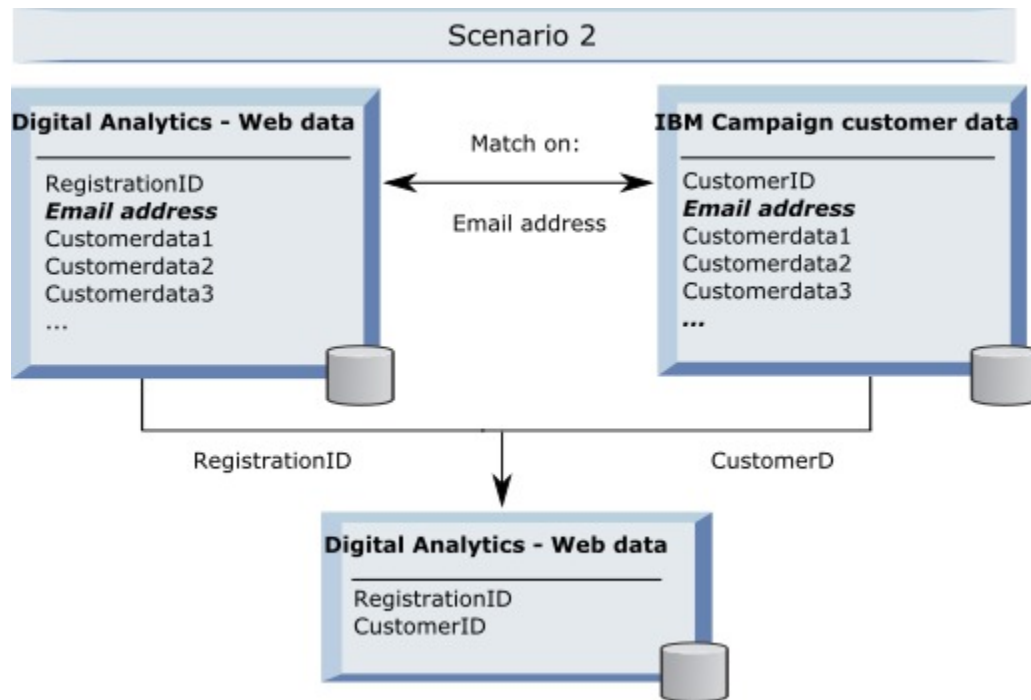
### Scenario 1: Same key in Web data and Unica Campaign

In Scenario 1, the Web data and customer data both contain the same key, `RegistrationID`. You can match on the `RegistrationID` to identify corresponding records.



**Scenario 2: Different keys in Web data and Unica Campaign, one binding unique key**

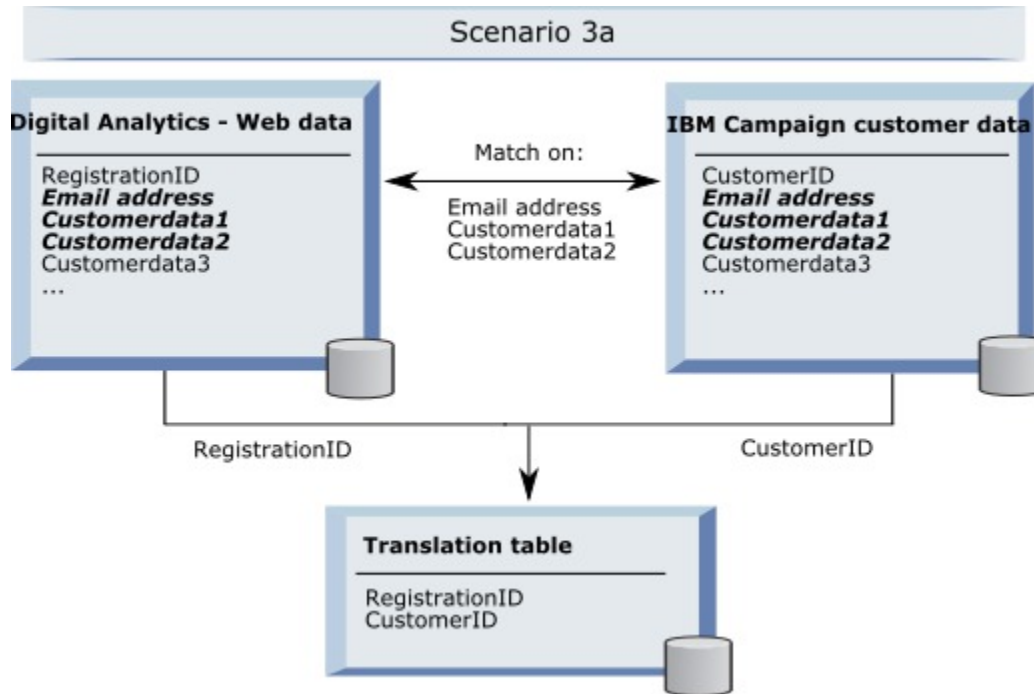
In Scenario 2, the Web data uses RegistrationID as its key, and the customer data uses an audience identifier (CustomerID). The email address is used to bind the keys.



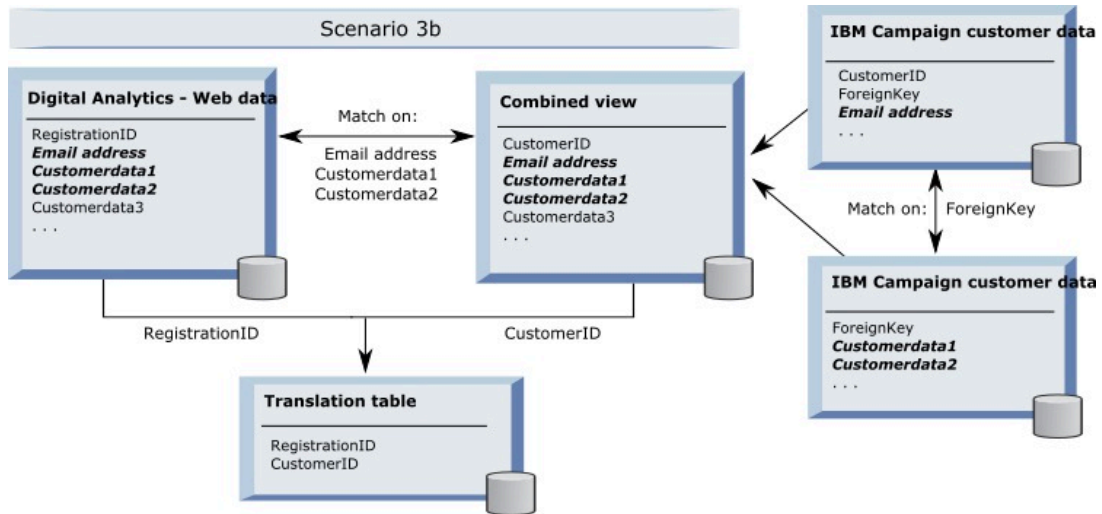
### Scenario 3: Different keys in Web data and Unica Campaign, multiple binding unique keys

- Scenario 3a: Multiple binding unique keys in one table
- Scenario 3b: Multiple binding unique keys in multiple tables
- Scenario 3c: Multiple binding unique keys in multiple databases (not depicted)

The following example shows Scenario 3a, Multiple binding unique keys in one table. In this scenario, the Web data uses RegistrationID as its key, and the customer data uses an audience identifier (CustomerID). The email address plus additional unique identifying data fields (Customerdata1, Customerdata2) are used to bind the keys.

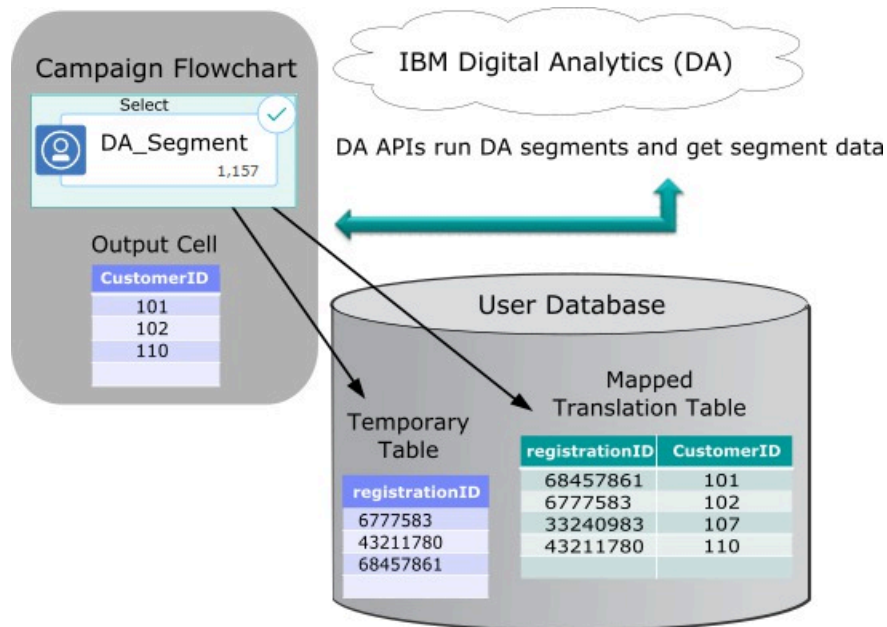


The following example shows Scenario 3b, Multiple binding unique keys in multiple tables. In this scenario, the Web data uses RegistrationID as its key, and a view is used to present data from multiple dimension tables. The combined view uses the audience identifier (CustomerID) as its key. The email address and several unique identifying data fields are used to bind the keys. As with all the examples, the translation table then uses the RegistrationID and CustomerID to identify individual records.



### Segment data is captured using API calls

The following illustration shows how a translation table maps selections between Unica Campaign and Digital Analytics. The IBM Digital Analytics segment data and related information are captured using API calls, for use in Unica Campaign flowcharts.



### Mapping the translation table

Map a translation table to make IBM Digital Analytics segments accessible in Unica Campaign. The table mapping identifies essential information such as the data source, table name and fields, and audience levels.

Before mapping a translation table, you must create and populate the table. For a complete list of tasks, see [Integrating IBM Digital Analytics and Unica Campaign \(on page 230\)](#).



Mapping a new base record table makes data accessible to processes in flowcharts. The instructions below explain how to map a new base record table for global use. You can also map a new base record table when you edit a flowchart, by using **Admin > Tables**.

1. Select **Settings > Unica Campaign Settings > Manage table mappings**.

For details, see [Mapping and unmapping user tables \(on page 67\)](#).

2. Optional: Save the mapping information in a table catalog for reuse. To make the information available for use in all flowcharts, store it in the default catalog (default.cat). Unica Campaign users can load the saved catalog to get the mappings.

For details, see [Table catalog administration \(on page 84\)](#).

3. If there is a change to the physical table (for example, if columns are added or deleted), you must remap the table. If you do not remap the table, flowcharts that use IBM Digital Analytics segments will return an error at run time indicating that the table schema has changed.



**Important:** When you map or remap a table, the **Unica Campaign Table name** that is assigned in the Table Definition wizard must exactly match the `TranslationTableName` that is defined in the Unica Campaign configuration settings. If you do not edit the table name when using the Table Definition wizard, the names will match. See [Campaign | partitions | partition\[n\] | Coremetrics \(on page 372\)](#).

Complete the remaining steps that are described in the topic [Integrating IBM Digital Analytics and Unica Campaign \(on page 230\)](#).

## Troubleshooting IBM Digital Analytics and Unica Campaign integrations

This topic provides troubleshooting information for setting up and using an integrated IBM Digital Analytics and Unica Campaign system.

### Troubleshooting Digital Analytics integration - Error 1714

This topic explains what to do if Error 1714 occurs when you open a Select process box in an Unica Campaign flowchart and select IBM Digital Analytics Segments.

Error 1714 occurs when you select **IBM Digital Analytics segments** in a Select process box.

The Unica Campaign backend listener server cannot access the `export.coremetrics.com` API URL due to a network connectivity issue. Therefore, the process box cannot list the Digital Analytics segments that were exported for use in flowcharts.

The server machine where the Unica Campaign listener is running must be able to reach the IBM Digital Analytics data center in order to communicate with the hosted Coremetrics URL (`https://export.coremetrics.com/eb/segmentapi/1.0/api.do`). Use port 443, which is the default port for https. Without direct access between the listener machine (which may sit behind a secure firewall) and the data center, the integration will not work.

## Troubleshooting Digital Analytics integration - Error 11528

**Symptoms:** This topic explains what to do if Error 11528 occurs during a Select process run when Digital Analytics segments are used as input.

**Cause:** Error 11528 occurs during a Select process run in an Unica Campaign flowchart. The error states that the SQL call failed and mentions a data type mismatch.

**Resolution:** This error can occur if the data type of the registrationid in the translation table does not match the data type that is defined in IBM Digital Analytics. The data type of the registrationid in the translation table does not match the data type that is defined for the registrationID in IBM Digital Analytics. For example, one might be NUMERIC but the other is VARCHAR.

If Digital Analytics segments are being used as input to the Select process, then you can resolve Error 11528 by changing the data type of the registrationid in the translation table so that it matches the data type that is defined in IBM Digital Analytics. For example, set them both to VARCHAR. For more information, read about creating and populating the translation table.

## Troubleshooting Digital Analytics integration - Error 13156

This topic explains what to do if error 13156 occurs when using the Digital Analytics Segment Selection pop-up window.

Unica Campaign users see Error 13156 when they try to select an Digital Analytics segment when configuring a Select process box in a flowchart. The error states: Error Received in @Digital Analytics Response. Refer to log for more details.

It is possible that the column name for the Digital Analytics IDs was not defined as registrationid in the mapped translation table and the **CoremetricsKey** configuration property in Unica Campaign was not set to registrationid. It is also possible that the credentials assigned to the UC\_CM\_ACCESS data source are incorrect. The UC\_CM\_ACCESS data source is the mechanism that Unica Platform uses to store the credentials that provide access to the integration service. To verify whether this is the case, increase the logging level of the flowchart log file to DEBUG. If the log file includes the following error, then it is an authentication issue: `{"error":{"message":"User authentication failed","code":"1000"}}}`.

To confirm that the translation table column which holds Digital Analytics IDs is called registrationid, choose **Settings > Configuration > Unica Campaign | partitions | partition[n] | Coremetrics** and confirm that **CoremetricsKey** is set to **registrationid**. To correct the data source credentials, choose **Settings > Users**, select the user that is defined in the **ASMUserForCredentials** configuration setting, click the **Edit Data Sources** link, and edit the data source:

- Ensure that the Data Source Login and Password use the same credentials as the Digital Analytics Client ID.
- Ensure that the Data Source exactly matches the **ASMDatasourceForCredentials** defined in the Unica Campaign configuration settings (for example, UC\_CM\_ACCESS).

## Troubleshooting Digital Analytics integration - Error 13169

This topic explains what to do if Error 13169 occurs during a Select process run.

Error 13169 occurs during a Select process run in an Unica Campaign flowchart.

There was a run failure on an IBM Digital Analytics segment. It is possible that a segment is not defined properly in IBM Digital Analytics.

Read the error carefully and take the appropriate action. For example, the error "There is no start date for the selected IBM Digital Analytics Segment" indicates an invalid date range.

Open the **IBM Digital Analytics segment selection** dialog from a Select process in an Unica Campaign flowchart and examine the segment definitions. The dialog shows segments that were defined in IBM Digital Analytics. You cannot change the segment definitions within Unica Campaign.

For example, the Start Date and End Date values come from IBM Digital Analytics. If a Start Date was not defined in IBM Digital Analytics, an administrator must correct the segment configuration in IBM Digital Analytics and republish the segment to Unica Campaign.

Ensure that the Segment range that is defined in the bottom of the dialog specifies a date range that falls within the Start and End Dates that are defined for the segment.

It can be helpful to turn on Debug level logging for the flowchart, run the process, then look at the flowchart log file. (Remember to revert to the default logging level after you are done, because Debug level logging can affect performance.)

## Troubleshooting Digital Analytics integration - "IBM Digital Analytics Segments" option is not available

This topic explains what to do if the IBM Digital Analytics link is not available in the Select process box in an Unica Campaign flowchart.

When a user opens the Select Process Configuration dialog in a flowchart, the Input list does not include **IBM Digital Analytics segments** as an option.

The configuration setting **UC\_CM\_integration** might not be enabled. Also, the user's permissions might not be set properly in Unica Campaign.

Enable the configuration: Choose **Settings > Configuration > Unica Campaign | partitions | partition[n] | server | internal** and set **UC\_CM\_integration** to Yes.

Grant user permission: Choose **Settings > User roles and permissions > Unica Campaign > Partition[n] > Global Policy**. Click **Add roles and assign permissions**, then **Save and edit permissions**. Under **Campaigns**, grant permission to **Access IBM Digital Analytics segments**.

## Troubleshooting Digital Analytics integration - Segments are not listed

This topic explains what to do if the **IBM Digital Analytics Segments selection** dialog does not list any segments.

In a Select process configuration dialog in an Unica Campaign flowchart, users open the Input list and click **IBM Digital Analytics segments**. After selecting a Client ID, no IBM Digital Analytics segments are listed.

The IBM Digital Analytics account did not publish any segments to Unica Campaign.

The Digital Analytics administrator must define segments in Digital Analytics and publish them for use in Unica Campaign:

---

In Digital Analytics, select **Manage > Report options > Report segments**. In the **Actions** menu, click the **Broadcast** icon next to the segment that you want to share with Unica Campaign. When opens, complete the segment form.

## Troubleshooting Digital Analytics integration - Number of records mismatch

When a flowchart runs, Unica Campaign detects if there is a mismatch between the number of IBM Digital Analytics keys and Unica Campaign audience IDs in the mapped translation table. A warning occurs if the number of registrationIDs does not match the number of audience IDs.

If a mismatch is detected, Unica Campaign writes a warning message to the flowchart log file asking you to verify that the mapped translation table contains updated records.

This behavior is intended to detect and prevent discrepancies between the IBM Digital Analytics keys and the corresponding Unica Campaign audience IDs in the mapped translation table. Consider the example where there are registration IDs in the IBM Digital Analytics segment which have not yet been added to the translation table because the ETL routines have not completed. In this case, you might have 100 customers from an IBM Digital Analytics segment, but only 95 CustomerIDs in Unica Campaign. The results are skewed now (100 records vs. 95 records) but they will be fine when the ETL routine completes.

To resolve this problem, (re)match the online and offline keys according to your corporate policy and repopulate the translation table with up-to-date data. The user must rerun the flowchart after the mapped translation table is updated.

## Overview of Opportunity Detect integration with Unica Campaign

When Opportunity Detect is integrated with Unica Campaign, data about customer transactions produced by Opportunity Detect can be used in your Unica Campaign flowcharts.

Opportunity Detect enables you to look for specified customer behaviors and patterns in your customer data. You define the transactions and patterns that Opportunity Detect looks for, and you specify the data that is written to the database when those criteria are met.

For example, you could configure Opportunity Detect to provide data about unusual purchase amounts or a decline in activity. You might want to use this data to target customers in a drip campaign for nurturing or retention.

Configuring the integration is described in the Unica Campaign Administrator's Guide. For more details on Opportunity Detect see the Opportunity Detect User's Guide and the Opportunity Detect Administrator's Guide.

## How to integrate Unica Campaign with Opportunity Detect

The integration between Unica Campaign and Opportunity Detect takes place at the data level; there is no integration of the user interface.

The feature that enables integration with Unica Campaign is the Opportunity Detect Expanded Outcome data source connector. The Expanded Outcome connector writes data to two database tables in a format that Unica Campaign can consume.

Opportunity Detect can process data in batches, or it can accept input data from a web service and function in a more interactive mode. This section includes usage examples for both batch and interactive modes.

### About the Expanded Outcome tables

The Expanded Outcome connector writes the Outcome data to two database tables, which you must create using scripts provided with Opportunity Detect.

DB2 is the only supported database type for the Expanded Outcome tables.

The tables are:

- A **primary** table that contains the text string specified in the **Message** field in the Action component.
- A **secondary** table that contains the data specified in the **Additional information** field in the Action component.

You provide a base name for the Expanded Outcome tables when you run the `ExpandedTable.sql` script to create the tables. The script appends the number 1 to the name of the primary table, and appends the number 2 to the name of the secondary table.

For example, if you specify the base name `ExpandedOutcome`, the script creates two tables: `ExpandedOutcome1` and `ExpandedOutcome2`.

### Fields in the Expanded Outcome tables

These descriptions of the fields in the Expanded Outcome tables refer to scalar and tabular values, which are defined as follows:

#### Scalar

A single unit of data.

#### Tabular

A data set, as in a database row. In Opportunity Detect Outcomes, tabular data is saved in XML format.

Depending on how you specify the Outcome data, the Outcome can contain either type of value, or both types. If you include tabular data in a Unica Campaign integration, additional processing is required before Unica Campaign can consume it.

**Table 42. Fields in the Expanded Outcome primary table**

Field	Description	Data type
OUTCOMEID	Unique sequence ID. Used as the primary key to link to the secondary Expanded Outcome table.	Integer
AUDIENCEID	ID of the audience member for which the trigger system fired. Examples of an audience are account, customer, or household. The audience ID is stored as a string. Multi-column audience IDs are not supported.	NVARCHAR(60)  If you use Oracle system tables and plan to integrate with Unica Campaign, you must change the data type of this field from NVARCHAR(60) to Varchar2(60) because Unica Campaign does not support the NVARCHAR(60) data type.
AUDIENCELEVEL	The single character audience code assigned on the Opportunity Detect Audience Levels page.	NVARCHAR(60)  If you use Oracle system tables and plan to integrate with Unica Campaign, you must change the data type of this field from NVARCHAR(60) to Varchar2(60) because Unica Campaign does not support the NVARCHAR(60) data type.
COMPONENTID	Unique ID of the Action component that fired to generate the Outcome.	Varchar
OUTCOMEDATE	The timestamp of the final event that caused the Action component to fire.	Timestamp
RUNID	ID of the run, for batch mode only. The Run ID helps distinguish between the Outcomes of one run versus the Outcomes of runs before or after it. Because of the Run ID, you do not need to truncate the Outcome table after every run because you can query the table for all of the Outcomes in a specific run.	Integer
MESSAGE	The text string that was specified in the <b>Message</b> field of the Action component.	NVARCHAR(60)  If you use Oracle system tables and plan to integrate with Unica Campaign, you must change the

**Table 42. Fields in the Expanded Outcome primary table (continued)**

Field	Description	Data type
		data type of this field from NVARCHAR(60) to Var- char2(60) because Unica Campaign does not sup- port the NVARCHAR(60) data type.
PROCESSED	A flag that indicates whether the da- ta has been consumed by Unica Cam- paign.	Integer

**Table 43. Fields in the Expanded Outcome secondary table**

Field	Description	Data type
OUTCOMEID	Unique sequence ID. Used as a foreign key to link the record to the primary Ex- panded Outcome table.	Integer
NAME	The name assigned in the <b>Additional Information</b> field of the Action compo- nent.	NVARCHAR(60)  If you use Oracle system tables and plan to inte- grate with Unica Campaign, you must change the data type of this field from NVARCHAR(60) to Var- char2(60) because Unica Campaign does not sup- port the NVARCHAR(60) data type.
VALUE	The scalar and tabular data that was specified in the <b>Additional Information</b> field of the Action component. Tabular values are saved in XML format.	Clob
DATATYPE	For scalar values, the data type can be one of the following.  <ul style="list-style-type: none"> <li>• boolean</li> <li>• currency</li> <li>• date</li> <li>• double</li> <li>• integer</li> <li>• string</li> </ul> For tabular values, the data type is set to string, because tabular values are	NVARCHAR(60)  If you use Oracle system tables and plan to inte- grate with Unica Campaign, you must change the data type of this field from NVARCHAR(60) to Var- char2(60) because Unica Campaign does not sup- port the NVARCHAR(60) data type.

**Table 43. Fields in the Expanded Outcome secondary table (continued)**

Field	Description	Data type
	stored in XML, and the data type for XML is string.	

### XML format of tabular values

Here is an example of the XML for a tabular value, where the record includes these fields:

- Field\_1
- Field\_2
- Field\_3

### Example

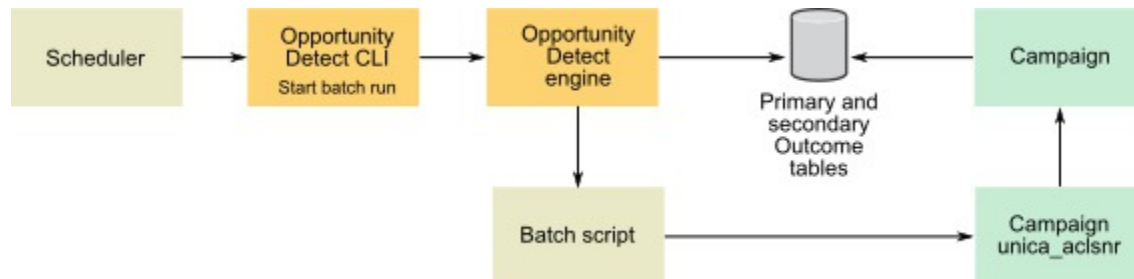
```
<SELECT name="S1">
<ROW>
<FIELD name="Field_1">abc</FIELD >
<FIELD name="Field_2">123.45</FIELD >
<FIELD name="Field_3">xyz</FIELD >
</ROW >
</SELECT >
```

## Integrating Opportunity Detect with Unica Campaign in batch mode

The following example illustrates how you might use the Expanded Outcome data in Unica Campaign, in batch mode.

You must have Unica Campaign and Opportunity Detect installed and running.

The following diagram illustrates the example described in this procedure.



1. Create the Expanded Outcome tables in your database using the script provided with Opportunity Detect.
2. On the Server Groups page in Opportunity Detect, do the following.



- If a database connection for the database where you created your Expanded Outcome tables does not exist, create one.
- If an Expanded Outcome data source connector does not exist, create one.

If you make the connector sharable, you can map the connector to your primary Expanded Outcome table on the Server Groups page or on the Deployment tab of the workspace. If you do not make the connector sharable, you can map it only on the Deployment tab.

3. Create the Opportunity Detect workspace and configure it to use the Expanded Outcome data source connector for Outcome data, either on the Server Groups page or on the Deployment tab of the workspace.
4. On the Deployment tab of the Opportunity Detect workspace, configure the deployment to call a batch file at the end of a successful run.

Create the batch script to call the Unica Campaign listener service, `unica_aclsnr`, to run a Unica Campaign flowchart that you design.

5. Use the Opportunity Detect command line utility, `RemoteControlCLI` (CLI), to run the workspace.

Use your own scheduling utility to run the CLI batch script at the desired interval; for example, daily.

When the workspace runs, Opportunity Detect inserts Outcome data into the Expanded Outcome tables.

6. Configure your Unica Campaign flowchart as follows.
  - a. In a Select process, create a new table mapping as follows.
    - Map your main audience in Unica Campaign to the OUTCOMEID field in the primary Expanded Outcome table. This is required so that you can select Outcome records for use in the flowchart. Selection must use the OUTCOMEID field, as the same AUDIENCEID field can be repeated in multiple Outcome records.
    - Map your alternate audience in Unica Campaign to the AUDIENCEID field in the primary Expanded Outcome table. This mapping defines the audience on which rest of the flowchart logic should be performed.



**Note:** If you plan to use Opportunity Detect Outcome data in multiple flowcharts, save the mapped table information into a table catalog and load this catalog in other flowcharts.

- b. Select records where the value in the `PROCESSED` field in the primary Expanded Outcome table is 0.

This value indicates that the record has not been processed yet.

- c. Set the value in the `PROCESSED` field in the primary Expanded Outcome table to 1, to indicate that the record has been processed.

You can write SQL in a Select process to set this value.

- d. In an Audience process, switch the audience from OUTCOMEID to AUDIENCEID.

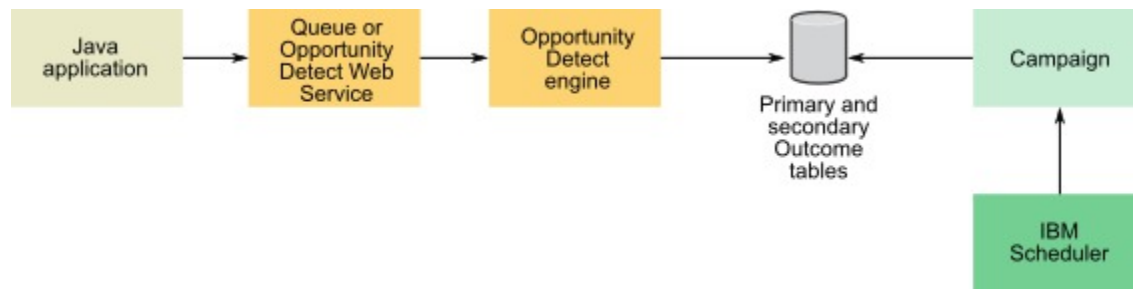
- e. Use the Opportunity Detect data as desired in your flowchart.
- f. Use a MailList process to assign an offer and update contact history.

## Integrating Opportunity Detect with Unica Campaign in interactive mode

The following example illustrates how you might use the Expanded Outcome data in Unica Campaign, in interactive mode.

You must have Unica Campaign and Opportunity Detect installed and running.

The following diagram illustrates the example described in this procedure.



1. Create the Expanded Outcome tables in your database using the script provided with Opportunity Detect.
2. Do one of the following.
  - If you plan to use a queue connector, configure a queue for your transaction data in your queue server.
  - If you plan to use the Web Service, develop the required Java™ classes.
3. On the Server Groups page in Opportunity Detect, do the following.
  - If a database connection for the database where you created your Expanded Outcome tables does not exist, create one.
  - If an Expanded Outcome data source connector does not exist, create one.

If you make the connector sharable, you can map the connector to your primary Expanded Outcome table on the Server Groups page or on the Deployment tab of the workspace. If you do not make the connector sharable, you can map it only on the Deployment tab.

4. Configure the Opportunity Detect workspace to use the Web Service or a queue data source connector for transaction data, and the Expanded Outcome data source connector for Outcome data.
5. Configure your Unica Campaign flowchart as follows.

- a. In a Select process, create a new table mapping as follows.
  - Map your main audience in Unica Campaign to the OUTCOMEID field in the primary Expanded Outcome table. This is required so that you can select Outcome records for use in the flowchart. Selection must use the OUTCOMEID field, as the same AUDIENCEID field can be repeated in multiple Outcome records.
  - Map your alternate audience in Unica Campaign to the AUDIENCEID field in the primary Expanded Outcome table. This mapping defines the audience on which rest of the flowchart logic should be performed.



**Note:** If you plan to use Opportunity Detect Outcome data in multiple flowcharts, save the mapped table information into a table catalog and load this catalog in other flowcharts.

- b. Select records where the value in the PROCESSED field in the primary Expanded Outcome table is 0.

This value indicates that the record has not been processed yet.

- c. Set the value in the PROCESSED field in the primary Expanded Outcome table to 1, to indicate that the record has been processed.

You could write SQL in a Select process to set this value.

- d. In an Audience process, switch the audience from OUTCOMEID to AUDIENCEID.

- e. Use the Opportunity Detect data as desired in your flowchart.

- f. Use a MailList process to assign an offer and update contact history.

6. Use your own scheduling utility or the Unica Scheduler to schedule flowchart runs at the desired interval; for example, every minute.

## Unica Campaign and Unica Journeys integration

Unica Campaign can integrate with Journey so that a continuous communication with users can be established based on the inputs from Campaign. Journey is more focused towards establishing a continuous dialog with users across multiple channels like email, sms and crm, and Campaign possesses rich segmentation capabilities. The Campaign segments or audience information can be pushed to Journey, thereby enabling a continuous customer dialog. In Campaign application, the new capability is added to publish the audience information to Journey. This is enabled through the Journeys process box in Campaign.

The following sections describe how this integration works.

### Overview

Campaign flowchart has capabilities to perform the segmentation. Using Journey process box Campaign send the audience information to Unica Journeys. Campaign sends the audience information on the designated kafka topic.

Unica Journeys read this audience information from the kafka topic and identifies it with entry sources. This audience information is further published to all Journeys, which are taking input from the mentioned entry source.

## Journey process box

Journey process box enables the marketers to streamline data flow between Campaign and Journey application. Here are the details.

- Journey process box allows the Campaign users to select the audience data to be sent to Journey (input cells).
- It allows the marketers to select the existing data definition from Journey.
- It enables the users to select the entry sources which exists in Journeys and are of type Campaign.
- The View Journey button available on Journey process box allows the users to view the Journeys associated with the selected entry source (ES) and data definition (DD). This Journey process box sends data to the associated Journeys.
- View Journey button is not enabled, unless users select ES and DD.
- On selecting data definition, the details of the selected data definition are populated under the Journey fields. This provides details of Journey fields like name, data type, mandatory / significant field on Journeys.
- Journey process box allows the users to map and generate output from the fields from Journey application as per the data definition selected.
- While executing the Journey process box, the audience information is sent as per the field mapping to Journey on kafka topic identified with entry source code.
- Once publishing of audience data is complete, Journey process box execution is marked as complete.
- The Campaign flowchart designer has access to view the entry source, data definition and only associated Journeys created.

### **Journey process box - Associated Journeys**

This tab lists all the associated journeys as per the selected entry source and data definition. This lists only the associated journeys. The Campaign user gets a quick view of the Journeys in which Campaign audience information is pushed.

### **Journey process box - Field mapping**

This allows the marketers to map Campaign fields which are pushed to Journey's entry source.

The left pane displays all available fields in Campaign including:

- Table mapping fields
- Campaign generated fields
- User variables
- Derived fields
- Extract fields

The users can create derived fields and profile the available fields.

The right pane display fields as per selected data definition. There is a field indicator ( \* ) to highlight the Journey field is mandatory.

It provides the ability to the users to map or match the fields. While mapping the fields, datatype of both side fields are not considered. Available field of any datatype can be mapped with journey field of any datatype. e.g. available field of String datatype can be mapped to Journey field of numeric datatype. Execution of Journey process box is successful even with field mappings of mismatched datatype. If the data format is not matched as per the data definition, then audience information would be discarded in Journey.

But mapping of all mandatory journey fields are required, otherwise, Journey process box configuration will not be complete.

### **Journey process box – General tab**

This tab has process box name and Notes field.

### **Delete Process box**

Delete process box deletes the process box. No associated journeys objects are deleted.

### **Copy Process box**

- The Process box copy operation creates a copy of process box. Copy operation is same at Campaign side.
- It does not create copies of Entry sources and data definitions associated.
- Copied process box retains all the information such as associated entry source, data definition, mapped field information, derived fields, or user variables, etc.

### **Sending Campaign/Flowchart information to Journey**

Journey process box allows the users to select and map data as per the data definition. It is important to let Journey know which Campaign or flowchart has sent the audience information to Journey. The Journeys data definition must be created along with CampaignName, CampaignCode, FlowchartName, etc, fields. While mapping the fields Campaign flowchart designer must map the identified fields with the data definition fields. New API is created from campaign to return flowchart information like Campaign Name, Campaign Code, FlowchartName, FlowchartId and Last Run date time. Any changes in this information will be updated as per the Run of respective flowchart.



**Note:** V12.1 FP3 onwards new API is created from campaign to return flowchart information.

### **Get Response from Journeys**

Campaign marketer would need to include Campaign / flowchart identification information along with the audience information being sent to Journey. This could include any Campaign generated information and audience information. User can pick up the required Campaign generated fields like (but not limited to)

- CampaignCode
- CampaignName
- FlowchartName
- FlowchartID
- CellName
- CellCode
- AudienceID
- RunID

Journey will maintain the audience information throughout Journey execution. In Journey flow, marketer can output the data using Publish touch point in flat file. Also, audience information can be outputted in to any database table in Journey application.

Campaign can read this information either from flat file or from respective database tables. Campaign/Flowchart information sent from Campaign would be available in this response. This will help marketer to identify the responses for respective audiences.

Please refer to Journeys User Guide for more information.

## Configuring Campaign web

Journey application gets registered with Marketing Platform, this registration details are used to connect Journey.

The following is the configuration required in Campaign for publishing the data on the Kafka topic. This configuration is specific to each partition.

Default configuration - `Affinium|Campaign|partition|partition1|Kafka|Journey`.

KafkaBrokerURL	Kafka server being used with Journeys application. Example is IP-0A862D46:9092
CommunicationMechanism	<p>Specify the connection mechanism to connect to Kafka server.</p> <p>Possible values: SASL_PLAINTEXT_SSL - Use this to connect to kafka with username/password and SSL enabled.</p> <p>NO_SASL_PLAINTEXT_SSL - Use this to connect kafka with no authentication and no SSL.</p> <p>SASL_PLAINTEXT - Use this to connect kafka with username and password only.</p> <p>SSL - Use this to connect kafka without username/password but with SSL.</p>

KafkaKeyFile	Specify the client key file if connection mechanism is using SSL. Example: /opt/Unica/Kafkakeys/client_key.pem
KafkaCertificateFile	Specify the certificate file if connection mechanism is using SSL. Example: /opt/Unica/Kafkakeys/client_cert.pem
CertificateAuthorityFile	It is signed certificate of Kafka Server, it is required when connection mechanism is using SSL. Example - /opt/Unica/Kafkakeys/ca-cert
UserForKafkaDataSource	Marketing Platform user contains the datasource credentials for Kafka while connecting with username / password
KafkaDataSource	DataSource containing the kafka user credentials.
TopicName	Journeys designated topic for Campaign to push data to Journey. Required value - STREAMING_IMPORT. Please do not change this as it would send data to Kafka topic which is not being used in Journey.

If these configurations are not available, Journey process box will get failed.

## Steps to generate client certificates to connect to Kafka

### Pre-requisites:

1. ca-cert, kafka.client.keystore.jks file should be generate at Journey side and use the same ca-cert & kafka.client.keystore.jks file while generating client\_key.pem & client\_cert.pem file for configuring Kafka at Campaign side.
2. Path to openssl executable must be set in environment variable.
3. Path to keytool executable must be set in environment variable. This you can find in java path.
4. On Cosole path to openssl config ie. openssl.cnf file MUST be set before going through following steps:

### Steps to generate Campaign side kafka certificates:

1. Generate client\_cert.pem file using following command:



**Note:** unica\*03 - This must match with the password value used whie generating 'kafka.client.keystore.jks' file.

```
#> keytool -noprompt -keystore kafka.client.keystore.jks -exportcert -alias localhost -rfc -storepass
unica*03 -file client_cert.pem
```

2. Generate client\_key.pem file using following command:



**Note:** unica\*03 - This must match with the password value used while generating 'kafka.client.keystore.jks' file.

```
#> keytool -noprompt -srckeystore kafka.client.keystore.jks -importkeystore -srcalias localhost  
-destkeystore cert_and_key.p12 -deststoretype PKCS12 -srcstorepass unica*03 -storepass unica*03
```

```
#> openssl pkcs12 -in cert_and_key.p12 -nocerts -nodes -passin pass:unica*03 -out client_key.pem
```

## Unica Campaign and Unica Link integration

Campaign application connects and exports data to external system (delivery channel) using pre-defined configuration done in process box and in Campaign configuration. To connect and export Campaign data to email, push, SMS, Campaign have separate process boxes for each delivery system. These integrations are tightly coupled with Campaign, which impose limits on configuring and integrating different delivery channels.

Unica Link has capabilities to create connectors which include the configuration and provisioning to connect to any delivery channel.

The Campaign-Link integration achieves the following:

- It enables Campaign to send data to customers' in-house developed delivery system, which sends email based on each custom configurations as per the client requirements.
- It can send data to CRM systems.
- Unica - Link has capability of AdTech Integration, which can be achieved via Facebook and LinkedIn connector



**Note:** V12.1 FP3 onwards AdTech Integration supports Facebook and LinkedIn connectors.

### Overview

- Unica Link can create Connectors. Connectors consist of configuration and provisioning to connect to any delivery channel. The out-of-the box connectors in Unica link are as follows:
  - MailChimp
  - Mandrill
  - Twilio
  - Salesforce CRM
  - Facebook
  - LinkedIn
- Unica Link has connections for the above mentioned connectors.
- Unica Campaign flowchart has a process box named Link.
- Link process box allows the users to create actions. There is a unique action associated with each process box.
- An action is associated with connections.



- Using Link process box, users can create or edit action.
- On executing a Link process box an action gets executed in Unica Link and an underlying execution starts in connectors.
- Campaign, internally triggers event tracking for selective action (mailchimp) based on connection and action properties specified (Activity fetch frequency, Activity fetch units, Fetch results end date).

As per these properties Unica Link tracks events.

## Manage Link Connections

The "Manage link connections" option will be visible under Campaign Settings, if the "linkInstalled" flag is enabled.

This flag can be configured from path: (Affinium|Campaign|partitions|partition1|server|internal)

Using "Manage Link Connections" configuration user can create, edit and delete Link connections. To perform this operation user is required to have Link roles and permissions.

Manage Link Connection can be found under:

Campaign Settings > Connectors -> Manage Link Connections

## Link Process Box

Link Process box is available in flowchart pellet. Link Process Box takes input from other process boxes like select, extract, segment, etc and does not provide output to downstream process boxes.

Here are the details:

1. Unica Link Administrator creates or configures connectors and connections in Unica Link.
2. Unica Link administrator communicates with delivery channel expert (e.g mailchimp) and identifies the following information:
  - a. Connection mechanism and connection details.
  - b. Identification and configuration of personalization fields, parameters, configurations, which are required by delivery channel.
  - c. Any other details regarding delivery channel APIs, authentication etc.
3. Based on the above information, Unica Link Administrator designs the connector.
4. Unica Link administrator also designs the connection from the different connectors created in Unica Link.
5. The Link process box renders the list of connections created in the Unica Link.
6. On selecting any of the connections, the action interface is rendered in process box which provisions the user to create an action in Unica Link.
7. Users can configure action properties. These are specific to delivery channel and can also map the personalization fields.
8. When process box is executed, mapped fields information and identity information is sent to Unica Link.
9. Unica link processes this audience information and performs the execution part like – sending email, SMS or CRM actions, etc.

10. Delivery status (success, failure, etc.) are sent back to Campaign and process box execution results are updated.
11. Unica Link keeps on tracking the events as per the connection/action properties configured. Based on that, connector records the contact and response data in flat file.



**Note:** To assign offer user can use maillist/CallList Process box. The exported output of maillist process box can be taken as input in downstream process box and send audience and offers information to Unica Link.



**Note:** For each successful run of Link Process Box, it shows the exported records count on Link Process Box tool tip.

## Link – Process box functionality

Link Process box is available in flowchart pellet. It takes input from other process boxes like select, extract segment, etc. It supports flowchart operations like annotations, Save, Delete, Copy, Configure process box, Run. New process box takes input from multiple input cells.



**Note:**

1. Link process box is not configured when it is pasted within flowchart or pasted from template library as it does not copy the underlying action in Unica Link.
2. If user deletes a campaign or flowchart or Link process box, it will not delete the corresponding action in Unica Link or does not clean up the action deployment from Unica Link Runtime Server.

The following tabs are available in Link Process box interface.

- Link Configuration tab
  - Input cells – This allows the Campaign marketers to select the input for the Link process box. Users can select input from multiple process boxes.
  - When process box is dragged, and opened for the first time - Link Configuration tab opens iframe to render Unica Link connections listing page.
  - Users can select the connection and navigate to Actions page.
  - The Personalization screen allows the Campaign marketers to map the Delivery channel fields with the Personalization fields from Campaign table mapping.
  - Action is created for each process box. Re-opening the process box opens the same action in edit mode.
- General tab:

- Name of process box
- Description



**Note:** Description or notes for this process box.

## Configuring Campaign web

The Link configuration is available per partition. One Unica Link instance can be configured with one Campaign partition. The following are the configuration parameters for Link. `Affinium|Campaign|partitions|partition1|Link`.

<code>LinkConnectionURL</code>	This configuration parameter is used to specify the Link Connections URL. Campaign application uses this URL to retrieve connections / action details.
<code>LinkProjectName</code>	This is the Campaign project name configured in Unica Link. This project contains the Campaign configurations like authentication, field mapping URL, etc.
<code>Application</code>	Application name used in Unica Link for Campaign integration
<code>AsmUserForLink-Credentials</code>	The Unica Platform user which stores the Unica Link credentials data source.
<code>DataSourceWith-LinkCredentials</code>	Data source name which holds the username and password to authenticate with <code>LinkConnectionURL</code> .
<code>LinkSharedLocation</code>	Location on Campaign Listener machine – which must be mounted on the HIP REST server on path mentioned under “ <code>HIPRestServerSharedLocation</code> ”.
<code>LinkRuntime-ServerURL</code>	Link runtime (HIP REST) server URL.

## Unica Campaign and ContactCentral Integration

In marketing world every communication reaching out to customer should have options for recipients to unsubscribe and setup their communication preferences.

ContactCentral is allowing users to setup the communication preferences by specifying the **OptIn** and **OptOut** for the various communication channels supported in Unica applications. Channels supported for customer consents:

- Email
- SMS
- Push
- WhatsApp

Configurations steps for Campaign to integrate with ContactCentral:

- Configure ContactCentral data source under CAMPAIGN\_HOME/partitions/partitionN/DataSources. Name must be ContactCentral. Point ContactCentral data source to Contact Central DB. User needs to create ODBC connection, similar to Campaign database ODBC connection. Consult Contact Central team for supported System databases.
- Configure ContactCentral data source under **Setting > Users**. For example, asm\_Admin user. User need to give user name and password for ContactCentral datasource. Similar to UA\_SYSTEM\_TABLES.
- Stop Listener and Delete unica\_aclsnr.ldb, unica\_aclsnr.pid and CAMPAIGN\_HOME/partitions/partition1/conf/unica\_tbmgr.cache and start the Listener.
- Logout and Login into Unica application, so that the required tables mapping is reflected.
- Create or open any existing Deliver PB to apply ContactCentral consents to exclude opt-outs from Communication Channel.
- User needs to create first Regional preferences in ContactCentral application to avail this feature on Campaign Summary. User can set Regional preference in Campaign Summary. Also, user can see selected respective Regional preference value in Deliver PB.
- To enable this Regional preferences parameter in Campaign configuration

Goto Settings for '**internal**'

```
(Affinium|Campaign|partitions|partition1|server|internal)
```

**contactCentralEnabled** - No/Yes

- **Default value** - No

## Unica Campaign integration with Audience Central and Segment Central

Unica Audience Central is a centralized application to create and maintain audience levels that can be targeted by a campaign. In Unica release 12.1.4, Unica Audience Central integrates with Unica Campaign and Unica Interact.

Segment Central is a centralized application to create and manage strategic segments. In Segment Central we can create strategic segments based on audiences having base tables. Using base table mapping you can create rules, based on these rules the set of ids present inside each base table forms a segment. Segments will select records from the existing base table based on the defined rule. Segments can also be created using existing segments.

Settings for 'Unica Platform' (Affinium|suite)

set **Enable Audience Central management?** = Yes Based on the Audience role assigned to Platform user - new Audience Central menu is available under marketing Central Menu. If user disables the integration - new Audience Central menu disappears

set **Enable Segment Central management?** = Yes Based on the the value of flag there are various changes related to Segment in Unica Campaign. Based on the Segment role assigned to Platform user - new Segment Central menu is available under marketing Central Menu. Existing Segments menu under Campaign will not be available. If user disables the integration - new Segment Central menu disappears and existing Campaign > Segments menu appears.

For More information Refer Audience Central and Segment Central User Guides

## Unica Campaign integration with Callout Process Box

Callout process box will be available to users. Callout is a generic Process Box which enable users to plug in custom logic based on a pre-defined interface. The main purpose of Callout Process Box is to connect with any third-party custom logic written by customer, based on provided interface. Callout PB functionality comes with a light weight, secure and robust spring boot application which is used to keep track of multiple callouts across flowcharts.

### Platform Configuration:

A new node named ExternalCallout has been introduced under Affinium|Campaign|partitions|partition

It consists of below properties:

**CalloutApplicationURL** : URL of the external-callout application (eg: http://localhost:9003)

**AsmUserForCalloutSecret** : User with which we would be configuring the data source

**DataSourceWithCalloutSecret** : Datasource name

Other than these properties ExternalCallout has a template named ExternalCalloutTemplate at child level. Admin needs to use this template to configure their custom callouts.

The template has below fields:

**New category name** : Callout Name (used to identify the callout further in Callout PB)

**jarPath** : JAR location of Customer's implementation of callout.

**mainClass** : Implementation class name (this is the class name inside JAR , which has implemented the ExternalModellerCallout interface)

**Config parameters**: Config parameters serve as inputs for the callout, allowing users to provide comma-separated key-value pairs. These parameters are integral to the external callout's implementation and can be accessed within the user's code. It's important to note that any addition or update of parameters necessitates restarting the callout application.

Example: value1=test, value2=demo, value3=300

In the above example, three parameters (value1, value2, and value3) are specified. These parameters become accessible in the callout's implementation, facilitating dynamic customization.



**Note:** As a good practice ensure each key has a unique name and avoid special characters.

**dataFileLocation** : Directory name where callout PB's input & output files would get created (Base location would be: InstallationDirectory\Campaign\partitions\partition\externalCallout\data)

So if user specifies SampleCallout as the dataFileLocation the files for this particular callout

will be created under (InstallationDirectory\Campaign\partitions\partition\externalCallout\data\SampleCallout)

**Mapping Fields** : When creating a callout, you have an option to define a template for mapping fields. These fields serve as inputs to your callout's implementation and are accessible on the Callout PB for mapping.

The template has below fields:

- New Category Name: Assign a category name for identification purposes in the configuration.
- fieldName: Specify the field name that users expect in their callout. This field is case-sensitive and will be available for mapping in Callout PB.
- fieldType: Define the data type of the field, which can be either "Text" or "Numeric."

### Datasource Configuration

Go to users and select a user which was provided in property **AsmUserForCalloutSecret**. Add a new datasource to user as mentioned in property "DataSourceWithCalloutSecret" & use the secret key as password.

Format of Secret key should be like below :

- Secret key size should be greater than or equal to 256 bits
- Eg : A11cx@#g91@A91zN@@@x103

The secret key should be provided for attribute "callout.campaign.key" later while running the command for starting callout application.

### File System

After Campaign 12.1.5 is successfully installed, user can see a new folder named externalCallout created under InstallationDirectory\Campaign. User needs to manual copy this folder and place in the respective partitions, in which they want to use Callout PB's functionality. For example: If user intends to run Callout PB in partition1, they would need to copy the externalCallout folder to InstallationDirectory\Campaign\partitions\partition(1).

Samples: This folder consists of the interface ExternalModellerCallout which users need to implement. Also, this folder consists of some sample code for users to get an idea of how to implement the given interface. Users can also use the TestCallout.jar (which outputs card category based on salary) provided in samples folder to get an idea of the entire flow. testInput.csv is a sample file to be used as input while running TestCallout.jar.



**Note:** The sample codes are just for reference and not intended to be used in production.

**external-callout.jar:** The spring boot application responsible of handling and keeping track of status of multiple callouts at once.

To run external-callout.jar use below command:

**java -Dcallout.campaign.key=<DatasourcePassword> -jar external-callout.jar**

This command expects a system property named `callout.campaign.key` whose value should be similar to the password of the DataSource name (`DataSourceWithCalloutSecret`) we gave in the platform configuration (`Affinium|Campaign|partitions|partition|`).

**application.properties.** : This file is used to configure Campaign URL, port on which we need to run this spring boot app, and partition name Campaign URL is what you configured under `Affinium|Campaign|navigation|serverURL`

# Chapter 17. Unica Campaign integration with Kafka

Kafka instance configuration template - `Affinium|Campaign|partitions|partition[n]|Kafka|(KafkaTemplate)`

This template provided to add different Kafka instances for Campaign under selected partition.

`Affinium|Campaign|partitions|partition[n]|Kafka|Campaign` is the default configuration created for Campaign while installation.

`Affinium|Campaign|partitions|partition[n]|Kafka|Journey` is the default configuration created for Journey while installation.

Below are the configuration parameters for `KafkaTemplate`:

KafkaBrokerURL	URL of the Kafka instance user wants to point and produce data on it. Kafka server being used with Journeys application. Example is IP-0A862D46:9092
CommunicationMechanism	Specify the connection mechanism to connect to Kafka server.  Possible values:  SASL_PLAINTEXT_SSL - Use this to connect to kafka with username/password and SSL enabled.  This is also utilized for connecting with Confluent Kafka.  NO_SASL_PLAINTEXT_SSL - Use this to connect kafka with no authentication and no SSL.  SASL_PLAINTEXT - Use this to connect kafka with username and password only.  SSL - Use this to connect kafka without username/password but with SSL.
SaslMechanism	SASL mechanism used for client connections.  Possible values:  PLAIN: This is default value, Use this if client connection is without Kerberos authentication.  Choose this option for connecting with Confluent Kafka.  GSSAPI: Use this if client connection is with Kerberos authentication.



KafkaKeyFile	Specify the client key file if connection mechanism is using SSL. Example: /opt/ Unica/Kafkakeys/client_key.pem
KafkaCertificateFile	Specify the certificate file if connection mechanism is using SSL. Example: /opt/ Unica/Kafkakeys/client_cert.pem
CertificateAuthorityFile	<p>It is signed certificate of Kafka Server, it is required when connection mechanism is using SSL. Example - /opt/Unica/Kafkakeys/ca-cer</p> <p>For Confluent Kafka, alongside the key and secret, a root CA certificate is necessary for communication. Reach out to your Confluent Kafka service provider to obtain the root CA certificate. Once acquired, place this certificate on the listener server, and provide the full path to the certificate file here.</p> <p>Sample steps to get Confluent Kafka hosted on Google Cloud Platform (GCP) is listed below the Kafka Campaign Nodes Configuration details table.</p>
UserForKafkaDataSource	Marketing Platform user contains the datasource credentials for Kafka while connecting with username / password. Also, for Confluent Kafka, create a datasource under the Marketing Platform user.
KafkaDataSource	DataSource containing the kafka user credentials. When setting up this Kafka datasource for Confluent Kafka, utilize the key and secret as the username and password.
TopicName	<p><b>CAMPAIGN_PB</b> - Default value for Campaign Kafka Instance created. User can change this default topic name while configuring process box.</p> <p>STREAMING_IMPORT - Default value for Journey Kafka Instance created. Journeys designated topic for Campaign to push data to Journey. Required value - STREAMING_IMPORT. Please do not change this as it would send data to Kafka topic which is not being used in Journey.</p> <p>For KafkaTemplate the field Topicname is blank. User can set this default topic name while configuring Kafka instance.</p>

NumberOfPartitions	Number of partitions supports Kafka to hold user exported data. For Confluent Kafka, the partitions for the topics being created are predefined at the broker level. Obtain this number from your Confluent Kafka service provider.
NumberOfReplicas	Each partition is replicated across a configurable number of servers for fault tolerance. For Confluent Kafka, the replication factor for the topics being created is predefined at the broker level. Obtain this number from your Confluent Kafka service provider.
RetentionPeriodInSeconds	The maximum time Kafka will retain messages exported over topic. Once retention period over Kafka clears all eligible exported messages.
SslKeyPasswordDataSource	If KafkaKeyFile is password protected, then please create a separated data source which will include that password. User name is not used so can be anything. Mention that data source name as value of this field.
SaslKerberosServiceName	When using GSSAPI i.e., Kerberos authentication, configure a service name that matches the primary name of the Brokers configured in the Broker JAAS file.  Example is: kafka
SaslKerberosKeytabPath	When using GSSAPI i.e., Kerberos authentication, set path of the keytab file created for kafka client.
SaslKerberosPrincipal	When using GSSAPI i.e., Kerberos authentication, set Kerberos principal created for kafka client.

To obtain the root CA certificate for Confluent Kafka hosted on Google Cloud Platform (GCP), you can typically download it from the Confluent Cloud Console. The root CA certificate is used for secure communication between your client applications and Confluent Cloud Kafka.

Here are the general steps to retrieve the root CA certificate:

1. Log in to Confluent Cloud Console:
  - Go to the Confluent Cloud Console at <https://confluent.cloud> and log in with your credentials.
2. Select Your Environment:
  - If you have multiple environments, select the environment for which you need the root CA certificate.
3. Navigate to Clusters:
  - From the left navigation pane, click on **Clusters**.
4. Select Your Cluster:
  - Click on the name of the Confluent Cloud cluster for which you want to obtain the root CA certificate.

5. Go to **Settings** Tab:
  - Within the cluster details, navigate to the **Settings** tab.
6. Download the Root CA Certificate:
  - Look for a section related to security or certificates. There should be an option to download the root CA certificate.
7. Follow the Instructions:
  - Follow any instructions provided on the Confluent Cloud Console to download and securely manage the root CA certificate.
8. Use the Root CA Certificate:
  - Once downloaded, you can use the root CA certificate in your client applications that connect to Confluent Cloud Kafka. Configure your Kafka clients to trust this root CA certificate for secure communication.

Remember to handle the root CA certificate securely, as it is a critical piece for establishing secure connections. Always keep it confidential and avoid exposing it in public repositories.

If you encounter difficulties or if the process has changed, refer to the official Confluent Cloud documentation or contact Confluent Cloud support for the most accurate and up-to-date information.

## Campaign|partitions|partition[n]|Kafka| (KafkaTemplate)

This category contains properties of template to configure different Kafka instances. Campaign is the node created under Campaign|partitions|partition[n]|Kafka is the default created by Installer. More Kafka instances can be added using this template, so user can produce data to that Kafka Instance using Campaign process boxes like Snapshot, Maillist and Calllist.

### **KafkaBrokerURL**

#### **Description**

URL of the Kafka instance user wants to point and produce data on it. Kafka server being used with Journeys application. Example is IP-0A862D46:9092

### **SaslMechanism**

#### **Description**

SASL mechanism used for client connections.

#### **Possible values:**

PLAIN: This is default value, Use this if client connection is without Kerberos authentication.

Choose this option when connecting to Confluent Kafka.

GSSAPI: Use this if client connection is with Kerberos authentication.

## CommunicationMechanism

### Description

Specify the connection mechanism to connect to Kafka server.

### Possible Values:

SASL\_PLAINTEXT\_SSL - Use this to connect to kafka with username/ password and SSL enabled.

This is also utilized for connecting to Confluent Kafka.

NO\_SASL\_PLAINTEXT\_SSL - Use this to connect kafka with no authentication and no SSL.

SASL\_PLAINTEXT - Use this to connect kafka with username and password only.

SSL - Use this to connect kafka without username/password but with SSL.

### Default value:

NO\_SASL\_PLAINTEXT\_SSL

## KafkaKeyFile

### Description

Specify the client key file if connection mechanism is using SSL.

Example: /opt/Unica/Kafkakeys/client\_key.pem

## KafkaCertificateFile

### Description

Specify the certificate file if connection mechanism is using SSL.

Example: /opt/Unica/Kafkakeys/client\_cert.pem

## CertificateAuthorityFile

### Description

It is signed certificate of Kafka Server, it is required when connection mechanism is using SSL.

For Confluent Kafka, alongside the key and secret, a root CA certificate is necessary for communication.

Reach out to your Confluent Kafka service provider to obtain the root CA certificate. Once acquired, place this certificate on the listener server, and provide the full path to the certificate file here.

Example - /opt/Unica/Kafkakeys/ca-cert

## UserForKafkaDataSource

### Description

Marketing Platform user contains the datasource credentials for Kafka while connecting with username / password. Also, for Confluent Kafka, create a datasource under the Marketing Platform user.

## KafkaDataSource

### Description

DataSource containing the kafka user credentials.

When setting up this Kafka datasource for Confluent Kafka, utilize the key and secret as the username and password.

## TopicName

### Description

This is the default topic name set at configuration level. User can change this default topic name at process box level, while configuring the process box apart from Journey process box.

For **KafkaTemplate** the Topicname is blank. User can set default topic name while configuring Kafka instance.

### Default value:

#### **CAMPAIGN\_PB**

Default value for Campaign Kafka Instance which is created by the installer. User can change this default topic name while configuring process box.

#### **STREAMING\_IMPORT**

Journeys designated topic for Campaign to push data to Journey.



**Note:** For Journey PB please do not change the default topic name, as it would send data to Kafka topic which is not being used in Campaign.

## NumberOfPartitions

### Description

Number of partitions supports Kafka to hold user exported data.

For Confluent Kafka, the partitions for the topics being created are predefined at the broker level. Obtain this number from your Confluent Kafka service provider.

## NumberOfReplicas

### Description

Each partition is replicated across a configurable number of servers for fault tolerance.

For Confluent Kafka, the replication factor for the topics being created is predefined at the broker level. Obtain this number from your Confluent Kafka service provider.

### **RetentionPeriodInSeconds**

#### **Description**

The maximum time Kafka will retain messages exported over topic. Once retention period over Kafka clears all eligible exported messages.

### **SslKeyPasswordDataSource**

#### **Description**

If KafkaKeyFile is password protected, then please create a separated data source which will include that password. User name is not used so can be anything. Mention that data source name as value of this field.

### **SaslKerberosServiceName**

#### **Description**

When using GSSAPI i.e., Kerberos authentication, configure a service name that matches the primary name of the Brokers configured in the Broker JAAS file.

Example is: kafka

### **SaslKerberosKeytabPath**

#### **Description**

When using GSSAPI i.e., Kerberos authentication, set path of the keytab file created for kafka client.

### **SaslKerberosPrincipal**

#### **Description**

When using GSSAPI i.e., Kerberos authentication, set Kerberos principal created for kafka client.

# Chapter 18. Unica Campaign listeners

The listener is a key component of Unica Campaign. It provides an interface between front-end clients and back-end analytical server processes.

## Definition of listener terms

The following terms are used when discussing Unica Campaign listeners and listener clustering.

Term	Definition
Back-end	The components and communication related to the Unica Campaign listener and its interactions with other back-end server processes.
Cluster	A listener cluster is a set of two or more listeners that act as a unit to provide load balancing and high availability with minimal down-time. Clustered nodes provide continued service when system components fail. Unica Campaign listener clusters are Active-Active, meaning each node serves requests using a load-balanced approach.
Failover	Automatically switching to an alternate node in a cluster.
Front-end	The components and communication related to the Unica Campaign web application that provides the user interface.
High availability (HA)	A continuously operational system or component.
Listener	A server process that provides an interface to back-end analytical server processes. This interface is used by clients (such as the Unica Campaign web application and the Unica Campaign Server Manager) to connect to back-end servers. Each listener spawns processes to handle user and flowchart interactions. The listener is sometimes called the analytic server.
Load balancer	A component of the Unica Campaign master listener whose purpose is to coordinate load balancing among clustered listener nodes.
Master listener	The listener that controls the coordination of clustered nodes. Every cluster has one master listener. Any node in the cluster can act as the master listener. The master listener includes the load balancer component.
Node	Each listener in a cluster. Any node in the cluster, including the master listener, can serve a request from the web application.
Load based routing	A load balancing algorithm that distributes traffic proportionally to servers based on the user-specified ranking (weight) of each server.
ListenerType	Starting Campaign 12.0, there is no separate listener for Optimize, since it is integrated into Campaign listener.

Term	Definition
	<p>Campaign Listener (<code>unica_aclsnr</code>) can handle not only flowchart execution, but also Optimize session execution, depending on the ListenerType parameter value. There are 3 ListenerType parameter values as follows.</p> <p>"1" - This is CAMPAIGN_ONLY type of listener which is able to execute only flowcharts.</p> <p>"2" - This is OPTIMIZE_ONLY type of listener which is able to execute only optimize sessions.</p> <p>"3" - This is BOTH type of listener which is able to execute flowcharts as well as optimize sessions.</p>

## Front-end and back-end components

Unica Campaign consists of two main components.

- On the front end: The Unica Campaign web application provides the user interface. Users access this J2EE component through a web browser.
- On the back end: The Unica Campaign listener provides an interface between front-end clients (such as the Unica Campaign web application and the Unica Campaign Server Manager) and back-end analytical server processes. You can have a single-node or clustered listener configuration.

The Unica Campaign web application (front-end) and the listener (back-end) communicate over TCP/IP to handle requests and process transactions.

A listener is a `unica_aclsnr` process. Each `unica_aclsnr` process spawns a separate Unica Campaign server process (`unica_acsvr`) for each login and each active flowchart. For example, if one user logs in and then opens a flowchart, the listener spawns two instances of `unica_acsvr`.

For every Optimize session execution `unica_aclsnr` process spawns a Optimize server process (`unica_acosvr`).

Multiple listeners can be configured to perform as a cluster. In a clustered configuration, one listener acts as the master listener to coordinate incoming requests to the clustered nodes.

## Unica Campaign Listener (`unica_aclsnr`)

The Unica Campaign listener (`unica_aclsnr`) allows clients such as the Unica Campaign web application to connect to back-end analytical server processes.

Before users who log in to Unica can work with any Unica Campaign features, the Unica Campaign listener must be running and the Unica Campaign web application must be deployed and running.

The listener automatically spawns a separate `unica_acsvr` process for each login and each active flowchart. For example, if one user logs in and then opens a flowchart, the listener spawns two instances of `unica_acsvr`.

For every Optimize session execution `unica_aclsnr` process spawns a Optimize server process (`unica_acosvr`).



You can start and stop the listener manually or automatically.

To have the Unica Campaign server start automatically with the system on which Unica Campaign is running:

- If Unica Campaign is installed on a Windows™ server, set up the listener as a service. See [Installing the Unica Campaign listener as a Windows service \(on page 284\)](#) for details.
- If Unica Campaign is installed on a UNIX™ server, set up the listener as part of the `init` process. See your UNIX™ distribution's documentation for information about setting up the `init` process.

## Unica Campaign listener requirements

The Unica Campaign listener requires that Unica Platform be running.

The listener connects to Unica Platform using the value of the `configurationServerBaseURL` property in the `config.xml` file, located in the `conf` directory of your Unica Campaign installation. Typically, this value is `http://hostname:7001/Unica`. If Unica Platform is not running, the Unica Campaign listener will fail to start.

As the listener is dependent on Unica Platform to start successfully, you should ensure that your web application server is running and the Unica Platform web application is deployed before starting the listener.

## Unica Campaign listener syntax and options

Use these options to install or uninstall `unica_aclsnr` as a Windows™ service, do a flowchart recovery run, or display the listener version.

Use the following syntax with the `unica_aclsnr` command:

```
unica_aclsnr {[-a] | [-i]} {[ -n | [-r]} [-d <service_dependencies>] [-u] [-v]
```

The `unica_aclsnr` utility supports the following options:

**Table 44. Unica Campaign listener options**

Option	Description
-a	This option installs the listener as a Windows™ service with auto-restart functionality. If the listener process fails to start or if it stops unexpectedly, this option tries to restart it automatically. It makes two retry attempts within a specified time period. This option is supported for both single-node and clustered listener configurations.
-i	This option installs the listener as a Windows™ service without auto-restart functionality. If the listener is unavailable, it will not be restarted.
-r (the default)	This option starts a recovery run by forcing the listener to find and register any running flowcharts. Use this parameter if for some reason the listener goes down and flowcharts ( <code>unica_aclsvr</code> processes) are still running. The listener stores flowchart information in a text file ( <code>unica_aclsnr.udb</code> ). When you use <code>-r</code> , the listener checks the <code>.udb</code> file for running flowcharts and re-establishes the connections.

**Table 44. Unica Campaign listener options (continued)**

Option	Description
	Even if the running flowchart processes (flowchart and branch production runs only) went down with the listener, the listener reloads those flowcharts and resumes running them from the last saved checkpoints.
-n	The opposite of <code>-r</code> . This option prevents the listener from checking the <code>unica_acslnr.udb</code> file.
-d	[ <code>-d &lt;service_dependencies&gt;</code> ] is an optional argument that tells the Microsoft™ Windows™ operating system to wait until the services in <code>&lt;service_dependencies&gt;</code> are completely started before launching the Unica Campaign listener. The most common use case is when the web application server that runs Unica Campaign is also installed as a service, because the web application server must be fully launched and running before you launch the Unica Campaign listener. To specify multiple services, use a comma-separated list. Use the Service name defined in Windows™ Services.
-u	This option uninstalls the listener as a service (Windows™ only).
-v	This option displays the current version of the listener.

## Configuration settings for a single-node listener configuration

The configuration properties for single-node listener environments are set automatically during installation or upgrade. However, you can adjust them by choosing **Settings > Configuration**.

The purpose of this topic is to identify the configuration properties that pertain to a single-node listener configuration. For details about configuration, see the appropriate topic for each configuration setting.

The following configuration options pertain to a single-node listener configuration:

- **Campaign|unicaACLlistener**: Use this category, and only this category, to define configuration settings for an unclustered listener environment. Properties include: `enableWindowsImpersonation`, `enableWindowsEventLogging`, `logMaxBackupIndex`, `logStringEncoding`, `systemStringEncoding`, `loggingLevels`, `maxReuseThreads`, `threadStackSize`, `logMaxFileSize`, `windowsEventLoggingLevels`, `useSSL`, `keepalive`.
- **Campaign|campaignClustering**: Set `enableClustering` to `FALSE`. This causes all other properties in this category to be ignored, because they do not apply to a single-node configuration.
- **Campaign|unicaACLlistener|node[n]**: An unclustered listener configuration should not have any nodes under this category. Nodes are created and used only for clustered listener configurations.
- **Campaign|partitions|partition[n]|server|flowchartSave**: It is a best practice to configure `autosaveFrequency` and `checkpointFrequency`. You can override these global settings by editing a flowchart and choosing **Admin > Advanced** to set **Auto-save (during user configuration)** and **Checkpoint (during flowchart execution)**.


## Configuration settings for a clustered listener configuration

The configuration properties for clustered listeners are set automatically during installation. However, you can adjust them by choosing **Settings > Configuration**.

The purpose of this topic is to identify the configuration properties that pertain to a clustered (multiple-node) listener configuration. For details about configuration, see the appropriate topic for each configuration setting.

After you make cluster configuration changes, use the `unica_svradm` utility Refresh command to inform the master listener of the change.

The following configuration options pertain to a clustered listener configuration:

- **Campaign|campaignClustering:** These properties pertain to the cluster as a whole. Set `enableClustering` to TRUE, then set all remaining properties in this category: `masterListenerLoggingLevels`, `masterListenerHeartbeatInterval`, `webServerDelayBetweenRetries`, `webServerRetryAttempts`, `campaignSharedHome`.
  - **Campaign|unicaACLlistener|node[n]:** Configure an individual child node for each listener in the cluster. If `enableClustering` is TRUE, you must configure at least one child node or an error will occur during startup. The following properties are available for each listener node: `serverHost`, `serverPort`, `useSSLForPort2`, `serverPort2`, `masterListenerPriority`, `loadBalanceWeight`, `listenerType`.
  - **Campaign|unicaACLlistener:** The following properties pertain to all listener nodes in the cluster: `enableWindowsImpersonation`, `enableWindowsEventLogging`, `logMaxBackupIndex`, `logStringEncoding`, `systemStringEncoding`, `loggingLevels`, `maxReuseThreads`, `threadStackSize`, `logMaxFileSize`, `windowsEventLoggingLevels`, `useSSL`, `keepalive`.
-  **Important:** The following **Campaign|unicaACLlistener** properties are ignored when `enableClustering` is TRUE: `serverHost`, `serverPort`, `useSSLForPort2`, `serverPort2`. Instead, use **Campaign|unicaACLlistener|node[n]** to set these properties for each individual node.
- **Campaign|partitions|partition[n]|server|flowchartSave:** It is a best practice to configure **autosaveFrequency** and **checkpointFrequency**. You can override these global settings by editing a flowchart and choosing **Admin > Advanced** to set **Auto-save (during user configuration)** and **Checkpoint (during flowchart execution)**.

## Listener clustering

Clustering involves the use of multiple listeners for the purpose of high availability and load balancing.

Clustered listeners ensure that failover occurs automatically from one machine to another. Additionally, clustered listeners provide parallel processing and load balancing, for improved performance.

Listener clustering, also referred to as back-end clustering, is important because flowchart runs happen at the back end. Flowchart runs create and update contact history, offer history, and other configured tables.

When multiple listeners are configured as a cluster, the front-end web application communicates over TCP/IP with all listener nodes. Within the cluster itself, one node acts as the master listener, and is responsible for performing load balancing of client requests across the nodes.

Listener clustering provides the following benefits:

- **Stability:** Multiple listeners run in parallel across multiple machines in the cluster.
- **Load balancing:** The back-end load is shared across listener nodes, using weighted round robin to distribute the load.
- **Failover:** If a listener goes down due to hardware, software, or network failure, failover occurs automatically, minimizing disruption.
- **Scalability:** Additional nodes can be added to run additional listeners.

## Diagram of listener clustering

This diagram illustrates a three-node listener cluster configuration.

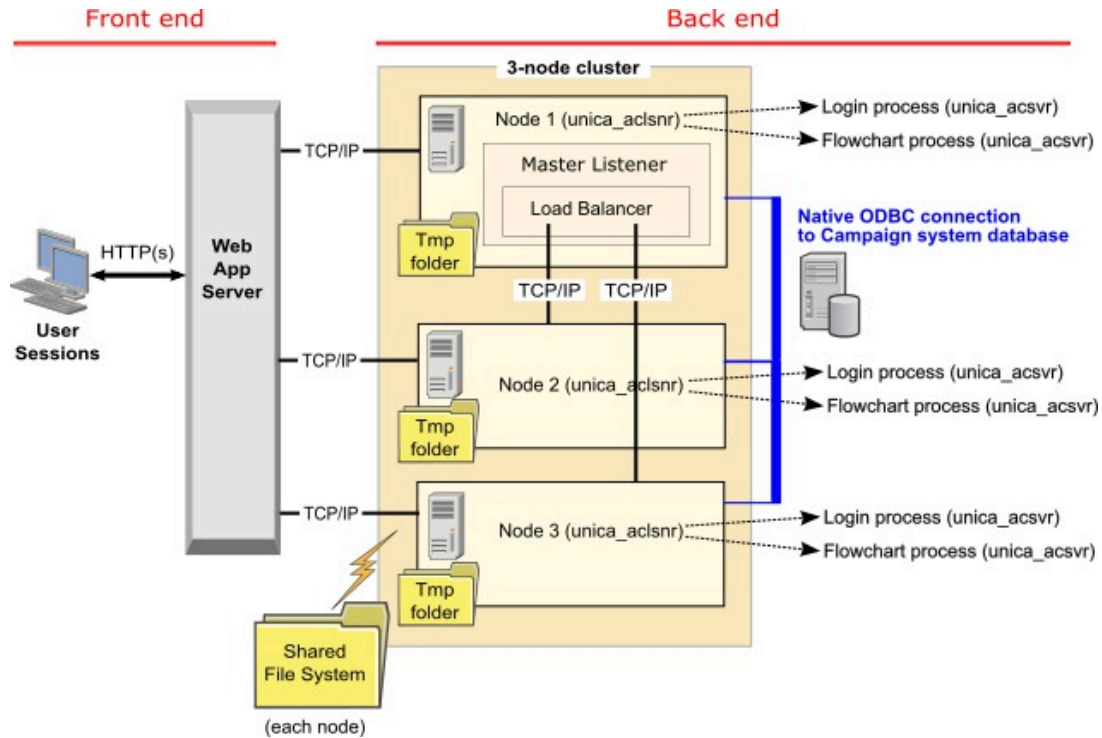


**Note:** The following summary provides a high-level overview of the components. Details are provided in separate topics.

A cluster consists of multiple listener nodes. Each node (`unica_aclsnr`) is on a separate physical machine, and each node has a native ODBC connection to the Unica Campaign system database. As with a single-node configuration, each `unica_aclsnr` process spawns additional back-end processes for logins and flowcharts.

Each node also has a connection to a back-end user database (not shown in diagram).

In a clustered configuration, one node acts as the master listener. The job of the master listener is to perform load balancing by distributing incoming requests to each node. The Unica Campaign web application sends client requests over TCP/IP, and the load balancer component communicates with clustered nodes over TCP/IP. All nodes share a network file system, so they can access shared files. Additionally, each node has its own local tmp folder and its own set of files that are not shared.



## Supported listener clustering configurations

This topic pertains to a clustered listener configuration.

Prerequisites and requirements for Unica Campaign listener cluster configurations:

- There can be only one listener per physical host machine.
- All back-end machines for clustered listeners must be running the same operating system type.
- All back-end machines for clustered listeners must have the same version of Unica Campaign installed.
- The shared network location (campaignSharedHome) must be in place and must be accessible from each physical host machine on which you plan to install a listener node. You must set this up prior to installing the listener nodes.
- listenerType –Campaign Listener when installed in cluster mode, ListenerType plays an important role to decide which listener should execute Campaign flowcharts or Optimize session or both.
  - ListenerType - 1 - Run only Campaign flowcharts
  - ListenerType - 2 - Run only Optimize sessions
  - ListenerType - 3 - Run Campaign flowchart and Optimize session

These listener types are set based on the option selected while installing Campaign listener. Users can switch between these values at any time and restart the application to reflect this.

## The master listener

A clustered listener configuration always includes a master listener. The master listener is a lightweight application whose job is to perform load balancing. It allocates requests to each of the running listeners in the cluster.

The master listener includes a load balancer component that coordinates load distribution across the cluster. The master listener and the load balancer function as a unit.

If the master listener goes down for any reason (hardware, software, or network failure), the Unica Campaign Web application detects the failure. The web application asks the next node to become the master listener. The requested listener performs the master listener election, and the highest priority node that is available becomes the master listener. Failover occurs automatically. Since the load balancer is a component of the master listener, the new master listener then handles load balancing.

There is always one master listener in a cluster. Any node in the cluster can perform as a master listener. The Unica Campaign configuration settings determine which node initially acts as the master listener (**masterListenerPriority**) and how load is balanced across clustered nodes (**loadBalanceWeight**).

If you have only a single listener, then load balancing and failover are not possible. A single listener performs all responsibilities without any additional listener nodes. However, if a failure occurs, and reconnection is possible, the listener is automatically restarted whenever possible. Upon restart, the listener recovers all of its back-end process connections.

For example, if a listener process is restarted, communication between the Web server and the listener are restored without user intervention. The Web server retries until the listener is available and then reconnects with the listener for each user session that was in progress.

## Master listener priority

A listener cluster always includes one master listener, whose job is to coordinate load balancing across the cluster. The **masterListenerPriority** configuration setting determines which node is initially used as the master listener.

Each node in a cluster has a **masterListenerPriority** value assigned in its configuration settings. A value of one (1) is the highest priority, so that node initially serves as the master listener. If the designated master listener cannot be contacted, the next node becomes the master listener, based on its **masterListenerPriority** value (for example, 2).

Every listener in the cluster must have a priority value. You cannot prohibit a listener from being designated as master. If you do not want a listener node to serve as the master listener, assign it the lowest priority, for example a 10.

For more information, see the topic that explains the configuration setting `Campaign|unicaACLlistener|node[n]|masterListenerPriority`.



**Note:** If you change the **masterListenerPriority**, you must run the `unica_svradm refresh` command to inform the listener cluster of the change.

## Load based routing load balancing

This topic pertains to a clustered listener configuration. To achieve load balancing, Unica Campaign uses a load based routing algorithm. This algorithm maintains a weighted list of servers, where a heavier (higher) weight indicates preference.

Each node in a cluster can process a portion of the total application traffic. The `loadBalanceWeight` configuration setting and current execution load on the listener determines how transactions are allocated to clustered nodes. Before allocating every new connection to the listener, it looks at the load of server processes running on the listener and listener's assigned weight. As a result, traffic is distributed fairly and efficiently to the listeners, that you rank as being more capable of handling requests.

The **loadBalanceWeight** assigns a relative value to each node. A higher value gives a node a greater proportion of the load, so more transactions are given to that listener. Lower values are assigned to less powerful or more heavily loaded machines, so fewer transactions are sent to those listeners. A value of 0 prohibits a listener from processing any transactions and typically is not used.

Starting Campaign 12.0, load balancing feature is enabled for Unica Optimize as well. When multiple optimize sessions are executed at a time, load of the session execution is fairly distributed to the listeners, based on current load and weight of the listener.

For details and an example, see the topic that explains the `Campaign|unicaACLlistener|node[n]|loadBalanceWeight` configuration setting.

If you change the `loadBalanceWeight`, run the `unica_svradmrefresh` command to inform the master listener of the change.

For details, see the topic that explains the `Campaign|unicaACLlistener|node[n]|listenerType` configuration settings.

### Listener Type

This topic pertains to a clustered listener configuration. From v12.0 onwards, there is a single listener which serves commands or execution related to both campaign flowchart and optimize sessions. Optimize session requires heavy hardware configuration for successful execution as compared to Campaign flowchart.

To help master listener to decide on which node it must send execution of flowchart or session considering `loadBalanceWeight`, this new field is introduced. Executing optimize session on the node and configuring to execute campaign flowcharts heavily slows down its execution. Vice-versa keeping a heavy hardware configured node in executing flowcharts is not desired. Based on this flag master, listener can utilize available resources in proper way. Users must choose appropriate `listenerType` during installation, depending on their hardware configuration or requirements.

## Listener failover

This topic pertains to a clustered listener configuration. As long as at least one Unica Campaign listener is viable, failover should occur without disruption.

Failover involves automatically switching to an alternate node in a cluster. Listener failover may occur for any of the following reasons:

- Network issues (TCP/IP)
- Listener (software) failure
- Hardware failure

Failover ensures that when a listener node is unresponsive for any reason, another node in the cluster takes over. Whenever possible, any flowchart sessions (`unica_acsvr`) that were spawned by the failed listener are also recovered, so flowchart work is not lost.

In rare cases, there may be a non-recoverable situation and any work that is in memory will be lost. In this case, a message alerts users to the situation so they can make a note of any flowchart changes that need to be redone.

To avoid possible loss to flowchart work, best practice is to configure **checkpointFrequency** and **autosaveFrequency** in the Unica Campaign partition settings. You can override the global configuration settings for individual flowcharts, using the **Admin > Advanced** option.

## Listener failover scenario 1: Non-master listener node fails

This topic pertains to a clustered listener configuration. In this scenario, a non-master listener node is unresponsive. Either the node is down or is not reachable due to network issues.

A node is determined to be unresponsive based on a limited number of retries within a certain time period.

In this case, the master listener concludes that the node is down. During the node down-time, the master listener stops routing requests to that node. Instead, requests are routed to any remaining listeners in the cluster, based on the assigned **masterListenerPriority** and **loadBalanceWeight**. If there are no other viable listeners, the single remaining listener processes all requests itself.

When the unresponsive node comes back up, requests are again routed to it. In this scenario, the interruption and subsequent recovery are recorded in the `masterlistener.log`. If the listener node comes back up before user takes any action, the user will not notice any disruption, because the connection has been restored. If the user takes action while the listener node is down, failover will occur and flowchart will be moved to another listener. In this case, the user is alerted with a message.

## Listener failover scenario 2: Master listener node fails

This topic pertains to a clustered listener configuration. In this scenario, the master listener node is unresponsive. Either the node is down or is not reachable due to network issues.

The node is determined to be unresponsive based on a limited number of retries within a certain time period.

In this case, the Unica Campaign web application asks the next node in the cluster to become the master listener, based on **masterListenerPriority**. The node becomes the master listener based on master listener election and takes over load balancing duties. The master listener also performs session synchronization between multiple listeners.



When the unresponsive node comes back up, it performs as a non-master listener. It does not automatically regain master listener status. If you want to make a different listener the master listener, you must stop the currently serving master listener first.

The cluster configuration changes are recorded in the `masterlistener.log`.



**Note:** If a user was editing a flowchart or other object, any unsaved data is lost. The cluster automatically re-establishes the connection to the same session file (.ses) for the flowchart in Edit mode. However, any data that was not saved (either manually or by configured **checkpointFrequency** and **autosaveFrequency**) is lost.

## Log files for clustered listeners

Log files for a clustered listener configuration are in the following locations.

<Campaign\_home>/logs

<Campaign\_home>/partitions/partition[n]/logs

<campaignSharedHome>/logs

<campaignSharedHome>/partitions/partition[n]/logs

<campaignSharedHome> is a shared location that was specified at installation time. It is configurable at `Campaign | campaignClustering | campaignSharedHome`.

<Campaign\_home> is an environment variable that represents the Unica Campaign application installation directory. This variable is set in `cmpServer.bat` (Windows™) or `rc.unica_ac.sh` (UNIX™).

## Shared network location for clustered listeners: campaignSharedHome

A clustered listener configuration for Unica Campaign requires that certain files and folders must be shared and accessible to all listeners in a cluster. Therefore, you must have a shared file system in place.

### Requirements

- The common area can be a machine or location that is accessible from all other machines in the listener cluster.
- Each listener in a cluster must have full access to the shared files and folders.
- Best practice is for all listeners to be on the same network, and for the shared home to also be on that network, to avoid latency issues.
- To avoid any single point of failure, the shared file system should use mirrored RAID or an equivalent method of redundancy.
- If you are installing a single-listener configuration, a shared file system is a best practice, in case you decide to implement listener clusters in the future.

## Shared files and folders

In a clustered configuration, all listeners share the folder structure shown below. The shared location (<campaignSharedHome>) is specified at installation time and is configurable at **Campaign|campaignClustering|campaignSharedHome**. The shared partition contains all logs, campaigns, templates, and other files.

```
campaignSharedHome
|--->/conf
|-----> activeSessions.udb
|-----> deadSessions.udb
|-----> etc.
|--->/logs
|-----> masterlistener.log
|-----> etc.
|--->/partitions
|-----> partition[n]
|-----> {similar to <Campaign_home> partition folder structure}
```

## Files and folders that are not shared

Each Unica Campaign listener has its set of folders and files that are not shared, under <Campaign\_home>. Campaign\_home is an environment variable that represents the Unica Campaign application installation directory. This variable is set in cmpServer.bat (Windows™) or rc.unica\_ac.sh (UNIX™). Partitions are specific to the local listener. Each local partition folder contains a **tmp** folder for temporary files during flowchart runs and a **conf** folder for the table manager cache file.

```
Campaign_home
|--->/conf
|-----> config.xml
|-----> unica_aclsnr.pid
|-----> unica_aclsnr.udb
|-----> etc.
|--->/logs
|-----> unica_aclsnr.log
|-----> etc.
|--->/partitions
|-----> partition[n]
|-----> /tmp
|-----> /conf
|-----> {other files specific to the partition}
```

## Utilities for clustered listeners


In general, you use the Unica Campaign utilities in a clustered listener environment the same way that you use them in a single-node environment. However, there are some differences to be aware of.

The following table summarizes the differences in using the utilities in a clustered listener environment.



**Note:** This table is just a summary. For details, see the appropriate topics on using the utilities.

**Table 45. Using Unica Campaign utilities with clustered listeners**

Utility	Notes® for clustered listener configurations
Unica Campaign Listener Shutdown utility (svrstop)	<p>Use the svrstop utility to gracefully shut down a listener node. For example, run the command prior to performing maintenance on a server.</p> <p>In a clustered environment, run the svrstop command with the -s (server host name) option to indicate which node to stop. It is not necessary to specify a port. If do not specify a host name, the utility will stop the listener on the current host.</p> <p> <b>Note:</b> The svrstop command does not stop an entire cluster. To shut down a cluster, use the Unica Campaign Server Manager Shutdown command.</p>
Unica Campaign Server Manager (unica_svradm)	<p>In a clustered listener environment, the default connection when you run unica_svradm is to the master listener. When you are connected to the master listener, you can issue the following commands to the master listener, and they are treated as commands to the entire cluster: Loglevel, Refresh, Shutdown, Status, Version.</p> <p>To affect only a single listener, use Connect -s to specify a node, then run the command.</p> <p>The unica_svradm command-line prompt indicates the server and port of the listener machine to which you are connected.</p> <p>For details on each command, see the appropriate topics on using the Unica Campaign Server Manager.</p>
Unica Campaign session utility (unica_acsesutil)	Run unica_acsesutil on each listener node, as needed. This utility operates on the .ses file.
Unica Campaign cleanup utility (unica_acclean)	Run unica_acclean on each listener node, as needed.
Unica Campaign report generation utility (unica_acgenrpt)	Run unica_acgenrpt on each listener node, as needed. This utility operates on the .ses file.

**Table 45. Using Unica Campaign utilities with clustered listeners (continued)**

Utility	Notes® for clustered listener configurations
Unica Campaign trigger utility (unica_actrg)	<p>In a clustered listener environment, all requests are automatically sent to the master listener, which broadcasts the trigger message to all clustered listener nodes. Example: <code>unica_actrg C003 web_hit</code></p> <p>You do not need to indicate the port or server name unless you run the command from a remote machine or a script.</p>

## Starting and stopping Unica Campaign listeners

If you install a listener as a Windows™ service or as part of the `init` process on UNIX™, the listener starts automatically when you start the server. You can also start and stop a listener manually.

### Installing the Unica Campaign listener as a Windows service

You can install the Unica Campaign listener as a Windows™ service so that it starts automatically whenever you start Windows™.

1. Add the `bin` directory under the Unica Campaign installation directory to the user `PATH` environment variable. If the `PATH` environment variable does not exist for the user, create it.

Make sure that you add this path to the user `PATH` variable, not the system `PATH` variable.

If the Unica Campaign `bin` directory exists in the system `PATH` environment variable, remove it. You do not need it in the system `PATH` environment variable to install the Unica Campaign listener as a service.

2. If you are upgrading from a previous version of Unica Campaign that had the server installed as a service, stop the service.
3. Open a command window and change directories to the `bin` directory under your Unica Campaign installation.
4. Run the following command to install the Unica Campaign listener as a Windows™ service:

```
unica_aclsnr -a
```



**Note:** The `-a` option includes auto-restart functionality. If you do not want the service to attempt to restart automatically, use `unica_aclsnr -i`.

The listener is now installed as a service.



**Note:** Make sure `CAMPAIGN_HOME` has been created as a system environment variable before starting the Unica Campaign listener service.



**Note:** You must add the appropriate listener type.

5. Open the **Unica Campaign Listener Service** properties dialog box. Click the **Log On** tab.
6. Select **This account**.
7. Enter the user name (system user) and password and start the services.

## Starting the Unica Campaign listener manually

Start the Unica Campaign listener by running the `cmpServer.bat` file for Windows™, or the `rc.unica_ac` command for UNIX™.

Follow the instructions appropriate for your operating system.

### Windows™

Start the Unica Campaign listener by running the `cmpServer.bat` file in the `bin` directory under your Unica Campaign installation. When the `unica_aclsnr.exe` process appears on the Windows™ Task Manager Processes tab, the server has started successfully.

### UNIX™

Start the Unica Campaign listener by executing the `rc.unica_ac` program with a `start` argument. You must run this command as root. For example:

```
./rc.unica_ac start
```

To determine whether the `unica_aclsnr` process has started successfully, enter this command:

```
ps -ef | grep unica_aclsnr
```

To determine the process ID for the server you started, view the `unica_aclsnr.pid` file located in the `conf` directory under your Unica Campaign installation.



**Note:** Since Unica Campaign supports listener clustering, a loop is added in the `rc.unica_ac` which automatically restarts the listener if in case the listener goes down or crashes. Also, process `rc.unica_ac` still remains after starting listener and the parent process may change. For example, if you are upgrading from Release 8.6.0.4 to Release 9.1.1, the parent process of `unica_aclsnr` can be changed to `rc.unica_ac` instead of `init`, which was the former parent process and the process `rc.unica_ac` still remains after starting the listener.

## Stopping the Unica Campaign listener

To stop the Unica Campaign listener, use the `svrstop -p 4664` command. On UNIX™ systems, you can also enter the following command at the system prompt: `rc.unica_ac stop`.

These instructions explain the basic steps for stopping a listener using the `svrstop` utility. The utility provides additional options which are explained in the Unica Campaign `svrstop` utility reference topic. Note that the `svrstop` command does not stop an entire cluster. To shut down a cluster, use the Campaign Server Manager (`unica_svradm`).

1. Go to the Unica Campaign `bin` directory and enter the following command: `svrstop -p 4664`

If you are prompted for the `CAMPAIGN_HOME` environment variable, set it as shown in the following example, then run the `svrstop` command again:

```
set CAMPAIGN_HOME=C:\<installation_path>\Unica Campaign
```

2. At the `Login` prompt, enter the Unica Campaign user name.
3. At the `Password` prompt, enter the password for the Unica Campaign user.

# Chapter 19. Unica Campaign utilities

Administrators use the Unica Campaign utilities to manage listeners, sessions, and flowcharts, and perform other important administrative tasks.

## Unica Campaign advanced search utility (advSrchUtil)

Unica Campaign v10 introduces the ability to search flowcharts. However, flowcharts are not searchable until you index them. Use `advSrchUtil` to index all flowcharts in the specified partition.

There are two ways to make flowcharts searchable:

- Batch index all flowcharts in a partition by running `advSrchUtil (.bat or .sh)`. This utility is a wrapper script that calls `unica_acsesutil`.
- Index a single flowchart by running `unica_acsesutil` with the `-J` option.

Existing indexes are not updated automatically. To ensure accurate search results, follow these guidelines:

1. Run `advSrchUtil` immediately after upgrading Unica Campaign, to index all existing flowcharts in a partition.
2. Run `advSrchUtil` periodically to create or update search indexes on all flowcharts in a partition.
3. If you want to affect just a single flowchart, run the `unica_acsesutil` session utility with the `-J` option.

When `advSrchUtil` runs for the first time, it processes all flowcharts in the given partition by serializing them into JSON and indexing them.

The next time `advSrchUtil` runs, it processes only the flowcharts that were modified or added since the last run.

For this reason, the first run takes longer than subsequent runs. A very large flowchart can take several minutes to process, so if there are many flowcharts the tool may take a long time to finish. Best practice is to run the tool regularly. For example, if you typically add or modify hundreds of flowcharts per week, consider running the tool twice a week.

1. Run the appropriate command for your operating system:
  - Unix/Linux™: `<Campaign_Home>/bin/advSrchUtil.sh <partition>`  
`<campaignSharedHome if clustered>`
  - Windows™: `<Campaign_Home>\bin\advSrchUtil.bat <partition>`  
`<campaignSharedHome if clustered>`

where:

`<partition>` is the name of the partition where the flowchart files exist.

`<campaignSharedHome if clustered>` is the shared network location for clustered listeners (required only for a clustered environment).

2. The utility interactively prompts for credentials. Enter the user name and password for the account that has permission to access all of the flowcharts in the given partition.

The utility looks for all flowchart files (`.ses`) in the specified partition that were created or modified since the last time the utility ran. It invokes `unica_acsesutil` for each flowchart session file (both campaign and session flowcharts). `Unica_acsesutil` then converts the `.ses` files to JSON and indexes them so they are searchable.

## Unica Campaign advanced search agent (advSrchAgent)

Use `advSrchAgent` to automatically index flowcharts and make them available for the search function whenever flowcharts are modified.

This search agent (`advSrchAgent.sh` on Aix/Linux and `advSrchAgent.bat` on Windows) is available in the `<CAMPAIGN_HOME>/bin` directory and must be kept running for every partition in Unica Campaign.



**Note:** `[cluster_home]` is a mandatory parameter if Unica Campaign is configured to use listener clustering. For a setup that uses listener cluster, this parameter (shared home directory for the listener cluster) must be provided accurately.

Run the appropriate command for your operating system:

- Unix/Linux: `advSrchAgent.sh`
  - Usage: `./advSrchAgent.sh <start | stop> <partition_name> [cluster_home] [-u <user_name>] [-p <password>]`
  - `<start | stop>` : Start or stop the agent
  - `<partition_name>` : Name of partition from which flowchart files are to be exported
  - `[cluster_home]` : Home directory of clustered environment, if clustering is enabled
  - `[-u <user_name>]` : User who has read access to all flowchart files partition specified
  - `[-p <password>]` : Password of the user specified
- Windows: `advSrchAgent.bat`
  - Usage: `advSrchAgent.bat <start | stop> <partition_name> [cluster_home] [-u <user_name>] [-p <password>]`
  - `<start | stop>` : Start or stop the agent
  - `<partition_name>` : Name of partition from which flowchart files are to be exported
  - `[cluster_home]` : Home directory of clustered environment, if clustering is enabled
  - `[-u <user_name>]` : User who has read access to all flowchart files partition specified
  - `[-p <password>]` : Password of the user specified



## Unica Campaign listener shutdown utility (svrstop)

Use the Unica Campaign listener shutdown utility (`svrstop`) to shut down the Unica Campaign listener or the Unica Optimize listener.

The listener shutdown utility can be used as a standalone command to stop the specified listener, or in a script if you also include the necessary authentication arguments.



**Important:** Best practice is to start up and shut down the Unica Optimize listener using the `ACOServer` script, which uses the `svrstop` utility. See the *Unica Optimize Installation Guide* for details.

### Unica Campaign svrstop utility reference

Use the `svrstop` utility to stop the Unica Campaign listener or the Unica Optimize listener, running on your local server or a server elsewhere on your network, for which you have the proper credentials.

The `svrstop` utility is installed automatically on every Unica Campaign server in the `<install_dir>/Campaign/bin` directory, where `<install_dir>` is the parent directory in which Unica Campaign is installed.

The `svrstop` utility uses the following syntax:

```
svrstop [-g] [-p <port> [-S]] [-s <serverName>] [-y <user>] [-z <password>] [-v] [-P <product>] [-f <force stop>]
```

Example:

```
svrstop -y asm_admin -z password -p 4664
```

Each argument is described in the following table:

**Table 46. svrstop syntax arguments**

Argument	Description
-g	Pings the specified server to determine whether the listener is active.
-p <port>	The port on which the listener is running. Set <i>&lt;port&gt;</i> to 4664 to shut down the Campaign listener. Set <i>&lt;port&gt;</i> to 2882 to shut down the Optimize listener.
-S	Specifies that the listener specified by the <code>-p</code> or <code>-P</code> argument is using SSL.
-s <serverName>	The host name of the server on which the listener is running, such as <code>optimizeServer</code> or <code>campaignServer.example.com</code> . If you omit this argument, the utility attempts to shut down the specified listener on your local server.
-y <user>	The Unica user with Unica Campaign administrator privileges to shut down the specified listener. If you omit this value, you will be prompted for a user when you run the utility.

**Table 46. svrstop syntax arguments (continued)**

Argument	Description
-z <password>	The password for the Unica user that you specified with the -y argument. If you omit this value, you will be prompted for a password when you run the utility.
-v	Reports the version information for the <code>svrstop</code> utility and exits without further actions.
-P <product>	The product whose listener you want to shut down. Set this to "Optimize" to shut down the Unica Optimize listener. Any other value for this argument, or omitting this argument, shuts down the Unica Campaign listener.
-f <force stop>	Use this option to force the server stop command. When you use this option, you are not prompted for any further information or confirmation. Without this option, you are prompted to confirm the server stop. (Do you really want to stop the listener? (Y/N))

## Using the svrstop utility to shut down the Unica Campaign listener

From a command prompt on the Unica Campaign server, you can run the `svrstop` utility to stop the Unica Campaign listener running on that server. To stop the Unica Campaign listener running on another server, use the `-s` argument, as in `-s servername.example.com`, and provide the required authentication.

Follow these steps to stop a Unica Campaign listener.



**Note:** The `svrstop` command does not stop an entire cluster. To shut down a cluster, use the Unica Campaign Server Manager (`unica_svradm`).

1. Open a command prompt on the Unica Campaign server.
2. Make sure that the `CAMPAIGN_HOME` environment variable is set to `<install_dir>/Campaign/bin`, where `<install_dir>` is the parent directory under which Unica Campaign is installed.
3. Enter the following command:

```
svrstop -p 4664
```

The `-p` argument specifies the port on which the listener is accepting connections. Port 4664 is the port that Unica Campaign uses internally to accept connections from the web client, so the `-p 4664` argument indicates you are stopping the Unica Campaign listener.

4. When prompted, provide the name and password of any Unica user with privileges to stop the listener.

Optionally, you could include `-y <username>` and `-z <password>` as arguments with the `svrstop` command to prevent the user name and password prompts from appearing.

## Using the svrstop utility to shut down the Unica Optimize listener

From a command prompt on the Unica Campaign server, you can run the `svrstop` utility to stop the Unica Optimize listener running on that server. To stop the Unica Optimize listener running on another server, use the `-s` argument, as in `-s servername.example.com`, and provide the required authentication.

1. Open a command prompt on the Unica Campaign server.
2. Make sure that the `CAMPAIGN_HOME` environment variable is set to `<install_dir>/Campaign/bin`, where `<install_dir>` is the parent directory under which Unica Campaign is installed.
3. Enter the following command:

```
svrstop -P "Optimize"
```

The `-P` argument specifies the product whose listener you want to shut down. Alternatively, you could enter `-P 2882` to shut down the listener using the internal port number 2882, which would also indicate the Unica Optimize listener.

4. When prompted, provide the name and password of any Unica user with privileges to stop the listener.

Optionally, you could include `-y <username>` and `-z <password>` as arguments with the `svrstop` command to prevent the user name and password prompts from appearing.

When you enter the required information, the Unica Optimize listener is shut down.

## Unica Campaign Server Manager (unica\_svradm)

The Unica Campaign Server Manager (`unica_svradm`) is a command-line server management utility.

Use `unica_svradm` to perform the following tasks:

- Connect to a Unica Campaign listener so you can run `unica_svradm` commands
- Disconnect from a listener
- View all open flowcharts and their states
- View and set environment variables
- View and set the listener logging level
- Change the owner of a campaign object.
- Run, suspend or resume, stop, or kill runaway flowcharts
- Gracefully shut down a listener or a listener cluster
- Refresh the configuration on the master listener (clustered listener configuration only)

When you start the `unica_svradm` utility, it checks whether a listener is running.

In a single-node configuration, it automatically connects to the running listener.

In a clustered configuration, it automatically connects to the master listener.

The command-line prompt indicates the server and port of the listener machine to which you are connected. For example: `unica_svradm[myhost01:4664]>`

## Running the Unica Campaign Server Manager (unica\_svradm)

Follow these instructions to run the `unica_svradm` command-line server management utility.

Before you can run `unica_svradm` utility:

- At least one listener must be running.
- The `UNICA_PLATFORM_HOME` and `CAMPAIGN_HOME` environment variables must be set for the command window that you are using.
- Your Unica login must have the **Run svradm command line tool** permission.

1. At a command prompt, enter:

```
unica_svradm -s listener_server -y Unica_Marketing_username -z Unica_Marketing_password
```

2. At the following prompt:

```
unica_svradm[server:port]>
```

issue the commands that are described in [Unica Campaign Server Manager commands \(unica\\_svradm\)](#) (on page 292).

## Unica Campaign Server Manager commands (unica\_svradm)

You can use any of the following commands with the Unica Campaign Server Manager (`unica_svradm`) utility. The commands are not case-sensitive but the parameters are. The command-line prompt indicates the server and port of the listener machine to which you are connected.



**Note:** When you run `unica_svradm` in a clustered listener environment, the default connection is to the master listener. When you are connected to the master listener, the following commands affect all of the nodes in the cluster: Loglevel, Refresh, Shutdown, Status, Version. Use the Connect command if you want to connect to a specific node.

## Cap (Unica Collaborate)

`Cap`

The `Cap` command prevents additional Unica Collaborate flowcharts from starting, while allowing flowcharts that are currently running to complete. Unset with the `uncap` command.

## Changeowner

```
Changeowner -o <olduserid> -n <newuserid> -p <policyid>
```

The `Changeowner` command allows you to change the owner of a user's campaigns objects. For example, use this command if you are deleting or disabling a user and you want to re-assign ownership of that user's campaigns objects to a new user. To run the command Change object/folder permission is required.

Option	Description
<code>-o &lt;olduserid&gt;</code>	User ID of the current owner of the campaign objects.
<code>-n &lt;newuserid&gt;</code>	User ID of the new owner that you will assign to the campaign objects.
<code>-p &lt;policyid&gt;</code>	Policy ID of the security policy to apply to the campaign objects.

## Connect

```
Connect [-f] [-s server] [-p port][-S]]
```

When you run `unica_svradm`, the command-line prompt indicates the server and port of the listener to which you are connected. Use the `connect` command if you want to connect to a different listener. You can connect to only one server at a time.

The following information pertains only to clustered listener environments:

- When you run `unica_svradm` in a clustered listener environment, the default connection is to the master listener.
- When you are connected to the master listener, the following commands affect all of the nodes in the cluster: Loglevel, Refresh, Shutdown, Status, Version. For example, the Status command displays the status of all nodes in the cluster.
- To affect only a single listener, use `Connect -s` to connect to a specific node, then run the desired command.
- If you are connected to the master listener and you run `Connect -s` against the master listener, you are reconnected in non-master listener mode. Any subsequent commands will affect only that node. To return to master listener mode, use the `disconnect` command.

Option	Description
<code>-s</code>	Identifies the server to connect to. In a single-node (unclustered) environment, <code>-s</code> must be followed by <code>-p</code> .
<code>-p</code>	In a single-node (unclustered) environment, <code>-s</code> and <code>-p</code> are required to identify the listener that you want to connect to.  In a clustered listener environment, <code>-p</code> is not necessary. Use <code>-s</code> to indicate the host, and the connection will be made based on the <code>serverPort</code> that is specified for <code>Campaign unicaACLis- -er node[n] </code> .
<code>-S</code>	If you use <code>-p</code> to specify a port, you may also include <code>-S</code> to establish an SSL connection.
<code>-f</code>	A common use for <code>-f</code> is when you are transitioning from a test environment to a production environment.

Option	Description
	<p>In a single-node (unclustered) environment: Use <code>-f</code> to force a connection to a listener that is not configured. The <code>-s</code> and <code>-p</code> options are required to identify the listener that you want to connect to.</p> <p>In a clustered listener environment: <code>-f</code> is never required to connect to a clustered listener node. However, you can use <code>-f</code> to force a connection to a listener that is not in the cluster. The <code>-s</code> and <code>-p</code> options are required.</p>

## Disconnect

Disconnect

The `Disconnect` command disconnects from the server. This command is available only if you are connected to a server.

In a single-node environment, you can use this command to disconnect, then use the `connect` command to connect to another server. Or you can run `connect` with the `-f` parameter, instead of disconnecting first.



**Note:** In a clustered listener environment, when you run `unica_svradm`, the default connection is to the master listener. If you disconnect from the master listener, `unica_svradm` is no longer connected to any listener. If you disconnect from a non-master listener, you automatically connect to the master listener. The command-line prompt indicates the server and port where you are connected. To force a connection to a different listener, use `connect` with the `-f` parameter.

## Exit

Exit

The `Exit` command logs you out of the Unica Campaign Server Manager.

## Help

Help

The `Help` command displays the available commands.

## Kill

`Kill -p pid [-h hostname]`

Use this command to end run-away processes associated with a listener. The `kill` command issues a "kill-p" to the specified process ID. The Windows NT™ equivalent is issued on Windows NT™. If you need to obtain the process ID (PID), use the `Status` command.

In a single-node (unclustered environment), you do not need to specify the hostname. Just run `kill -p pid`

In a clustered listener environment:

- The `kill` command only affects a single listener node (it never propagates to all nodes in the cluster).
- If you are connected to a non-master listener, you can omit the hostname. The command affects only that node.
- If you are connected to the master listener, you must specify the name of the server that is running the master listener. For example: `kill -p 1234 -h HostABC`

## Loglevel

`Loglevel [high | low | medium | all]`

To display the listener logging level, enter the `loglevel` command without any argument.

To set the listener logging level, enter the `loglevel` command followed by the desired logging level. `All` is the most detailed level and should be avoided except in troubleshooting situations.



**Note:** In a clustered environment, run the `loglevel` command when you are connected to the master listener to affect all clustered listener nodes. For example, `loglevel low` sets all listener nodes to the same logging level. If you are connected to a non-master listener, the command affects the current node only.

The change takes effect immediately, so there is no need to restart or refresh the listener after entering this command.

## Quit

`Quit`

The `Quit` command logs you out of the Unica Campaign Server Manager.

## Refresh

`Refresh`

The `Refresh` command is for use with a clustered listener configuration. If you have a single-node listener, the command has no effect.

The `Refresh` command informs the master listener of configuration changes and refreshes the configuration data on the master listener node. This avoids the need to reboot and provides a way to control when a refresh event occurs.

You must run the `Refresh` command in the following situations:

- After you adjust `Campaign|unicaACLlistener|node[n]|serverPort`.
- After you adjust `Campaign|unicaACLlistener|node[n]|masterListenerPriority`.
- After you adjust `Campaign|unicaACLlistener|node[n]|loadBalanceWeight`.

- After you add or remove a listener node at `Campaign|unicaACLlistener|node[n]`.



**Important:** Before you remove a listener node from the configuration, you must use the `svrstop` utility on each clustered listener node. In other words, you must stop all nodes, remove a node, then refresh. Otherwise, any existing sessions on the removed listener will continue to run but the master listener will not be able to contact the removed listener. This can cause unexpected results.

The `Refresh` command does not update the web application server. In most cases, it is sufficient to update the master listener only, but in certain cases you may also need to restart the web server.

## Resume

```
Resume {-s flowchart_name |-p pid |-a} [-h hostname]
```

The `Resume` command resumes running one or more suspended flowcharts.

- Use `-s` to resume a flowchart by name. All flowcharts with that name, in all campaigns and sessions, are affected. Therefore, it is best to use a relative flowchart path when you specify the flowchart name.
- Use `-p` to resume the specified process ID. (Use the `Status` command to obtain the PID.)
- Use `-a` to resume all suspended flowcharts.

In a single-node (unclustered) listener environment, you can omit the hostname.

In a clustered listener environment, if you are connected to the master listener, the listener host name is required. For example: `Resume -a -h Hostname`. If you are connected to a non-master listener, you can omit the hostname.

## Run

```
Run -p relative-path-from-partition-root -u MarketingPlatform_user_name [-h partition] [-c catalogFile] [-s] [-m]
```

The `Run` command opens and runs a flowchart file, as specified by the relative flowchart path and file name, partition, catalog file, and user name.

Use the following syntax:

```
[-S dataSource -U db_User -P db_Password]*
```





**Note:** On Unix platforms, flowcharts are run by the Unix account specified as the username's alternate login. On Windows NT™, flowcharts are run as the administrator's user login.

The `Run` command takes the following options.

Option	Description
-h	Specifies the partition name.



Option	Description
-l	<p>Indicates an alternate location to store the flowchart log file. Follow this option with a path that is relative to the Unica Campaign installation, as in <code>\partition1\logs</code>. Do not specify a file name because the file name is assigned automatically.</p> <p> <b>Note:</b> To use this option, <code>AllowCustomLogPath</code> must be enabled in <code>Campaign partitions partition [n] server logging</code>.</p>
-m	<p>Specifies that you are running multiple flowcharts. This option is not supported for batch flowcharts.</p>
-p	<p>Specifies the relative path from the partition root.</p>
-P	<p>Specifies the data source password.</p>
-s	<p>Specifies a synchronous run.</p>
-S	<p>Specifies the data source.</p>
-u	<p>Specifies the Unica user name.</p>
-U	<p>Specifies the data source user name.</p>
-v	<p>Specifies the user variable values for a flowchart directly in the command, using this syntax:</p> <pre data-bbox="402 1081 716 1104">[-v "varname=[ ' ]value[ ' ]"]*</pre>
-x	<p>Specifies the user variable values for a flowchart in an XML file, using this syntax:</p> <pre data-bbox="402 1213 591 1236">[-x xml-filename]</pre> <p><b>Example of an XML file for the -x argument:</b></p> <p>This sample XML file sets the user variable named <code>UVAacctType</code> to the value <code>Gold</code>.</p> <p> <b>Note:</b> Unica Campaign sets the user variable's value exactly as written in this file. Do not enclose the value in quotation marks if the value should not contain quotation marks.</p> <pre data-bbox="414 1535 1427 1877"> &lt;?xml version="1.0" encoding="UTF-8" standalone="yes" ?&gt;   &lt;UserVariables&gt;     &lt;UserVar Name="UVAacctType"&gt;       &lt;Values&gt;         &lt;Option&gt;Gold&lt;/Option&gt;       &lt;/Values&gt;     &lt;/UserVar&gt;   &lt;/UserVariables&gt; </pre>

## Save

```
Save {-s flowchart_name|-p pid|-a}-h hostname
```

The `save` command saves the current state of an active flowchart. In a clustered listener environment `-h hostname` is required, otherwise this can be ignored.

Option	Description
<code>-s</code>	Saves the flowchart that you identify by its <i>flowchart_name</i> . All flowcharts with that name, in all campaigns and sessions, will be saved. Therefore, it is best to use a relative flowchart path when you specify the flowchart name.
<code>-p</code>	Saves a flowchart defined by the process ID (pid). Use the <code>status</code> command to obtain the PID.
<code>-a</code>	Saves all running flowcharts.
<code>-h</code>	In a clustered listener environment, if you are connected to the master listener, the listener host name is required. For example: <code>save -a -h Hostname</code> . If you are connected to a non-master listener, you can omit the hostname

## Set

```
Set [variable[=value]]
```

The `set` command displays and sets environment variables. Omit the value to view the current value and provide it to set the specified variable.

## Shutdown

```
Shutdown [-f]
```

The `shutdown` command shuts down the listener.

The system checks for any running flowchart. If running flowcharts are found, a warning message asks for confirmation of shutdown.

To override and force shutdown, use `-f`.



**Note:** In a clustered listener environment, if you issue the `shutdown` command to a master listener, all clustered listener nodes are shut down. To shut down an individual node in a clustered configuration, connect to that listener and then run the `shutdown` command.

## Status

```
Status [-d |-i] [-u] [-v | -c]]
```

The `status` command provides information about active, suspended, and Unica Collaborate flowcharts. Information includes the flowchart owner (user name), process status, process ID, port, flowchart name, filename, and other

details. Use this command to identify disconnected or orphaned processes. Also use this command to obtain the process ID for commands that accept PID as an argument.



**Note:** In a clustered environment, run the `status` command when you are connected to the master listener to see the status of all clustered listener nodes. If you are connected to a non-master listener, the command displays the status for the current node only.

Option	Description
d	Adds Server ID, Campaign Code, and Campaign ID to the displayed output.
i	Displays the process ID (PID) only.
u	Use this option when the displayed data contains non-ASCII characters.
v	Verifies the existence of <code>unica_acsvr</code> processes before displaying output, which prevents crashed processes from being displayed in the Status list.
c	Verifies the existence of <code>unica_acsvr</code> processes before displaying output, which prevents crashed processes from being displayed in the Status list. Option <code>c</code> also instructs the listener to clean up any temporary files in the partition <code>temp</code> directory that are associated with crashed server processes.

The `Status` command identifies the processes as follows:

- `c` - connected (client is connected to the listener process -- may be running, may not)
- `d` - disconnected (client is closed but the flowchart is running in the background)
- `o` - orphaned (client is not connected to the flowchart and it is not running in the background. This is a lost process that cannot be reconnected to the listener and should be killed so people can log into it.)



**Note:** A value of `<no writer>` in the WRITER column refers to a server process that does not have a client in Edit mode, which may happen when no client is connected, and for login sessions.

## Stop

```
Stop [-f] {-s flowchart_name |-p pid |-a} [-h hostname]
```

The `Stop` command checks for active clients, warns if any are present (you can override this with the `-f` force option), and stops server processes for the specified flowchart.

In a single-node (unclustered) listener environment, you can omit the hostname.

In a clustered listener environment, if you are connected to the master listener, the listener host name is required. For example: `Stop -a -h Hostname`. If you are connected to a non-master listener, you can omit the hostname.

Option	Description
-s	Stops the flowchart that you identify by its <i>flowchart_name</i> . All flowcharts with that name, in all campaigns and sessions, will be affected. Therefore, it is best to use a relative flowchart path when you specify the flowchart name.
-p	Stops flowcharts by process ID (PID). Use the <code>Status</code> command to obtain the PID.
-a	Stops all running flowcharts.
-f	To override and force stop.

## Suspend

```
Suspend [-f] [-s flowchart_name | -p pid | -a] [-h hostname]
```

The `Suspend` command allows you to "quiesce" a running flowchart and save the state for later restart with the matching command, `Resume`. The system finishes running any currently running processes and does not allow any subsequent processes to start. Flowcharts that are currently running an output process complete their data export activities. Only then is the flowchart saved in a paused state and written to the list of suspended flowcharts. This causes the least amount of work to be lost and preserves data integrity of output files.

If flowcharts must be stopped immediately, issue a `Save` command followed by `Stop`.



**Note:** If a flowchart is not running at the time of `Suspend`, it is saved but not written to the listener and cannot be started with `Resume`.



**Note:** In a clustered listener environment, the `Suspend` command only affects a single listener node (it never propagates to all nodes in the cluster).

Option	Description
-s	Suspends the flowchart that you identify by its <i>flowchart_name</i> . All flowcharts with that name, in all campaigns and sessions, will be affected. Therefore, it is best to use a relative flowchart path when you specify the flowchart name.
-p	Suspends flowcharts specified by process ID (PID). Use the <code>Status</code> command to obtain the PID.
-a	Suspends all running flowcharts.
-f	The <code>-f</code> parameter allows you to force a suspend. Once suspended, the flowchart is written to the listener (the master listener, if you have a clustered configuration) as a suspended flowchart.
-h	-h indicates the name of the host that is running the listener.  In a single-node (unclustered environment), you do not have to specify the hostname.  In a clustered listener environment:

Option	Description
	<ul style="list-style-type: none"> <li>• If you are connected to a non-master listener, omit the hostname. The command affects only that node.</li> <li>• If you are connected to the master listener, the hostname is required (specify the name of the server that is running the master listener).</li> </ul>

## Uncap (Unica Collaborate)

`Uncap`

The `Uncap` command revokes the `Cap` (Unica Collaborate) command.

## Version

`Version`

This command displays the version of the listener process (`unica_aclsnr`) and Unica Campaign Server Manager (`unica_svradm`). Use this command to help troubleshoot a version mismatch error. For example, if you have multiple listener nodes acting as a cluster, each listener node must be running the same version of software.



**Note:** In a clustered environment, run the `version` command when you are connected to the master listener to display the version of all clustered listener nodes. If you are connected to a non-master listener, the command shows the version for the current node only.

Here is an example of output for a non-clustered configuration, or when you are connected to a non-master listener:

```
unica_svradm version: 9.1.1
unica_aclsnr version: 9.1.1
```

Here is an example of output when you are connected to the master listener:

```
unica_aclsnr version at <myhost01 : 4664> is: 9.1.1
unica_aclsnr version at <myhost02 : 4664> is: 9.1.1
unica_aclsnr version at <myhost03 : 4664> is: 9.1.1
unica_svradm version: 9.1.1
```

## Killing a running flowchart

Kill a flowchart if you need to stop it immediately. When you kill a flowchart, its buffer is not flushed to disk. Instead, the last checkpoint copy is saved.

Flowchart names can be the same in different campaigns and sessions. To ensure that you kill only the intended flowchart, be sure to follow the instructions in this topic.

1. At the command prompt, enter the following command to obtain a list of running flowcharts on the server:

```
% unica_svradm status
```

If multiple flowcharts have the same name, you can use the absolute path to uniquely identify a flowchart.

2. Note the PID associated with the flowchart that you want to kill.
3. To kill the flowchart, enter the following command at the command prompt, replacing *PID* with the PID of the flowchart that you want to kill:

```
unica_svradm kill -p PID
```

4. To kill the process in one line, run the following command.

```
unica_svradm.exe -y <User_Name> -z <password> -x "KILL -p <PID>"
```

## Unica Campaign session utility (unica\_acsesutil)

Use the Unica Campaign session utility (`unica_acsesutil`) to perform the following tasks.

- Import and export campaigns, sessions, and flowcharts from one server to another.
- Pass in a flowchart file or table catalog as input and generate a table catalog in either binary or XML format as output.
- Update record counts and lists of distinct values for a session or a catalog.
- Make the specified flowchart session file searchable by creating or updating its indexes.

Follow these guidelines when you run the utility:

- If you have clustered listeners, run the utility on each listener where you want to perform these tasks.
- The utility supports importing and exporting objects only between servers with the same version of Unica Campaign installed.
- If the campaign, session, or flowchart name contains characters from a locale that is not available or installed on your computer, you must install or set the locale in the terminal window before running the utility. For example: `export LANG=ja_JP.utf8`. If you have a file with more than one locale, set each locale in the terminal and then run the tool once.

If errors occur, the utility generates a log file on the listener server where it is run, at `<Campaign_home>/partitions/partition [n]/logs/unica_acsesutil.log`.

## Unica Campaign session utility syntax and options

Use the following syntax and options to run the Unica Campaign session utility. The `unica_acsesutil` utility operates on the `.ses` file.

```
unica_acsesutil -s sesFileName -h partitionName
[-r | -c | -x [-o outputFileName]] [-u]
```

```

[{-e exportFileName [-f {flowchart | campaign | session}]]
| {-i importFileName [-t ]
[-b {abort | replace | skip}]]}
[-p] [-a | -n | -l]
[-S dataSource -U DBUser -P DBPassword]*
[-y userName] [-z password]
[-j owner] [-K policy]
[-J

```

-J export session file into JSON and include in Flowchart Advanced Search.

The `unica_acsesutil` utility supports the following options.

```

unica_acsesutil -s sesFileName -h partitionName
[-r | -c | -x [-o outputFileName]] [-u]
[{-e exportFileName [-f {flowchart | campaign | session}]]
| {-i importFileName [-t ]
[-b {abort | replace | skip}]]}
[-p] [-a | -n | -l]
[-S dataSource -U DBUser -P DBPassword]*
[-y userName] [-z password]
[-j owner] [-K policy]
[-J]
[-Z]

```

In above command -Z is newly introduced

The `unica_acsesutil` utility supports the following options.


```
unica_acsesutil -A -y <userName> -z <password> -h <partitionName>
```

```
unica_acsesutil -U -y <userName> -z <password> -h <partitionName> -T <timeInSeconds>
```

**Table 47. Unica Campaign session utility (`unica_acsesutil`) options**


Option	Syntax	Description
-a	-a	Recomputes record counts and the list of distinct values for all tables.
-b	-b {abort   replace   skip}	Applies only for the import option (-i). Specifies that the import be done in batch mode.  Requires one of the following arguments to specify how to handle duplicate objects (if there is an ID conflict):

**Table 47. Unica Campaign session utility (unica\_acsesutil) options (continued)**

Option	Syntax	Description
		<ul style="list-style-type: none"> <li>• <code>abort</code> - If a duplicate object is detected, the import stops.</li> <li>• <code>replace</code> - If a duplicate object is detected, replace it with the imported object.</li> <li>• <code>skip</code> - If a duplicate object is detected, do not replace it and continue the import.</li> </ul>
<code>-c</code>	<code>-c &lt;outputFileName&gt;</code>	Generates a table catalog in <code>outputFileName</code> in <code>.cat</code> format (Unica Campaign internal format). This option is ignored with the <code>-s</code> option.
<code>-e</code>	<code>-e &lt;exportFileName&gt;</code>	Exports the object type specified by the <code>-f</code> option to a file named <code>exportFileName</code> .  If the <code>-f</code> option is not used, by default a flowchart is set for export.
<code>-f</code>	<code>-f {flowchart   campaign   session}</code>	Specifies the type of object to export. If this option is omitted, by default a flowchart is set for export.  If <code>-f</code> is used, requires one of the following arguments: <code>flowchart</code> , <code>campaign</code> , <code>session</code> .
<code>-h</code>	<code>-h &lt;partitionName&gt;</code>	Specifies the name of the partition in which the flowchart file (specified with <code>-s</code> ) is located. This parameter is required.
<code>-i</code>	<code>-i &lt;importFileName&gt;</code>	Specifies the name of the file being imported. This should be a file that was exported using the <code>-e</code> option in a previous export operation.
<code>-j</code>	<code>-j &lt;owner&gt;</code>	Specifies the owner of the file being imported or exported.
<code>-J</code>	<code>-s &lt;Flowchart ses file&gt; -h &lt;partition_name&gt; -J -y &lt;user_id&gt; -z &lt;password&gt;</code>	<p>Creates or updates search indexes for the specified flowchart <code>.ses</code> file. The <code>-J</code> option performs JSON serialization and indexes the flowchart so it is searchable. Use this option to index a single flowchart session file or refresh the index after making changes to the flowchart.</p> <p>You must use <code>-s</code> to indicate the session file. The options <code>-y</code> and <code>-z</code> are optional; if you omit them, the tool prompts for user ID and password.</p> <p> <b>Note:</b> Until flowcharts are indexed, they are not searchable. Flowcharts are not automatically indexed, and indexes are not</p>



**Table 47. Unica Campaign session utility (`unica_acsesutil`) options (continued)**

Option	Syntax	Description
		<p> automatically updated. You must index them initially (for example, after upgrading to Campaign v10), then periodically refresh the indexes to keep them up-to-date. To reindex all existing flowchart session files in a partition at one time, use the Campaign session utility (<code>advSrchUtil</code>).</p> <p>For example</p> <pre data-bbox="688 667 1425 806">unica_acsesutil -s &lt;Flowchart ses file&gt; -h &lt;partition_name&gt; -J -y &lt;user_id&gt; -z &lt;password&gt;</pre> <p>You must use <code>-s</code> to indicate the session file. The options <code>-y</code> and <code>-z</code> are optional; if you omit them, the tool prompts for user ID and password.</p>
-k	-k <policy>	Specifies the security policy of the file being imported.
-l	-l	Recomputes only the list of distinct values.
-n	-n	Recomputes only record counts.
-o	-o <outputFileName>	Specify the catalog with the name <code>outputFileName</code> . If unspecified, the default is <code>catFileName.xml</code> or <code>catFileName.cat</code> , depending on whether you use the <code>-x</code> or <code>-c</code> option. The output file name should specify a destination directory when using wildcards.
-P	-P <DBPassword>	Specifies the password for the database user account. Used with the <code>-U</code> and <code>-s</code> options.
-p	-p	Print table mappings to the console.
-r	-r <outputFileName>	Generate a flowchart XML report in <code>outputFileName</code> . This parameter is ignored when you use the <code>-t</code> option (using a table catalog as input).

**Table 47. Unica Campaign session utility (unica\_acsesutil) options (continued)**

Option	Syntax	Description
-S	-S <dataSource>	Specifies the name of the data source for the object being operated on. Use with the -U <database_user> and -P <database_password> options.
-s	-s <sesFileName>	<p>Specifies a Unica Campaign flowchart (.ses) file on which to operate. Specifying the .ses file is always required for exporting and importing, regardless of the object type (campaign, session, or flowchart). When exporting or importing a campaign or session with multiple associated flowcharts, any of the associated .ses files can be used.</p> <p>The file name should include the path below the partition in which this flowchart file is located (defined using the -h option). For example, a valid value for -s is:</p> <pre>"campaign/Campaign C00001_C00001_Flowchart 1.ses"</pre> <p>The &lt;sesFileName&gt; can contain wildcard characters to operate on multiple matching flowcharts.</p>
-t	-t <catFileName>	Read a table catalog named <catFileName> as the input. The <catFileName> can contain wildcard characters.
-U	-U <DBUserName>	Specifies the user login for the data source specified by the -s option. Use with the -P option, which specifies the database password for this database user.
-u	-u	Uses existing database authentication information when saving table catalogs.
-v	-v	Displays the version number and exits.
-x	-x <outputFileName>	Generates a table catalog file in an alternate XML format in outputFileName. If the input table catalog is a .cat file, it generates a corresponding .xml file, and vice versa.
-y	-y <userName>	Specifies the Unica user name.

**Table 47. Unica Campaign session utility (`unica_acsesutil`) options (continued)**

Option	Syntax	Description
-Z	-z <password>	Specifies the password for the user specified by the <code>-y</code> option.
-Z	-z	Adding new record in UA_FlowchartSeg system table, After processing .ses file. During campaign upgrade procedure.
-T	-T	Time in Seconds.
-A	-A	Updates UA_TableMappingXRef system table after processing all the flowchart's ses files.
-U	-U	Updates UA_TableMappingXRef system table after processing the ses files of flowcharts which was run in last <timeInSeconds> , We specify the time in seconds with -T option.

## Exporting and importing objects between servers

Use `unica_acsesutil` to export and import campaigns, sessions, and flowcharts from one server to another.

For all operating systems, set the following environment variables:

- `UNICA_PLATFORM_HOME`
- `CAMPAIGN_HOME`

On UNIX™ only, set the database-specific library path appropriate to your UNIX™ platform:

- `LIBPATH` for AIX®
- `SHLIB_PATH` for HP-UX
- `LD_LIBRARY_PATH` for Linux™ or Sun Solaris

The following information pertains to imports and exports:

- The source and target servers must have the same version of Unica Campaign installed.
- Regardless of whether you are exporting or importing a campaign, session, or flowchart, you must use `-s` to specify a `.ses` file. If a campaign or session includes multiple flowcharts, you can specify any of the associated `.ses` files.
- Before you can import a flowchart to a target system, the flowchart `.ses` file and its associated campaign or session must already exist on the target system. Therefore, you must: 1) manually copy the entire `Campaign|partitions|partition[n]` folder structure to the target system. You do not need to copy the `tmp` folder, and you may or may not want to copy the `logs` folder. Before you remove any files from the source system, be sure to do a complete backup of the entire folder structure (although you can omit the `tmp` folder). 2) Ensure that the `.ses` file for the flowchart exists on the target system (it should, if you copied the folder structure),

and 3) Use `unica_acsesutil` to import the associated campaign or session to the target system. After you complete those steps, you can use `unica_acsesutil` to import each flowchart.

- When you do an import, the `unica_acsesutil` imports data (such as session info, triggers, or custom macros) into the system tables. During the import, it checks whether each object already exists on the target system. The check is done based on internal object ID. If the internal campaign ID is not unique, `unica_acsesutil` asks if the campaign should be overwritten. If you choose to overwrite the campaign, `unica_acsesutil` removes all data associated with the existing campaign on the target server, and then imports the new campaign. Similarly, when importing offers, `unica_acsesutil` checks the internal Offer ID for uniqueness. If an object with the same ID already exists, you have the option to skip that object in the import process or to replace the existing object.



**Note:** If you know that an object (such as a campaign, session, or offer) already exists on the target system prior to import, consider deleting the object before you run the import, to avoid being prompted to resolve the conflict.

- If you plan to import Unica Deliver or Unica Collaborate flowcharts, that application must already be installed on the target system. If the application is not installed, `unica_acsesutil` generates an error and does not import the objects.

Moving objects between servers is done in several stages, with some manual steps required. A complete export and import is described below. You may choose to perform some subset of these steps.

1. To export a campaign or session: Use `-s` to specify any `.ses` file that is associated with the campaign or session, `-e` to specify an output file (`.exp`), and `-f` to indicate whether you want to export a campaign or a session.

Using the information in the flowchart `.ses` file that was specified with the `-s` option, the `unica_acsesutil` utility writes exported objects and information to an intermediate output file, specified by the `-e` option. Only the system tables and metadata are exported. If you want to export flowcharts, you must do so separately, one flowchart at a time, as explained below.

See the provided examples for details on command syntax.

2. To export a flowchart: Use `-s` to specify the `.ses` file, `-e` to specify an output file (`.exp`), and `-f` to indicate that you want to export a flowchart.

Repeat for each flowchart that you want to export, using a separate output file each time. For example: `Camp008_FC1.exp`, `Camp008_FC2.exp`, `Camp008_FC3.exp`.

See the provided examples for details on command syntax.

3. Determine whether the `Campaign|partitions|partition[n]` folder structure exists on the target system.

If it does not, you must manually copy the entire folder structure from the source to the target system. You do not need to copy the `tmp` folder, and you may or may not want to copy the `logs` folder.



**Note:** Before you remove any files from the source system, be sure to do a complete backup of the entire folder structure (although you can omit the `tmp` folder).

4. Manually copy the output file to the target server.

The output file is the `.exp` file that you specified using `-e` when you did the export. You may have multiple export files if you exported more than one campaign, session, or flowchart.

5. To import objects: On the target server, use `unica_acsesutil` with the `-i` option to import the output file.



**Important:** You must import the campaign or session before you import its flowcharts.

See the provided examples for details on command syntax.

If the operation is successful, the utility returns a value of 0. If no files are found with the specified flowchart or catalog file name, there is a return value of 1.

If errors occur, the utility generates a log file on the listener server where it is run, at `<Campaign_home>/partitions/partition [n]/logs/unica_acsesutil.log`.

## Example: Exporting a campaign or flowchart

The following examples show how to use the session utility (`unica_acsesutil`) to export a campaign or flowchart.

```
unica_acsesutil -s <sesFileName> -h <partitionName>
-e <exportFileName> [-f { flowchart | campaign | session }]
[-S <datasource> -U <DBusername> -P <DBpassword>]
```

### Example 1: Exporting a campaign

```
unica_acsesutil -s "campaigns/Campaign C000001_C000001.ses" -h partition1
-e campaign.exp -f campaign
```

Example 1 generates an output file named `campaign.exp` for exporting the campaign associated with Flowchart1, based on the `"campaigns/Campaign C000001_C000001.ses"` file located on `partition1`.

### Example 2: Exporting a flowchart

```
unica_acsesutil -s "campaigns/Campaign C000001_C000001_Flowchart1.ses"
-h partition1 -e flowchart.exp -f flowchart
```

Example 2 generates an output file named `flowchart.exp` for exporting flowchart `C000001_Flowchart1`, based on the `"campaigns/Campaign C000001_C000001_Flowchart1.ses"` file located on `partition1`.

## Example: Importing a campaign or flowchart

The following examples show how to use the session utility (`unica_acsesutil`) to import a campaign or flowchart.

```
unica_acsesutil -s <sesFileName> -h <partitionName>
-i <importFileName> [-f { flowchart | campaign | session }]
[-b { abort | replace | skip }]
[-S <datasource> -U <DBusername> -P <DBpassword>]
```

### Example 1: Importing a campaign

```
unica_acsesutil -s "campaigns/Campaign C000001_C000001.ses" -h partition1
-i campaign.exp -f campaign
```

Example 1 uses the previously generated `campaign.exp` file and imports Campaign C000001 data to the system tables on the target system, as well as to the `"campaigns/Campaign C000001_C000001.ses"` file located on partition1.

### Example 2: Importing a flowchart

Before you can import a flowchart to a target system, the flowchart `.ses` file and its associated campaign or session must already exist on the target system. Therefore, you must: 1) manually copy the entire `Campaign|partitions|partition[n]` folder structure to the target system. You do not need to copy the `tmp` folder, and you may or may not want to copy the `logs` folder. Before you remove any files from the source system, be sure to do a complete backup of the entire folder structure (although you can omit the `tmp` folder). 2) Ensure that the `.ses` file for the flowchart exists on the target system (it should, if you copied the folder structure), and 3) Use `unica_acsesutil` to import the associated campaign or session to the target system. After you complete those steps, you can use `unica_acsesutil` to import each flowchart.

```
unica_acsesutil -s "campaigns/Campaign C000001_C000001_
Flowchart1.ses" -h partition1 -i import.exp -f flowchart
```

Example 2 uses the previously generated `flowchart.exp` file and imports the data associated with Campaign C000001\_Flowchart1 to the system tables on the target system, as well as to the `"campaigns/Campaign C000001_C000001_Flowchart 1.ses"` file located on partition1.

## Backing up sessions

Use the Unica Campaign session utility (`unica_acsesutil`) to back up sessions.

You can write a script to export every file in the session directory and import the files to a backup system.

## Updating record counts and lists of distinct values

Use the Unica Campaign session utility (`unica_acsesutil`) to update record counts or lists of distinct values, as well as to schedule automatic re-computation of those counts.

Three parameters are available to indicate what type of counts to re-compute:

- `-n` -- recompute record counts only
- `-l` -- recompute list of distinct values only
- `-a` -- recompute record counts and list of distinct values for all tables

Use these options to recompute all record counts and/or list of values for a session (`-s`) or for a catalog (`-t`). You can combine these options with other options such as `import (-i)`.

### To recompute counts for all mapped tables in a flowchart

```
unica_acsesutil -s sesFileName -i importFileName
[{-a | -n | -l }][-S Datasource -U DBUser -P DBPassword]
```

### To recompute counts for tables in a table catalog

```
unica_acsesutil -h <partitionName> -t catFileName
[{-a | -n | -l }][-S Datasource -U DBUser -P DBPassword]
```



**Note:** You must specify the parameters that define the database connection (`-S`, `-U`, `-P`) if the connection information is not stored in the flowchart.

## Manipulating table catalogs

You can use the Unica Campaign session utility to manipulate table catalogs outside of Unica Campaign.

A common use of XML table catalogs is to perform global search and replace of data source names, for example to convert a table catalog developed for use against a test database, so that it works against a production database. In this case, you can export the table catalog as XML, perform global search and replace as needed, then save the XML table catalog and load it for use.

### Step 1 - Convert to XML format

You use the Unica Campaign session utility only for the first step of this process, which is to generate an XML format file that contains all the data from the requested catalog. If the catalog is already in an XML format, this step is not required.

Use the command:

```
unica_acsesutil -t catFileName -x [-o outputFileName] [-u] [-p]
[{-a | -n | -l }][-S dataSource -U DBUserName -P DBPassword]
```

### Step 2 - Edit as needed

You can now edit the XML file generated in step 1 as needed. To ensure that the file remains well formed, you should use an XML editor that checks the file syntax.

### Step 3 (optional) - Convert to binary format

If required, you can convert the XML catalog file back to a binary format catalog.

Use the command:

```
unica_acsesutil -t <catFileName> -x -o <outputFileName>
```



**Note:** Keeping catalogs in XML format risks exposing data access passwords. If you maintain catalogs in XML format, you should ensure that the file(s) are protected at the operating system level.

### Step 4 - Load the new catalog in a session

After conversion back into a binary format, you can now load the new catalog into a session.

## Documenting catalog contents

You can use `unica_acsesutil` to generate an XML format report, or to print table mappings.

### Using an XML catalog file

Use `unica_acsesutil` to generate an XML format file that contains all the data from the requested catalog.

Currently, there is not an utility to convert an XML catalog file into a user-friendly report.

### Printing table mappings

Use `unica_acsesutil` to print out the table mapping information from a catalog.

Use the command:

```
unica_acsesutil -t catFileName -h partitionName -p
```

### Updating UA\_TableMappingXRef system table

You can use `unica_acsesutil` to update the system table `UA_TableMappingXRef` for the table mappings that are 'in use' in the flowcharts.

#### Save All Table Mappings in Use:

Use the command:

```
unica_acsesutil -A -y <userName> -z <password> -h <partitionName>
```

This command will scan all the flowcharts ses files and saves the table mappings that are 'in use' in the database table `UA_TableMappingXRef`.

#### Example

```
unica_acsesutil -A -y asm_admin -z "" -h partition1
```



**Save Table Mappings Used in the Last T Seconds:**

```
unica_acsesutil -U -y <userName> -z <password> -h <partitionName> -T <timeInSeconds>
```

This command scans ses files of flowcharts that have been executed in the last <timeInSeconds>, saving the 'in use' table mappings in the `UA_TableMappingXRef` database table.

It is essential for flowcharts to execute at least once for processing. Updates to flowcharts will only be considered if the provided `-T <TimeInSeconds>` and the Last Run Start Date-Time fits within the specified time frame with respect to the current time.

**Example**

```
unica_acsesutil -U -y asm_admin -z password -h partition1 -T 3600
```

Suppose current time is 4 pm and the last run start date of the flowchart is 3pm.

So, in the command `-T` should be atleast "3600" seconds to identify the updated table mapping for the flowchart to include it in `UA_TableMappingXRef` table.

## Unica Campaign cleanup utility (unica\_acclean)

Use the cleanup utility (`unica_acclean`) to identify and clean up temporary files and database tables in the current partition. The cleanup utility can be used on the Unica Campaign system tables database and user table databases.



**Note:** When you run the `unica_acclean` utility, you must stop all flowcharts that are currently running or are scheduled to run.

To run this utility, users must have "Perform Cleanup Operations" permission, granted by their Unica Campaign administrator. If users attempt to run this utility without the appropriate privileges, the tool displays an error and then stops.



**Note:** This tool does not operate across partitions. Each time it is run, `unica_acclean` operates on tables and files only in the specified partition.

If you have clustered listeners, run the utility on each listener where you want to perform clean-up.

The utility can identify and clean up the following items:

- Temporary files and tables associated with a specified object or object type, based on given criteria.
- Orphaned temporary files and tables: temporary files and tables that were left behind after their associated object was deleted.

## Environment variables required for unica\_acclean

To run `unica_acclean`, the following environment variables must be set:

- UNICA\_PLATFORM\_HOME
- CAMPAIGN\_HOME
- LANG

Setting `CAMPAIGN_PARTITION_HOME` is optional.

## Unica Campaign cleanup utility syntax and options

The `unica_acclean` utility supports the following syntax and options.

```
unica_acclean {-d|-o <list file name>}
-w {flowchart | campaign | session | sessionfolder | campaignfolder | other}
-s <criteria>
[-u <user name>] [-p <password>] [-n <partition name>]
[-l {low|medium|high|all}]
[-f <log file name>][-D <datasource list>]
[-S <dataSource> -U <DB-user> -P <DB-password>]*
```


Run `unica_acclean` on each listener node, as needed.

The cleanup utility is non-interactive unless user name or password are not specified. If a user name is not specified, the utility prompts for user name and password. If a password is not specified, the utility prompts for a password.

**Table 48. Unica Campaign cleanup utility (`unica_acclean`) options**

Option	Syntax	Description
-d	-d	Deletes temp tables and files. All flowchart files are scanned. Based on the results, temporary files and tables are determined.
-f	-f <log file name>	Specifies the name of the file where errors are logged, located in the <PARTITION_HOME>/logs directory. By default, this file is named <code>unica_acclean.log</code> . You can modify the log file name but not the location.
-h	-h	Displays usage help. Any invalid command-line invocation also displays the help.
-i	-i <clean file name>	Specifies the file that lists the items to be deleted. Best practice is to use the same file generated by the cleanup tool, using the <code>-o</code> option.
-l	-l {low medium high all} [-f <logFileName>]	Specifies logging level and the log file name. If no level is specified, <code>medium</code> is used by default.
-n	-n <partition name>	Use this option to supply the name of the partition. If the partition name is not specified, the default of "partition 1" is used.

**Table 48. Unica Campaign cleanup utility (unica\_acclean) options (continued)**

Option	Syntax	Description
-o	-o <listfilename>	Outputs the list of tables and files to the specified file, but does not delete them.
-p	-p <password>	Must be used when the -u option is used. Use this option to supply the password for the user specified with the -u option.
-r	-r	<p>This option can be used only with the -w option for either campaignfolder or sessionfolder objects.</p> <p>When a folder is specified for cleanup and the -r option is added, unica_acclean performs the operation for all subdirectories of the specified folder. If only the -w option is used with a folder, unica_acclean performs the operation only on the top-level folder.</p>
-S	-S <dataSource>	<p>Use this option to override the credentials stored in Unica Platform or provide authentication for data sources whose ASMSaveDBAuthentication is set to FALSE.</p> <p>Specify the name of the data source along with the -U &lt;database_user&gt; and -P &lt;database_password&gt; options.</p> <p> <b>Note:</b> unica_acclean acts on all data sources irrespective of the data source specified in the -S option.</p>
-s	-s <criteria>	<p>Used with the -w option, defines criteria for cleanup, specified as a SQL query. The SQL LIKE operator can be used to do a search based on wildcards.</p> <p>Any data table column for the specified object can be used as criteria:</p> <ul style="list-style-type: none"> <li>• When a campaign folder or session folder is the object, criteria are based on columns in the UA_Folder table.</li> <li>• When a campaign is the object, criteria are based on columns in the UA_Campaign table.</li> <li>• When a flowchart is the object, criteria are based on columns in the UA_Flowchart table.</li> <li>• When a session is the object, criteria are based on columns in the UA_Session table.</li> </ul>

**Table 48. Unica Campaign cleanup utility (unica\_acclean) options (continued)**

Option	Syntax	Description
-U	-U <DBUserName>	Specifies the user login for the data source specified by the -s option. Use with the -P option, which specifies the database password for this database user.
-u	-u <user name>	Must be used when the -P option is used. Use this option to supply the Unica user name of the user running the utility.
-V	-v	Displays version and copyright information for the cleanup utility.
-W	-w {flowchart   campaign   session   sessionfolder   campaignfolder   orphan} -s <criteria> [-r]	<p>Searches for temp files and tables associated with the specified object type, based on the specified criteria, except when used with the orphan option.</p> <p>Only when used with orphan, searches for orphaned temp files and tables across the entire system.</p> <p>Requires -s &lt;criteria&gt; for all options except "orphan".</p> <p>Optionally use -r to recursively search subfolders.</p>
-D	[-D <datasource list>]	<p>Accepts single or multiple datasources separated by comma and enclosed in double quotes to search and delete orphan temp tables, orphan temp files, temp tables and temp files only from the specified datasource or a list of datasources.</p> <p>For example: <code>unica_acclean -o "log.txt" -w orphan -D "Test, UA_SYSTEM_TABLES"</code></p> <p>The above command will list all the orphan temp tables and files by scanning through only test and UA_SYSTEM_TABLES datasources.</p> <p><code>./unica_acclean -d -w Campaign -s "Name like '%Camp%'" -D Baseinfo</code></p> <p>The above command scans and deletes the temporary files and tables only from the Baseinfo datasource.</p>

## Unica Campaign cleanup utility use cases

Use the cleanup utility (unica\_acclean) to get information about orphaned files and tables and optionally delete all or some of them.

### Generating a list of orphaned files and tables

You can use the cleanup utility to identify and output a list of orphaned temp files and tables.



**Note:** recommends as a best practice that you output a list of identified orphaned files and tables for verification before performing deletions using the cleanup utility, rather than running the utility to immediately delete files and tables. This can help prevent accidental deletes; there is no recovery after deletion.

## To output a list of orphaned files and tables

```
unica_acclean -o <list file name> -w orphan
```

For this usage, `-w orphan` is required, and you cannot specify any criteria.

Use the `-o` option to specify the file name. You can also specify the path where you want the file to be saved. If you do not include the path, the file is saved in the same directory as the `unica_acclean` utility.

### Example

```
unica_acclean -o "OrphanList.txt" -w orphan
```

This example generates a list of orphaned files and tables and writes it to the file `OrphanList.txt`.

## Deleting the files and tables listed in a file

You can use the cleanup utility to delete all temp files and temp tables listed in a file generated by the utility.

## To delete the files and tables listed in a file

```
unica_acclean -d -i "OrphanList.txt"
```

where `OrphanList.txt` is the file containing the list of files to be deleted, generated by the cleanup utility.

If a line is read from the list file that is not a temp file or a temp table, the cleanup tool skips that item and logs an error to the console and the log file indicating that the item will not be deleted.

## Deleting all orphaned temp files and tables

You can use the cleanup utility to delete all temp files and tables that it identifies as being orphaned, from the system and user tables databases and the file system.

## To delete all orphaned temp files and tables from your system

```
unica_acclean -d -w orphan
```

## About orphaned files and tables

The `unica_acclean` utility uses the criteria described in this section to determine whether files and tables are orphaned.

## Tables

The utility scans databases in the current partition to obtain the list of temporary tables. Tables are identified as "temporary" based on the "TempTablePrefix" properties specified for each data source on the Unica Platform Configuration page.

Once the list of temporary tables is compiled, all flowchart files in the system are scanned to see if any of these temporary tables are used by a flowchart. Any temporary table not referenced by a flowchart is considered orphaned.



**Note:** The cleanup utility scans only those data sources defined in the Unica Platform User Administration module for the user running the utility. Therefore, users running the cleanup utility should always make sure that they have authentication rights to the global or appropriate set of data sources for scanning.

## Files

The utility scans two locations to identify temporary files:

- the partition's temp directory (*<partition home>/<partition>/tmp*) to obtain the list of files identified as "temporary" files based on the *.t~#* extension.
- the *<partition home>/<partition>/[campaigns | sessions]* directory for files having a known Unica Campaign temp file extension.

Once the list of temporary files is compiled, all flowchart files in the system are scanned to see if any of these temporary files are not used by any flowchart. Any temporary file not referenced by a flowchart is considered orphaned.

### Generating a list of files and tables selectively by object type and criteria

You can use the cleanup utility to generate a list of files and tables by object type and criteria.

### To generate a list of files and tables selectively by object type and criteria

```
unica_acclean -o <list file name> -w {flowchart | campaign | session | sessionfolder | campaignfolder} -s
criteria [-r]
```

#### Example 1: list temp files and tables by campaign folder

```
unica_acclean -o "JanuaryCampaignsList.txt" -w campaignfolder -s "NAME='JanuaryCampaigns'" -r
```

This example generates a list of temporary files and tables associated with campaigns and flowcharts in the campaign folder named "JanuaryCampaigns" as well as all subfolders of "JanuaryCampaigns", and writes it to the file *JanuaryCampaignsList.txt*.

#### Example 2: list temp files and tables by flowchart LASTRUNENDDATE

```
unica_acclean -o "LastRun_Dec312006_List.txt" -w flowchart -s "LASTRUNENDDATE < '31-Dec-06'"
```

This example generates a list of all temp files and tables with LASTRUNENDDATE earlier than December 31, 2006, in all flowcharts, and writes it to the file `LastRun_Dec312006_List.txt`.



**Note:** Make sure that any date criteria are specified in the correct date format for your database.

## Deleting files and tables selectively by object type and criteria

You can use the cleanup utility to delete temporary files and tables by object type and criteria.

## To delete files and tables selectively by object type and criteria

```
unica_acclean -d -w {flowchart | campaign | session | sessionfolder | campaignfolder} -s <criteria> [-r]
```

### Examples

#### Example 1: Delete temp files and tables by campaign folder

```
unica_acclean -d -w campaignfolder -s "NAME='JanuaryCampaigns'" -r
```

This example deletes temporary files and tables associated with campaigns and flowcharts in the campaign folder named "JanuaryCampaigns", as well as to all subfolders of "JanuaryCampaigns".

#### Example 2: delete temp files and tables by flowchart LASTRUNENDDATE

```
unica_acclean -d -w flowchart -s "LASTRUNENDDATE < '31-Dec-06'"
```

This example deletes all temp files and tables with LASTRUNENDDATE earlier than December 31, 2006, in all flowcharts.



**Important:** Make sure that any date criteria are specified in the correct date format for your database.

## Unica Campaign report generation utility (unica\_acgenrpt)

The `unica_acgenrpt` command-line report generation utility exports a flowchart cell report from a specified flowchart. The report is generated from the flowchart's `.ses` file.

Use the `unica_acgenrpt` utility to generate and export the following types of cell reports:

- Cell List
- Cell Variable Profile
- Cell Variable Crosstab
- Cell Content
- Cell Waterfall

For more information about these reports, see the Unica Campaign User's Guide.

The default file name of the exported file is unique and based on the flowchart name. It is saved in the directory you specify. If the file already exists, it is overwritten. The default file format is tab-delimited.



**Note:** The exported file contains the current data from the flowchart's `.ses` file. If a flowchart is writing to the `.ses` file when the `unica_acgenrpt` utility is run, the resulting report file could contain data from the previous run of the flowchart. If you are invoking the `unica_acgenrpt` utility using an on-success trigger, your script should contain an appropriate delay before executing `unica_acgenrpt` to account for the length of time that your flowcharts will need to finish writing to the `.ses` file. The amount of time required to save the `.ses` file varies greatly depending upon the size and complexity of the flowchart.

Use of the `unica_acgenrpt` utility requires the `Run genrpt Command Line Tool` permission in the Administrative Roles security policy. For more information on security policies and permissions, see [Security in Unica Campaign \(on page 21\)](#).

## Use case: capture cell counts from flowchart runs

To analyze cell counts over time, use the `unica_acgenrpt` utility to capture the cell counts from flowchart production runs. For the report type, specify `CellList`.

To automate this data capture, use an on-success trigger in your flowcharts to call a script that invokes the `unica_acgenrpt` utility. Use the `<FLOWCHARTFILENAME>` token to return the full path name of the flowchart's `.ses` file. To make the data available for analysis, use another script that loads the resulting export file into a table.

## Unica Campaign report generation utility syntax and options

The `unica_acgenrpt` utility supports the following syntax and options. Run `unica_acgenrpt` on each listener node, as needed. This utility operates on the `.ses` file.

The `unica_acgenrpt` utility has the following syntax:

```
unica_acgenrpt -s <sesFileName> -h <partitionName> -r <reportType> [-p <name>=<value>]* [-d <delimiter>] [-n]
[-i] [-o <outputFileName>] [-y <user>] [-z <password>] [-v]
```

**Table 49. Unica Campaign report generation utility (unica\_acgenrpt) options**

Option	Syntax	Description
-s	-s <sesFileName>	Specifies a Unica Campaign flowchart ( <code>.ses</code> ) file on which to operate. The file name should include the path below the partition in which this flowchart file is located (defined using the <code>-h</code> option). For example, a valid value for <code>-s</code> is:  "campaign/Campaign C00001_C00001_Flowchart 1.ses"



**Table 49. Unica Campaign report generation utility (unica\_acgenrpt) options (continued)**

Option	Syntax	Description
		The <code>&lt;sesFileName&gt;</code> can contain wildcard characters to operate on multiple matching flowcharts.
-h	-h <code>&lt;partitionName&gt;</code>	Specifies the name of the partition in which the flowchart file (specified with -s) is located.
-r	-r <code>&lt;reportType&gt;</code>	<p>Specifies the type of report to be generated. Valid values include:</p> <ul style="list-style-type: none"> <li>• CellList (Cell List report)</li> <li>• Profile (Cell Variable Profile report)</li> <li>• XTab (Cell Variable Crosstab report)</li> <li>• CellContent (Cell Content report)</li> </ul>
-p	-p <code>&lt;name&gt;=&lt;value&gt;</code>	Specifies report parameters using name=value pairs. The -p option can appear multiple times, and must appear after the -r option. For a list of valid name=value pairs supported by the -p option, see <a href="#">Parameters used with the unica_acgenrpt -p option (on page 322)</a> .
-d	-d <code>&lt;delimiter&gt;</code>	Separates columns in the output file. The default is TAB.
-n	-n	Includes column names before the report data in the output file.
-i	-i	Appends a unique text identifier to the end of the output file.
-o	-o <code>&lt;outputFileName&gt;</code>	Specifies the output file name. The default is <code>&lt;sesFileName&gt;</code> with <code>.ses</code> replaced by <code>.csv</code> ; specifies a destination directory when using wildcards.
-y	-y <code>&lt;user&gt;</code>	Specifies a login user name for Unica Campaign.
-Z	-z <code>&lt;password&gt;</code>	Specifies the password for the user login.

**Table 49. Unica Campaign report generation utility (unica\_acgenrpt) options (continued)**

Option	Syntax	Description
-V	-v	Displays the version number of the utility and exits.

## Parameters used with the unica\_acgenrpt -p option

The `unica_acgenrpt` utility's `-p` option allows you to specify parameters using name=value pairs for the Cell Variable Profile, Cell Variable Crosstab, and Cell Content reports.

### Cell Variable Profile report

Parameter name	Usage	Description
<code>cell</code>	Required	Name of a cell you want to profile.
<code>field</code>	Required	Name of the field you want to use to profile the cell.
<code>cell2</code>	Optional	Name of an additional cell to profile.
<code>bins</code>	Optional	Number of bins you want to include in the report. If the number you specify is less than the number of different field values, some fields will be joined in one bin. The default is 25.
<code>meta</code>	Optional	Specifies whether you want to profile by meta type. Valid values are <code>TRUE</code> and <code>FALSE</code> . The default is <code>TRUE</code> .

### Cell Variable Crosstab report

Parameter name	Usage	Description
<code>cell</code>	Required	Name of a cell you want to profile.
<code>field1</code>	Required	Name of the first field you want to use to profile the cell.
<code>field2</code>	Required	Name of the second field you want to use to profile the cell.
<code>cell2</code>	Optional	Name of an additional cell to profile.
<code>bins</code>	Optional	Number of bins you want to include in the report. If the number you specify is less than the number of different field values, some fields will be joined in one bin. The default is 10.

Parameter name	Usage	Description
<code>meta</code>	Optional	Specifies whether you want to profile by meta type. Valid values are <code>TRUE</code> and <code>FALSE</code> . The default is <code>TRUE</code> .

### Cell Content report

Parameter name	Usage	Description
<code>cell</code>	Required	Name of the cell you want to include in the report.
<code>field</code>	Optional	Name of a field you want to include in the report. Repeat multiple times to specify additional fields. If no field is specified, the report displays values for the audience field(s).
<code>records</code>	Optional	Number of records you want to include in the report. The default is <code>100</code> .
<code>skipdups</code>	Optional	Specifies whether you want to skip records with duplicate ID values. Enabling this option is useful if you are using non-normalized tables. Valid values are <code>TRUE</code> and <code>FALSE</code> . The default is <code>FALSE</code> .

### Cell Waterfall report

Parameter name	Usage	Description
<code>cell</code>	Required	Name of the cell you want to include in the report.
<code>path</code>	Required	Path in the flowchart that you want to analyze.

## Database testing utilities

Unica Campaign provides several command-line database testing utilities that you can use to test connections to a target database, execute queries, and perform a variety of tasks.

These utilities are in the `/Campaign/bin` directory on the Unica Campaign server.



**Note:** If your operating system does not provide the `db2test` utility, use the `cxntest` utility to test connections to the target database.

### Using the cxntest utility

Use `cxntest` to test connections to a target database and to issue commands when connected.

1. From a command prompt on the Unica Campaign server, run the `cxntest` utility.
2. Enter the following information when prompted:
  - a. The name of the connection library for your database.  
Library files are located in the same directory as the `cxntest` utility. For example: `libdb24d.so` (for DB2® on Linux™) or `db24d.dll` (for DB2® on Windows™).
  - b. The name of the data source.  
For example, the Oracle SID.
  - c. A database user ID.
  - d. The password that is associated with the database user ID.

The utility does not prompt for confirmation of your selections.

3. If the connection is successful, you can enter the following commands at the prompt:

- `bprint [ pattern ]`

Does an array fetch on a list of tables, 500 at a time. Optionally, specify a search pattern, where the pattern matches SQL standards, such as % for zero or more characters. For example, `bprint UA_%` finds all Unica Campaign tables that start with "UA\_".

- `describe table`

Describes the specified table. Returns each column name and its corresponding data type, storage length, precision, and scale.

- `exit`

Terminates the database connection and exits.

- `help`

Displays a list of supported commands.

- `print [pattern]`

Returns a list of tables. Optionally, specify a search pattern, where the pattern matches SQL standards, such as % for zero or more characters.

- `quit`

Terminates the database connection and exits.

- `SQL_command`

Executes any valid SQL command or series of SQL commands.

## Using the odbctest utility

The `odbctest` utility allows you to test Open DataBase Connectivity (ODBC) connections to a target database and, once connected, to issue a variety of commands.

This utility is supported on AIX®, Solaris, Windows™, and HP-UX systems (32-bit only). For Oracle and DB2® databases, use their native utilities.

1. From a command prompt on the Unica Campaign server, run the `odbctest` utility.

The utility returns a list of databases that you can connect to, similar to the following:

```
Registered Data Sources:
MS Access Database (Microsoft Access Driver (*.mdb))
dBASE Files (Microsoft dBase Driver (*.dbf))
Excel Files (Microsoft Excel Driver (*.xls))
```

2. Enter the following information exactly at the prompts:
  - a. The name of the database that you want to connect to (taken from the list of Registered Data Sources)
  - b. A database user ID
  - c. The password associated with the database user ID

The utility does not prompt for confirmation of your selections.

3. When you have successfully connected to the database, the utility prints out messages like the following and presents you with a command prompt:

```
Server ImpactDemo conforms to LEVEL 1.
Server's cursor commit behavior: CLOSE
Transactions supported: ALL
Maximum number of concurrent statements: 0
For a list of tables, use PRINT.
```

4. You can enter the following commands at the prompt:

- `bulk [number_of_records]`

Sets the number of records to return, as specified by `number_of_records`. The default is 1.

- `descres SQL_command`

Describes the columns returned by the SQL command specified by `SQL_command`.

- `describe pattern`

Describes the table or tables specified by `pattern`. Returns the corresponding type, data type, storage length, precision, and scale.

- `exit`

Terminates the database connection and exits.

- `help`

Displays a list of supported commands.

- `print[pattern]`

Returns a list of tables. Can optionally specify a search pattern.

- `quit`

Terminates the database connection and exits.

- `SQL_command`

Executes any valid SQL command or series of SQL commands.

- `typeinfo`

Returns a list of supported data types for the database.

## Using the `db2test` utility

The `db2test` utility allows you to test connections to a DB2® database and, once connected, to issue a variety of commands.

If your operating system does not provide the `db2test` utility, use the `cxntest` utility to test connections to the target database.

1. From a command prompt on the Unica Campaign server, run the `db2test` utility.

The utility returns a list of databases (registered data sources) that you can connect to.

2. Enter the following information exactly at the prompts:
  - The name of the database you want to connect to (taken from the list of Registered Data Sources)
  - A database user ID
  - The password associated with the database user ID

The utility does not prompt for confirmation of your selections.

3. When you have successfully connected to the database, the utility prints out messages like the following and presents you with a command prompt:

```
Server ImpactDemo conforms to LEVEL 1.
Server's cursor commit behavior: CLOSE
Transactions supported: ALL
Maximum number of concurrent statements: 0
For a list of tables, use PRINT.
```

4. You can enter the following commands at the prompt:

◦ `describe pattern`

Describes the table or tables specified by pattern. Returns the corresponding type, data type, storage length, precision, and scale.

◦ `exit`

Terminates the database connection and exits.

◦ `help`

Displays a list of supported commands.

◦ `print [pattern]`

Returns a list of tables. Can optionally specify a search pattern.

◦ `quit`

Terminates the database connection and exits.

◦ `SQL_command`

Executes any valid SQL command or series of SQL commands.

◦ `typeinfo`

Returns a list of supported data types for the database.

## Using the odbctest Utility to test connection with MariaDB

The following requirements and restrictions apply to using MariaDB with Unica Campaign.

- MariaDB version: 10.3.13 - 64 Bit
- MariaDB ODBC Connector



**Note:** V12.1 FP2 onwards we need to use MariaDB version: 10.4.x - 64 Bit

### Steps to Install MariaDB Connector 3.1.9 for SUSE

Download link : [https://downloads.mariadb.org/mariadb/repositories/#distro=SLES&distro\\_release=sles12-amd64--sles12&mirror=tuna&version=10.4](https://downloads.mariadb.org/mariadb/repositories/#distro=SLES&distro_release=sles12-amd64--sles12&mirror=tuna&version=10.4)

```
sudo rpm . --import https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
sudo zypper addrepo --pgpcheck --refresh https://yum.mariadb.org/10.4/sles/12/x86_64 mariadb
sudo zypper --pgp-auto-import-keys refresh
```

```
sudo zypper addrepo http://download.opensuse.org/distribution/11.4/repo/oss/ oss
sudo zypper install mariadb-connector-odbc
```

## Steps to Install MariaDB Connector 3.1.0 for Windows

### 1. Download links

- [ <https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.0/mariadb-connector-odbc-3.1.0-win64.msi> ]
- [ <https://mariadb.com/kb/en/library/installing-mariadb-msi-packages-on-windows/> ]



**Note:** From 12.1 FP 2 onwards, use 3.1.6 MariaDB ODBC Connector version for Windows and linux.

URL : <https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.6/>

## Steps to install MariaDB Connector/ODBC 3.1.0 for RHEL

1. Download the link : [ [https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.0/mariadb-connector-odbc-3.1.0-rc-rhel7-x86\\_64.tar.gz](https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.0/mariadb-connector-odbc-3.1.0-rc-rhel7-x86_64.tar.gz) ]
2. mkdir odbc\_package
3. cd odbc\_package
4. wget [ [https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.0/mariadb-connector-odbc-3.1.0-rc-rhel7-x86\\_64.tar.gz](https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.0/mariadb-connector-odbc-3.1.0-rc-rhel7-x86_64.tar.gz) ]
5. tar -xvzf mariadb-connector-odbc-3.1.0-ga-rhel7-x86\_64.tar.gz
6. sudo install lib64/libmaodbc.so /usr/lib64/



**Note:** V12.1 FP2 onwards we need to use MariaDB ODBC connector version 3.1.6 for Windows and Linux and 3.1.9 for Suse operating system.

Also, we need to add new MariaDB Driver level configurations as mentioned below.

### **MariaDB ODBC connector versions**

#### **Windows**

Maria DB ODBC Connector : 3.1.6

Download link : <https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.6/mariadb-connector-odbc-3.1.6-win64.msi>

#### **RHEL**

Maria DB ODBC Connector : 3.1.6



Download link : [https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.6/mariadb-connector-odbc-3.1.6-ga-rhel7-x86\\_64.tar.gz](https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.6/mariadb-connector-odbc-3.1.6-ga-rhel7-x86_64.tar.gz)

## SUSE

Maria DB ODBC Connector : 3.1.9

Download link : <https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.9/mariadb-connector-odbc-3.1.9-sles123-amd64.tar.gz>

## MariaDB ODBC driver level configurations

Example for windows - need to set in ODBC Driver Manager under statements parameter:

```
SET SQL_MODE='PIPES_AS_CONCAT,ANSI_QUOTES,IGNORE_SPACE,NO_KEY_OPTIONS';
```

```
Example for SUSE/linux - need to set in odbc.ini:
[<CampaignSystem>]
Driver=/usr/local/mariaDBConnector/lib/libmaodbc.so
DATABASE=<DATABASE_NAME>
DESCRIPTION=MariaDB
SERVER=<DATABASE_SERVER>
UID=<USER_NAME>
PASSWORD=<PASSWORD>
PORT=3306
InitStmnt=SET SQL_MODE='PIPES_AS_CONCAT,ANSI_QUOTES,IGNORE_SPACE,NO_KEY_OPTIONS';
```



**Note:** Use MariaDB database user name. DON'T use root and unica\*03 as username and password for UA\_SYSTEM\_TABLES datasource.

## MariaDB JAVA Client (JDBC Client): v2.4.0 -

Download the link:[ <https://mariadb.com/kb/en/library/about-mariadb-connector-j/> ]



**Note:** For AIX, we do not support Maria DB as System and User Database.

The odbctest utility allows you to test connections to an MariaDB server.

- From a command prompt on the Unica Campaign server, run the odbctest utility.
- Enter the following information exactly at the prompts:
  - The name of the MariaDB datasource name that you want to connect to.
  - A database user ID.
  - The password associated with the database user ID.

The utility does not prompt for confirmation of your selections. If successful, the utility prints a **Connection Successful** message and exits with a return value of zero (0).

## Using the oratest utility

The `oratest` utility allows you to test connections to an Oracle server.

1. From a command prompt on the Unica Campaign server, run the `oratest` utility.
2. Enter the following information exactly at the prompts:
  - a. The name of the Oracle server that you want to connect to
  - b. A database user ID
  - c. The password associated with the database user ID

The utility does not prompt for confirmation of your selections.

If successful, the utility prints a **Connection Successful** message and exits with a return value of zero (0).

# Chapter 20. Configuring the ACOOptAdmin tool

Configure the ACOOptAdmin tool to run Unica Optimize sessions from the command line.

Complete the following steps to configure the ACOOptAdmin tool:

1. Open the `ACOOptAdmin.sh` (UNIX™) or `ACOOptAdmin.bat` (Windows™) file.  
The file is in the `bin` directory under the Campaign/tools directory.
2. Edit **JAVA\_HOME** to replace `[Change Me]` with the path to the Java™ directory used by your web application server.
3. Save and close the file.
4. Add or edit the **JAVA\_OPTIONS** variable in your web application server startup script. Add the following options to configure the ACOOptAdmin tool in the SSL environment for Windows™ and UNIX™:
  - `-Djavax.net.ssl.keyStoreType=keyStoreTypeValue`
  - `-Djavax.net.ssl.keyStore=<keyStoreValue>`
  - `-Djavax.net.ssl.keyStorePassword=<keyStorePasswordValue>`
  - `-Djavax.net.ssl.trustStore=<trustStoreValue>`
  - `-Djavax.net.ssl.trustStorePassword=<trustStorePasswordValue>`

# Chapter 21. Non-ASCII data in Unica Campaign

Unica Campaign supports the use of localized data and non-United States locales, including multiple user-preferred locales within the same installation of the application.

To ensure that your application is set up to work correctly with non-ASCII data, non-United States locales, or user-specified locales, you must perform some specific configuration tasks. `unicacampaign` strongly recommends that you avoid using your application before you have completely configured your system and tested it for your data and locale. It is recommended that you perform these configuration steps on a new installation of the application.

## About using non-ASCII data or non-US locales

Before you perform any of the configuration procedures, you should understand the basic concepts as they apply to data and locale configuration in your Unica application.

### About character encodings

To configure your application to work with non-ASCII languages, you need to understand the character encodings used to store textual data in both your files and your databases.

A character encoding is the means by which human language is represented on a computer. Many different encodings are used to represent different languages. Special cases in character encoding occur in some text formats.

For more details, see [The encoding of text in character-based fields \(on page 333\)](#).

The supported encodings are listed in [Character encodings in Unica Campaign \(on page 544\)](#).

### About interactions with non-ASCII databases

You must understand the encoding and date format that your database server and client use, and ensure that you configure Unica Campaign correctly for these settings.

When an application communicates with a database, several language-sensitive areas must be understood between them, including:

- The format of date and time fields
- The encoding of text in character-based fields
- The sorting order expected in `ORDER BY` clauses of SQL `SELECT` statements

Unica Campaign communicates directly with the database client and the client communicates with the database. Each database varies in how it handles language-sensitive data.

### The format of date and time fields

This section describes considerations relating to date and time formats.

The format of a date field can have various characteristics, including:

- Order of the day, month, and year
- Delimiters between day, month, and year
- Representation of fully-written-out dates
- Type of calendar (Gregorian or Julian)
- Abbreviated and full weekday names
- Abbreviated and full month names

The format of a time field can have various characteristics, including:

- The hour format (for example, 12-hour or 24-hour format)
- The representation of minutes and seconds
- The locale-specific indicator for AM/PM



**Important:** If you use the multi-locale feature, you should not use date formats containing 3-letter months (MMM), %b (abbreviated month name), or %B (full month name). Instead, you should use a delimited or fixed format with a numeric value for the month. For more details about date formats, see [Date and time formats \(on page 549\)](#). For more details about the multi-locale feature, see [About the multi-locale feature \(on page 333\)](#).

Date and time formats can appear in SQL statements as well as in the data returned by the database (called a resultset). Some database clients support or require different formats between SQL statements (output) and resultsets (input). The Configuration page for Unica Campaign includes a parameter (`DateFormat`, `DateOutputFormatString`, `DateTimeFormat`, and `DateTimeOutputFormatString`) for each of the different formats.

## The encoding of text in character-based fields

Data in `CHAR`, `VARCHAR`, and other text-based fields have a specific character encoding. When databases are created, they can specify the encoding that will be used throughout the database.

Unica Campaign can be configured to use one of many different character encodings on a database-wide scale. Per-column encodings are not supported.

In many databases, the database client transcodes between the encoding of the database and the encoding that an application uses. This is common when an application uses some form of Unicode but the database uses a language-specific encoding.

## About the multi-locale feature

Unica Campaign supports multiple languages and locales for a single installation. Unica Campaign has a default language and locale which is set during installation, but you can optionally set an individual locale preference for each user in Unica.

Setting a user's locale preference is optional. Unless a preferred locale is set explicitly in Unica for a user, there is no "preferred" locale at the user level, and when that user logs in, Unica Campaign uses the suite-level locale set in Unica.

When a preferred locale is explicitly set for a user, this preference overrides the suite-level setting, and when this user logs in to Unica Campaign, the user interface appears in the user's preferred language and locale. This setting applies until the session ends (that is, when the user logs out). Thus, the multi-locale feature allows multiple users to log in to Unica Campaign and work simultaneously, each in their preferred language and locale. For details about setting user locale preferences in Unica, see the Unica Platform Administrator's Guide.

To configure your system for multi-locale functionality, see [Configuring Unica Campaign for multiple locales \(on page 341\)](#). You complete the tasks in that section after configuring Unica Campaign for a non-ASCII language or non-US locale.



**Important:** If you use the multi-locale feature, you should not use date formats containing 3-letter months (MMM), %b (abbreviated month name), or %B (full month name). Instead, use a delimited or fixed format with a numeric value for the month.

## Areas not affected by user locale preferences

The user locale preference does not control all areas of display in Unica Campaign. The following areas are not affected by user locale preferences:

- Parts of the Unica Campaign interface that do not have a user context (for example, the log in page that appears prior to any user logging in). These parts of the interface are displayed in the default language.
- Any user-defined items within the user interface, if they are read from a user database (for example, custom or external attributes) are displayed only in their original database language.
- Data input – regardless of the locale setting, if your system tables are correctly set up with Unicode encoding, you can input data to Unica Campaign in any language.
- Unica Campaign command line tools – these display in the default language. You can override the Unica Campaign default language with the language specified in your system's `LANG` environment variable. If you change the `LANG` environment variable, the following Unica Campaign programs require a new invocation for the change to take effect:
  - `install_license`
  - `svrstop`
  - `unica_acclean.exe`
  - `unica_acgenrpt.exe`
  - `unica_aclsnr`
  - `unica_acsesutil`
  - `unica_actrg`
  - `unica_svradm`



**Note:** In Windows™, the language and regional settings must match. Regional settings affect all non-Unicode programs in Windows™, and must be explicitly set.

## Limitations of the multi-locale feature

The multi-locale feature has the several limitations, as described in this section.

- It is not supported for Japanese operating systems. To install Unica Campaign with a single locale on a Japanese OS, contact Technical Support.



**Note:** The multi-locale feature installed on non-Japanese operating system environments does not correctly support `ja` as a user locale preference.

- It is not supported by all applications. See the documentation for each application for information about multi-locale support.
- In multi-locale installations of Unica Campaign, when file names are in mixed languages, or when the command shell language (encoding) does not match the file name encoding, file names might appear garbled in command-line displays.
- Multi-locale installations of Unica Campaign on Windows™ platforms is supported only on NTFS drives, as FAT32 does not support Unicode character sets.
- Cell Profile Reports are not localized and remain in English regardless of locale.

## Configuring Unica Campaign for a non-ASCII language or a non-US locale

Several steps need to be performed in a specific order to correctly configure Unica Campaign.

Before you begin, read all topics in [About using non-ASCII data or non-US locales \(on page 332\)](#).

To configure Unica Campaign for localized data or non-ASCII locales, complete the tasks in the following list. Each step is described in detail later in this section.



**Important:** Do not skip any of the steps. Skipping a step could result in incorrect or incomplete configuration, causing errors or corrupted data.

1. [Set the operating system language and regional settings \(on page 335\)](#).
2. [Set encoding parameters for the web application server \(WebSphere only\) \(on page 336\)](#).
3. [Set Unica Campaign language and locale property values \(on page 336\)](#).
4. [Un-map and re-map system tables \(on page 338\)](#).
5. [Test your database and server configurations \(on page 338\)](#).

## Set the operating system language and regional settings

This is one of the required steps when configuring Unica Campaign to use a non-ASCII language or a non-US locale.

Configure the operating system language and regional settings on the Unica Campaign server and the system where the Unica Campaign web application is deployed. Your database might also require that you set the operating system

language and locale on the machine where the database is installed. See the database documentation to determine whether this is required.

### About setting the language and locale on UNIX™

UNIX™ systems require that appropriate languages are installed. To determine if the desired language is supported on an AIX®, HP, or Solaris machine, use this command:

```
# locale -a
```

The command returns all supported locales on the system. Note that Unica Campaign does not require that you install support for X Fonts and translations.

If the language that you need is not yet installed, use the information from the following sources to configure supported UNIX™ variants to work with a given language:

- [Solaris 9 International Language Environments Guide \(http://docs.sun.com/app/docs/doc/806-6642\)](http://docs.sun.com/app/docs/doc/806-6642)
- [AIX® 5.3 National Language Support Guide and Reference \(http://www-01.ibm.com/support/knowledgecenter/ssw\\_aix\\_53/com.ibm.aix.nls/doc/nlsgdrf/nlsgdrf.htm\)](http://www-01.ibm.com/support/knowledgecenter/ssw_aix_53/com.ibm.aix.nls/doc/nlsgdrf/nlsgdrf.htm)

### Setting the language and locale on Windows™

If the regional and language options on a Windows™ system are not yet configured for the language you need, do so now. If you need information about Windows™ language settings, see the resources available at <http://www.microsoft.com>.

You might need your system installation CD to complete this task.



**Note:** Be sure to restart the Windows™ system after you change the language settings.

The next step is to set encoding parameters for the web application server.

## Set encoding parameters for the web application server (WebSphere® only)

This is one of the required steps when configuring Unica Campaign to use a non-ASCII language or a non-US locale.

For WebSphere® only, if you are using Unica Campaign with a non-ASCII encoding, you must set

`-Dclient.encoding.override=UTF-8` as a JVM argument to ensure that the application server uses UCS Transformation Format for encoding.

For detailed instructions, see the IBM® WebSphere® documentation.

The next step is to set Unica Campaign language and locale property values.

## Set Unica Campaign language and locale property values

This is one of the required steps when configuring Unica Campaign to use a non-ASCII language or a non-US locale.



Unica Campaign supports multiple languages and locales for a single installation. Setting the Unica Campaign language and locale property values is a required step when configuring Unica Campaign to use a non-ASCII language or a non-US locale.

Use the Unica Campaign configuration settings in Unica Platform to set the values of configuration properties that control how Unica Campaign performs the following tasks:

- reads and writes data in text files and log files
- reads and writes date, time, and text fields within the database
- handles text it receives from the database.

The configuration settings determine the language and locale that is used for translated Unica Campaign messages (for example, text in the Unica Campaign user interface) and the format of dates, numbers, and currency symbols on the application's web pages. The display language is also used to initialize the flowchart editor, which is vital for enabling flowcharts to display non-ASCII text.



**Note:** Unica Campaign supports non-ASCII column names, table names, and database names. However, Unica Campaign supports NCHAR, NVARCHAR, etc. columns on SQL Server databases only. DB2® treats NCHAR and NVARCHAR format columns as if they were regular text fields. Oracle treats them as numeric fields.

1. Choose **Settings > Configuration**.
2. Adjust the following properties, and record the values for future reference.

- Campaign > currencyLocale
- Campaign > supportedLocales
- Campaign > defaultLocale
- Campaign > partitions > partition[n] > dataSources > [data\_source\_name] > DateFormat
- Campaign > partitions > partition[n] > dataSources > [data\_source\_name] > DateOutputFormatString
- Campaign > partitions > partition[n] > dataSources > [data\_source\_name] > DateTimeFormat
- Campaign > partitions > partition[n] > dataSources > [data\_source\_name] > DateTimeOutputFormatString
- Campaign > partitions > partition[n] > dataSources > [data\_source\_name] > EnableSelectOrderBy
- Campaign > partitions > partition[n] > dataSources > [data\_source\_name] > ODBCUnicode
- Campaign > partitions > partition[n] > dataSources > [data\_source\_name] > StringEncoding
- Campaign > partitions > partition[n] > dataSources > [data\_source\_name] > SuffixOnCreateDateField
- Campaign > partitions > partition[n] > server > encoding > stringEncoding
- Campaign > partitions > partition[n] > server > encoding > forceDCTOneBytePerChar
- Campaign > unicaACLlistener > logStringEncoding
- Campaign > unicaACLlistener > systemStringEncoding

The next step is to unmap and remap system tables.

## Un-map and re-map system tables

This is one of the required steps when configuring Unica Campaign to use a non-ASCII language or a non-US locale.

If any language-sensitive parameter is not set correctly, it can be difficult to configure system tables when you map them in the Administration area of Unica Campaign. The best practice is to set all parameters, then unmap ALL tables in the data source, log out, log in again, and then map all the tables again. Unica Campaign retains its existing settings for a data source until the data source is no longer used (that is, until it is unmapped).

The next step is to test your database and server configurations.

## Test your database and server configurations

Before you start creating campaigns or other objects, you must ensure that your database and server settings are correctly configured. This is also a required step when configuring Unica Campaign to use a non-ASCII language or a non-US locale.

Perform the following tests to ensure that your configurations are correct:

- [Test the database configuration \(on page 338\)](#)
- [Test that your attribute table is correctly configured \(on page 338\)](#)
- [Test campaigns and flowcharts that have ASCII and non-ASCII characters \(on page 339\)](#)
- [Test flowchart input and output that contains ASCII and non-ASCII characters \(on page 339\)](#)
- [Test that the correct language directory is used \(on page 340\)](#)
- [Test the date formats in calendar reports \(on page 341\)](#)
- [Test that the currency symbol displays correctly for your locale \(on page 341\)](#)

### Test the database configuration

1. Select **Settings > Campaign Settings**.

The Campaign Settings page appears.

2. Select **View Datasource Access**.
3. In the **Database Sources** dialog, select your data source name.

The data source details are displayed, including the database type and its configuration settings.

4. Scroll down to the **StringEncoding** property, and verify that the value is the same as the value you set for `dataSources > StringEncoding` on the Unica Platform Configuration page.
5. If the encoding is not as expected, remap your database tables and perform this test again.

### Test that your attribute table is correctly configured

1. Select **Settings > Campaign Settings**.

The Unica Campaign Settings page appears.

2. Select **Manage Table Mappings**

3. In the **Table Mappings** dialog, in the list of Unica Campaign system tables, select the Attribute Definition Table (UA\_AttributeDef) and click **Browse**.
4. In the **Attribute Definition Table** window, verify that non-ASCII characters display correctly.

## Test campaigns and flowcharts that have ASCII and non-ASCII characters

1. In Unica Campaign, create a campaign using the following guidelines:
  - Use only ASCII characters for the names, but use non-ASCII characters in other fields, such as the **Description** and **Objective** fields.
  - The default dates displayed in the **Effective/Expiration dates** fields should appear in your locale's date format. Select new dates for each of the **Effective/Expiration dates** fields using the Calendar Tool, making sure that you select a day greater than "12" so that it will be obvious if the day is incorrectly represented as the month.
  - Verify that the dates you selected using the Calendar Tool are displayed correctly in the fields.
  - If there are custom campaign attributes existing, the field labels for these should appear in your database encoding, regardless of your default locale or user locale.
2. When you are finished with the basic campaign fields, click **Save and add a flowchart**.
3. Accept the default flowchart name, but use non-ASCII characters in the **Flowchart description** field.
4. Click **Save and edit flowchart**.
5. Verify that the campaign and flowchart are saved successfully and any campaign and flowchart labels with non-ASCII characters display correctly.
6. On the campaign's Summary tab, click **Edit summary** and modify the campaign name to use non-ASCII characters.
7. Click **Save changes** and verify that the non-ASCII characters display correctly.
8. Select the flowchart you just created, click **Edit** and rename the flowchart using non-ASCII characters.
9. Click **Save and finish** and verify that the non-ASCII characters display correctly.

## Test flowchart input and output that contains ASCII and non-ASCII characters

1. Still in the test flowchart you created in [Test campaigns and flowcharts that have ASCII and non-ASCII characters \(on page 339\)](#), click **Edit**.
2. Add a Select process to the flowchart and configure it using the following guidelines:
  - In the **Input** field, select a mapped user table. The available fields from the selected table are displayed in the **Available fields** area.
  - Select a field that you know contains non-ASCII characters, and click **Profile**.
  - Verify that the non-ASCII characters display correctly.
3. In the same Select process configuration, do another test, this time using a flat file with non-ASCII characters as the input:
  - In the **Input** field, select a select a flat file that uses non-ASCII characters. The available fields from the selected file are displayed in the **Available fields** area.
  - Verify that the non-ASCII characters display correctly.

4. On the **General** tab of the **Select process configuration** window, replace the default name in the **Process name** field with a name containing non-ASCII characters, then click **OK**.
5. Verify that the non-ASCII process name displays correctly on the process.
6. Add a Snapshot process to the flowchart and connect it so that it takes input from the existing Select process.
7. Configure the Snapshot process to **Export to file**.
8. Run the **Select > Snapshot** flowchart and locate the specified output file.
9. Verify that the output looks correct.
10. Add a Schedule process to the flowchart and configure a custom run:
  - In the **Process configuration** window, select **Custom run** from the **Schedule to run** field.
  - Use the **Calendar** to specify a date and time. For the date, choose a day greater than "12" so that it will be obvious if the day is incorrectly represented as the month.
  - Remember to click **Apply**, then click **OK** to save the date and time before closing the Calendar Tool.
11. Verify that the date and time are displayed correctly in the **Run on time** field.
12. Close the Process Configuration window and click **Save and exit**.
13. Select **Settings > Campaign settings**.

The Campaign Settings page appears.

14. Select **Manage table mappings**.
15. In the Table Mappings window, in the list of Unica Campaign system tables, select the `UA_Campaign` table and click **Browse**.
16. In the Campaign Table window, verify that non-ASCII characters display correctly.
17. In the Table Mapping window, select the `UA_Flowchart` table and verify that non-ASCII characters display correctly.
18. When you successfully complete this test, delete the test campaigns and their flowcharts, and any files that you used for testing.

## Test that the correct language directory is used

1. In Unica Campaign, select **Analytics > Unica Campaign Analytics > Calendar reports > Calendar of Campaigns**.

The Calendar of Campaigns appears. Notice that the time delineation selector that appears vertically to the right of the report (day/week/2 week/month) is an image

2. Right-click the image and select **Properties**.
3. In the Properties window for the image, examine the Address (URL) for the image.

For example, the Address might be:

```
http://localhost:7001/Campaign/de/images/calendar_nav7.gif
```

indicating that the language and locale setting is German (de).

4. Verify that the language and locale setting matches either your default application setting or your user locale preference (if any).

## Test the date formats in calendar reports

1. In Unica Campaign, click **Analytics > Unica Campaign Analytics > Calendar reports > Calendar of Campaigns**.
2. Click through the **Day**, **Week**, **2 Week**, and **Month** tabs on the right to verify that the date formats in this report are correct.

## Test that the currency symbol displays correctly for your locale

1. Select **Settings > Campaign settings**.

The Campaign Settings window appears.

2. Select **Offer template definitions**.
3. Create a new and on the **New offer template (Step 2 of 3)** page, select `Cost Per Offer` from the **Available standard and custom attributes** list, and move it to the **Selected Attributes** list.
4. Click **Next** and on the **New offer template (Step 3 of 3)** page, examine the **Cost per offer** attribute field under **Parameterized attributes**. Verify that the currency symbol appearing in parentheses is correct for your locale.
5. When you complete this test successfully, click **Cancel** because you don't need to create the offer template.

## Configuring Unica Campaign for multiple locales

Configuring Unica Campaign for more than one locale requires you to configure the system tables to support more than one locale. First, you run the appropriate unicode version of the database creation scripts when creating the system tables. Then you configure certain encoding properties, date and time formats, environment variables, and so on, depending on your database type.

### Before you begin: Unica Campaign must be installed

The information in the rest of this section is presented under the assumption that Unica Campaign is already installed AND that the Unica Campaign system tables were created with the unicode version of the database creation script appropriate for your database type. The unicode versions are located in the `<CAMPAIGN_HOME>\ddl\unicode` directory.

### Configuring for multiple locales on SQL Server

Log in to Unica and configure the encoding properties listed in the following table. Set the values for the properties as specified here.

Property	Value
<code>Campaign &gt; partitions &gt; partition[n] &gt; dataSources &gt; [data_source_name] &gt; StringEncoding</code>	WIDEUTF-8
<code>Campaign &gt; partitions &gt; partition[n] &gt; server &gt; encoding &gt; stringEncoding</code>	UTF-8

Property	Value
Campaign > unicaACLlistener > logStringEncoding	UTF-8
Campaign > unicaACLlistener >systemStringEncoding	UTF-8. If necessary, you can set more than one encoding, separated by commas, but keep UTF-8 first in the series. For example: for example, UTF-8,ISO-8859-1,CP950.
Campaign > partitions > partition[n] > dataSources > [data_source_name] > ODBCUnicode	UCS-2

For the configuration properties that specify date and time formats, accept the default values.

## Configuring for multiple locales on Oracle

When configuring for multiple locales and your system tables are Oracle, you configure encoding properties, date/time settings, environment variables, and the startup script for the Unica Campaign Listener.

### Configure encoding properties (Oracle)

When you configure Unica Campaign for multiple locales on Oracle, it is important to set the correct encoding properties.

Choose **Settings > Configuration**, then specify the encoding property values listed in the following table.

Property	Value
Unica Campaign > partitions > partition[n] > dataSources > [data_source_name] > StringEncoding	UTF-8
Unica Campaign > partitions > partition[n] > server > encoding > stringEncoding	UTF-8
Unica Campaign > unicaACLlistener > logStringEncoding	UTF-8
Unica Campaign > unicaACLlistener >systemStringEncoding	UTF-8

## Configure date/time settings (Oracle)

When you configure Unica Campaign for multiple locales on Oracle, be sure to adjust the date and time values.

Choose **Settings > Configuration**, then specify the values for the properties listed in the following table.

Property	Value
Unica Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateFormat	DELIM_Y_M_D
Unica Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateOutputFormatString	%Y-%m-%d
Unica Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateTimeFormat	DT_DELIM_Y_M_D
Unica Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateTimeOutputFormatString	%Y-%m-%d %H:%M:%S  For Japanese databases, the delimiter for the time portion must be a period (.). So, for Japanese databases set the value to:  %Y/%m/%d %H.%M.%S
Unica Campaign > partitions > partition[n] > dataSources > [data_source_name] > SQLOnConnect	<b>With Native (without Oracle ODBC)</b>  ALTER SESSION SET NLS_LANGUAGE='American' NLS_TERRITORY='America' NLS_TIMESTAMP_FORMAT='YYYY-MM-DD hh24:mi:ss'  NLS_DATE_FORMAT='YYYY-MMDD'  <b>Non-native(Via ODBC)</b>  ALTER SESSION SET NLS_LANGUAGE='American' NLS_TERRITORY='America' NLS_TIMESTAMP_FORMAT='YYYY-MM-DD HH24:MI:SSXFF' NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'

## Configure environment variables (Oracle)

On Unica Campaign client machines, set the value for the `NLS_LANG` variable as follows:

```
AMERICAN_AMERICA.UTF8
```

For example:

```
set NLS_LANG=AMERICAN_AMERICA.UTF8
```

## Configure the cmpServer.bat file (Oracle)

On the Unica Campaign client machines, modify the Unica Campaign Listener startup script as follows:

### For Windows™

Add the following line to the `cmpServer.bat` file, located in the `<CAMPAIGN_HOME>/bin` directory:

```
set NLS_LANG=AMERICAN_AMERICA.UTF8
```

From V12.1 Fixpack 4 onwards, if oracle connection with **Non-native (via ODBC)**

Add following lines in `setenv` file located in the `<CAMPAIGN_HOME>/bin` directory.

```
set NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'
```

```
set NLS_TIMESTAMP_FORMAT='YYYY-MM-DD HH24:MI:SSXFF'
```

```
set ODBCINI=<Path_to_odbc.ini>
```

### For UNIX™

Add the following lines to the `rc.unica_ac` file, located in the `<CAMPAIGN_HOME>/bin` directory:

```
NLS_LANG=AMERICAN_AMERICA.UTF8
```

```
export NLS_LANG
```

(The syntax will vary depending on operating system).

From V12.1 Fixpack 4 onwards, if oracle connection with **Non-native (via ODBC)**

```
export NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'
```

```
export NLS_TIMESTAMP_FORMAT='YYYY-MM-DD HH24:MI:SSXFF'
```

```
export ODBCINI=<Path_to_odbc.ini>
```

## Configuring for multiple locales on DB2®

To configure Unica Campaign for multiple locales when your system tables are DB2®, you must adjust encoding properties, date/time settings, environment variables, and the application server startup script.



First, identify the DB2® database code set and code page. For localized environments, the DB2® database must have the following configuration:

- Database code set = UTF-8
- Database code page = 1208

When you configure Unica Campaign, make the following adjustments:

- Set the `StringEncoding` properties to the DB2® database code set value (UTF-8), and
- Set the `DB2CODEPAGE` DB2® environment variable to the DB2® database code page value.

Both of these adjustments are explained in the following sections.

## Configure encoding properties (DB2®)

When you configure Unica Campaign for multiple locales on DB2®, it is important to set the correct encoding properties.

Choose **Settings > Configuration**, then specify the encoding property values listed in the following table.

For important information, see the property descriptions in "Setting properties for integration with Unica products" in the Unica Campaign Installation Guide.

Property	Value
Campaign > partitions > partition[n] > dataSources > [data_source_name] > StringEncoding	UTF-8
Campaign > partitions > partition[n] > server > encoding > stringEncoding	UTF-8
Campaign > unicaACLlistener > logStringEncoding	UTF-8
Campaign > unicaACLlistener > systemStringEncoding	UTF-8

## Configure date/time settings (DB2®)

When you configure Unica Campaign for multiple locales on DB2®, be sure to adjust the date and time values.

Choose **Settings > Configuration**, then specify the values for the properties listed in the following table.

Property	Value
Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateOutputFormatString	%Y-%m-%d
Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateTimeFormat	DT_DELM_Y_M_D
Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateTimeOutputFormatString	<p>%Y-%m-%d %H:%M:%S</p> <p>For Japanese databases, the delimiter for the time portion must be a period (.). So, for Japanese databases set the value to:</p> <p>%Y/%m/%d %H.%M.%S</p>

## Configure environment variables (DB2®)

To configure environment variables for DB2®, identify the DB2® database code page, then set the `DB2CODEPAGE` DB2® environment variable to the same value. For localized environments, the DB2® database code page must be 1208.

Follow these steps to set the `DB2CODEPAGE` DB2® environment variable to 1208.

1. On Windows™, add the following line to the Unica Campaign Listener startup script (`<CAMPAIGN_HOME>\bin\cmpServer.bat`):

```
db2set DB2CODEPAGE=1208
```

2. On UNIX™:

- a. After DB2® is started, the system administrator must type the following command from the DB2® instance user:

```
$ db2set DB2CODEPAGE=1208
```

After completing this step, the administrator does not need to run the `db2set DB2CODEPAGE=1208` command from the DB2® instance user again, because the value is registered for the DB2® instance user. The root user cannot run the command because the root user might not have sufficient permission.

- b. To verify the setting, type the following command and confirm that the output is 1208:

```
$ db2set DB2CODEPAGE
```

- c. To confirm that the `DB2CODEPAGE` setting works for the root user, type the following command on the `$(CAMPAIGN_HOME)/bin` directory and confirm that the output is `1208`:

```
# ./setenv.sh

# db2set DB2CODEPAGE
```

- d. Run the following command to start the Unica Campaign listener:

```
./rc.unica_ac start
```

## Configure application server startup scripts (DB2®)

If you set the code page variable as described in [Configure environment variables \(DB2\) \(on page 346\)](#), complete the following task. If you did not, the following modification is not required.

Modify your startup script for Weblogic or WebSphere®, adding the following under `JAVA_OPTIONS`:

```
-Dfile.encoding=utf-8
```

For example:

```
$(JAVA_HOME)/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
-Dfile.encoding=utf-8 -Dweblogic.Name=${SERVER_NAME}
-Dweblogic.ProductionModeEnabled=${PRODUCTION_MODE}
-Djava.security.policy="${WL_HOME}/server/lib/weblogic.policy" weblogic.Server
```

## Configuring for multiple locales on MariaDB

To configure Unica Campaign for multiple locales when your system tables or user tables are MariaDB, you must adjust encoding properties, date/time settings, environment variables, and the application server startup script.

First, identify the MariaDB database code set and code page. For localized environments, the MariaDB database must have the following configuration:

- Database code set = UTF-8

### Configuring encoding properties (MariaDB)

Choose **Settings > Configuration**, then specify the encoding property values listed in the following table.

For important information, see the property descriptions in "Setting properties for integration with Unica products" in the Campaign Installation Guide.

Property	Value
Campaign > partitions > partition[n] > dataSources > [data_source_name] > StringEncoding	UTF-8
Campaign > partitions > partition[n] > server > encoding > stringEncoding	UTF-8
Campaign > unicaACLlistener > logStringEncoding	UTF-8
Campaign > unicaACLlistener > systemStringEncoding	UTF-8

### Configure date/time settings (MariaDB)

When you configure Unica Campaign for multiple locales on MariaDB, be sure to adjust the date and time values.

Choose **Settings > Configuration**, then specify the values for the properties listed in the following table.

Property	Value
Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateOutputFormatString	%Y-%m-%d
Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateTimeFormat	DT_DELIM_Y_M_D
Campaign > partitions > partition[n] > dataSources > [data_source_name] > DateTimeOutputFormatString	<p>%Y-%m-%d %H:%M:%S</p> <p>For Japanese databases, the delimiter for the time portion must be a period (.). So, for Japanese databases set the value to:</p> <p>%Y/%m/%d %H.%M.%S</p>

### Configuring for multiple locales on Postgres

To configure Unica Campaign for multiple locales when your system tables or user tables are Postgres, you must adjust encoding properties, date/time settings, environment variables, and the application server startup script. First, identify the Postgres database code set and code page. For localized environments, the Postgres database must have the following configuration:

- Database code set = UTF-8

Configuring encoding properties (Postgres)

Choose **Settings > Configuration**, then specify the encoding property values listed in the following table.

For important information, see the property descriptions in "Setting properties for integration with Unica products" in the Campaign Installation Guide.

Property	Value
<b>Campaign &gt; partitions &gt; partition[n] &gt; dataSources &gt; [data_source_name]</b> StringEncoding	UTF-8
<b>Campaignpartitionspartition[n]serverencodingstringEncoding</b>	UTF-8
<b>Campaign &gt; unicaACLlistener &gt; logStringEncoding</b>	UTF-8
<b>Campaign &gt; unicaACLlistener &gt; systemStringEncoding</b>	UTF-8

Configure date/time settings (Postgres)

When you configure Unica Campaign for multiple locales on Postgres, be sure to adjust the date and time values.

Choose **Settings > Configuration**, then specify the values for the properties listed in the following table.

Property	Value
<b>Campaign &gt; partitions &gt; partition[n] &gt; dataSources &gt; [data_source_name] &gt; DateOutputFormatString</b>	%Y-%m-%d
<b>Campaignpartitionspartition[n]dataSources[data_source_name]DateTimeFormat</b>	DT_DELIM_Y_M_D
<b>Campaign &gt; partitions &gt; partition[n] &gt; dataSources &gt; [data_source_name] &gt; DateTimeOutputFormatString</b>	%Y-%m-%d %H:%M:%S For Japanese databases, the delimiter for the time portion must be a period (.). So, for Japanese databases set the value to: %Y/%m/%d %H.%M.%S

## Configuring for multiple locales on OneDB

To configure Unica Campaign for multiple locales when your system tables or user tables are OneDB, you must adjust encoding properties, date/time settings, environment variables, and the application server startup script.

## Settings for Campaign Non-Ascii in UA\_system\_tables

Select Settings > Configurations and then specify the values for string related properties listed in the following table.

Property	Values
Campaign > partitions > partitions[n] > dataSource >[data_source_name] > StringEncoding	WIDEUTF-8
Campaign > partitions > partitions[n] > dataSource >[data_source_name] > ODBCUnicode	UCS-2
Campaign > partitions > partitions[n] > server >encoding > stringEncoding	UTF-8
Campaign > unicaACLlistener > logStringEncoding	UTF-8
Campaign > unicaACLlistener > systemStringEncoding	UTF-8

Select Settings -> Configurations, then specify the values for date related properties listed in the following table.

Property	Values
Campaign > partitions > partition[n] > dataSource >[data_source_name] > DateFormat	DELIM_Y_M_D
Campaign > partitions > partition[n] > dataSource >[data_source_name] > DateOutputFormatString	%Y-%m-%d
Campaign > partitions > partition[n] > dataSource >[data_source_name] > DateTimeFormat	DT_DELIM_Y_M_D
Campaign > partitions > partition[n] > dataSource >[data_source_name] > DateTimeOutputFormatString	%Y-%m-%d %H:%M:%S

# Chapter 22. Unica Campaign configuration properties

The configuration properties of Unica Campaign are located at **SettingsConfiguration**. For more details on the configuration properties, see the Campaign Administrator Guide.


## Campaign

To specify the locales and component applications that your installation supports, choose **Settings > Configuration**, then click the Unica Campaign category.

### currencyLocale

#### Description

The `currencyLocale` property is a global setting that controls how currency is displayed in the Unica Campaign web application, regardless of the display locale.

 **Important:** No currency conversion is performed by Unica Campaign when the display locale changes (for example, if the multi-locale feature is implemented and the display locale changes based on user-specific locales). You must be aware that when a locale is switched, for example, from English US, in which a currency amount is, for example, US\$10.00, to a French locale, the currency amount is unchanged (10,00) even if the currency symbol changes with the locale.

#### Default value

en\_US

### supportedLocales

#### Description

The `supportedLocales` property specifies the locales or language-locale pairs that Unica Campaign supports. The value of this property is set by the installer when you install Unica Campaign. For example: de,en,fr,ja,es,ko,pt,it,zh,ru.

#### Default value

All languages/locales into which Unica Campaign has been localized.

### defaultLocale

#### Description

The `defaultLocale` property specifies which of the locales specified in the `supportedLocales` property is considered the default display locale for Unica Campaign. The value of this property is set by the installer when you install Unica Campaign.

#### Default value

en

## **acoInstalled**

### **Path**

### **Description**

The `acoInstalled` property specifies whether Unica Optimize is installed.

When Unica Optimize is installed and configured, set the value to `yes`, which causes the Unica Optimize process to be displayed in flowcharts. If the value is `true` and Unica Optimize is not installed or configured, the process is displayed but disabled (grayed out).

### **Default value**

`false`

### **Valid Values**

`false` and `true`

## **collaborateInstalled**

### **Description**

The `collaborateInstalled` property specifies whether Unica Collaborate is installed. When Unica Collaborate is installed and configured, set the value to `true`, which causes the Unica Collaborate features to be available in the Unica Campaign user interface.

### **Default value**

`false`

### **Valid Values**

`true` | `false`

## **Campaign | collaborate**

The properties in this category pertain to Unica Collaborate configuration.

## **CollaborateIntegrationServicesURL**

### **Description**

The `CollaborateIntegrationServicesURL` property specifies the server and port number of Unica Collaborate. This URL is used by Unica Campaign when a user publishes a flowchart to Unica Collaborate.

### **Default value**

`http://localhost:7001/collaborate/services/CollaborateIntegrationServices1.0`



## Campaign | navigation

Some of the properties in this category are used internally and should not be changed.

### welcomePageURI

**Configuration category**

Campaign | navigation

**Description**

The `welcomePageURI` property is used internally by other applications. It specifies the Uniform Resource Identifier of the Unica Campaign index page. You should not change this value.

**Default value**

No default value defined.

### seedName

**Configuration category**

Campaign | navigation

**Description**

The `seedName` property is used internally by other applications. You should not change this value.

**Default value**

No default value defined.

### type

**Configuration category**

Campaign | navigation

**Description**

The `type` property is used internally by other applications. You should not change this value.

**Default value**

No default value defined.

### httpPort

**Configuration category**

Campaign | navigation

**Description**

This property specifies the port used by the Unica Campaign web application server. If your installation of Unica Campaign uses a port that is different from the default, you must edit the value of this property.

#### Default value

7001

### httpsPort

#### Configuration category

Campaign|navigation

#### Description

If SSL is configured, this property specifies the port used by the Unica Campaign web application server for secure connections. If your installation of Unica Campaign uses a secure port that is different from the default, you must edit the value of this property.

#### Default value

7001

### serverURL

#### Configuration category

Campaign|navigation

#### Description

The `serverURL` property specifies the URL used by Unica Campaign. If your installation of Unica Campaign has a URL that is different from the default, you should edit the value as follows:

```
http://machine_name_or_IP_address:port_number/context-root
```

If users access Unica Campaign with the Chrome browser, use the fully qualified domain name (FQDN). If the FQDN is not used, the Chrome browser cannot access the product URLs.

#### Default value

http://localhost:7001/Campaign

### logoutURL

#### Configuration category

Campaign|navigation

#### Description

The `logoutURL` property is used internally to call the logout handler of the registered application if the user clicks the logout link. Do not change this value.

### serverURLInternal

#### Configuration category

**Campaign|navigation****Description**

The `serverURLInternal` property specifies the URL for the Unica Campaign web application when SiteMinder is used; this property is also used for internal communication with other Unica applications, such as Unica Deliver and Unica Interact. If the property is empty, the value in the `serverURL` property is used. Modify this property if you need internal application communication to be http and external communication to be https. If you use SiteMinder, you must set this value to the URL for the Unica Campaign web application server, formatted as follows:

```
http://machine_name_or_IP_address:port_number/context-root
```

**Default value**

No default value defined.

**campaignDetailPageURI****Configuration category**

**Campaign|navigation**

**Description**

The `campaignDetailPageURI` property is used internally by other applications. It specifies the Uniform Resource Identifier of the Unica Campaign detail page. You should not change this value.

**Default value**

campaignDetails.do?id=

**flowchartDetailPageURI****Configuration category**

**Campaign|navigation**

**Description**

The `flowchartDetailPageURI` property is used to construct a URL to navigate to the details of a flowchart in a specific campaign. You should not change this value.

**Default value**

flowchartDetails.do?campaignID=&id=

**schedulerEditPageURI****Configuration category**

**Campaign|navigation**

**Description**

This property is used to construct a URL to navigate to the Scheduler page. Do not change this value.

**Default value**

jsp/flowchart/scheduleOverride.jsp?taskID=

**offerDetailPageURI**

**Configuration category**

Campaign | navigation

**Description**

The `offerDetailPageURI` property is used to construct a URL to navigate to the details of a specific offer. You should not change this value.

**Default value**

offerDetails.do?id=

**offerlistDetailPageURI**

**Configuration category**

Campaign | navigation

**Description**

The `offerlistDetailPageURI` property is used to construct a URL to navigate to the details of a specific offer list. You should not change this value.

**Default value**

displayOfferList.do?offerListId=

**mailingDetailPageURI**

**Configuration category**

Campaign | navigation

**Description**

This property is used to construct a URL to navigate to the mailing details page for Unica Deliver. Do not change this value.

**Default value**

view/MailingDetails.do?mailingId=

**optimizeDetailPageURI**

**Configuration category**

Campaign | navigation

**Description**

This property is used to construct a URL to navigate to Unica Optimize details page. Do not change this value.

**Default value**

optimize/sessionLinkClicked.do?optimizeSessionID=

**optimizeSchedulerEditPageURI****Configuration category**

Campaign|navigation

**Description**

This property is used to construct a URL to navigate to the Unica Optimize Scheduler edit page. Do not change this value.

**Default value**

optimize/editOptimizeSchedule.do?taskID=

**displayName****Configuration category**

Campaign|navigation

**Description**

The `displayName` property specifies the link text used for the Unica Campaign link in the drop-down menu that exists in the GUI of each product.

**Default value**

Campaign

## Campaign | caching

Certain objects, such as offers, are cached in the web application server to improve response times in the Unica Campaign user interface. The `Campaign|caching` configuration properties specify the length of time that cached data is retained. Smaller values result in more frequent cache updates, which can adversely affect performance by consuming processing resources on both the web server and the database.

**offerTemplateDataTTLSeconds****Configuration category**

Campaign|caching

**Description**

The `offerTemplateDataTTLSeconds` property specifies the length of time, in seconds, that the system retains the Offer Template cache data (Time to Live). An empty value means the cache data is never purged.

**Default value**

600 (10 minutes)

**campaignDataTTLSeconds**

**Configuration category**

Campaign | caching

**Description**

The `campaignDataTTLSeconds` property specifies the length of time, in seconds, that the system retains the Unica Campaign cache data (Time to Live). An empty value means the cache data is never purged.

**Default value**

600 (10 minutes)

**sessionDataTTLSeconds**

**Configuration category**

Campaign | caching

**Description**

The `sessionDataTTLSeconds` property specifies the length of time, in seconds, that the system retains the Session cache data (Time to Live). An empty value means the cache data is never purged.

**Default value**

600 (10 minutes)

**folderTreeDataTTLSeconds**

**Configuration category**

Campaign | caching

**Description**

The `folderTreeDataTTLSeconds` property specifies the length of time, in seconds, that the system retains the Folder Tree cache data (Time to Live). An empty value means the cache data is never purged.

**Default value**

600 (10 minutes)

**attributeDataTTLSeconds**

**Configuration category**

Campaign|caching

### Description

The `attributeDataTTLSeconds` property specifies the length of time, in seconds, that the system retains the Offer Attribute cache data (Time to Live). An empty value means the cache data is never purged.

### Default value

600 (10 minutes)

## initiativeDataTTLSeconds

### Configuration category

Campaign|caching

### Description

The `initiativeDataTTLSeconds` property specifies the length of time, in seconds, that the system retains the Initiative cache data (Time to Live). An empty value means the cache data is never purged.

### Default value

600 (10 minutes)

## offerDataTTLSeconds

### Configuration category

Campaign|caching

### Description

The `offerDataTTLSeconds` property specifies the length of time, in seconds, that the system retains the Offer cache data (Time to Live). An empty value means the cache data is never purged.

### Default value

600 (10 minutes)

## segmentDataTTLSeconds

### Configuration category

Campaign|caching

### Description

The `segmentDataTTLSeconds` property specifies the length of time, in seconds, that the system retains the Segment cache data (Time to Live). An empty value means the cache data is never purged.

### Default value

600 (10 minutes)

## Campaign | partitions

This category contains properties to configure Unica Campaign partitions, including the default partition, which is named partition1.

One category should be created for each Unica Campaign partition. This section describes the properties in the partition[n] category, which apply to all partitions that you configure in Unica Campaign.

### Campaign | partitions | partition[n] | WatsonAssistant

These properties control authentication between Unica Campaign and IBM WatsonAssistant if the products are integrated.

To access these properties, choose **Settings > Configuration**. If your Unica Campaign installation has multiple partitions, set these properties for each partition that uses the integration.

#### Enable Watson Assistant

##### Configuration category

Campaign | partitions | partition[n] | WatsonAssistant

##### Description

Integration will be enabled and Watson assistant link will be visible only if this integration flag is true.

##### Default value

False

##### Example

#### Watson Assistant Service URL

##### Configuration category

Campaign | partitions | partition[n] | WatsonAssistant

##### Description

The `Watson Assistant Service URL` indicates the URL where Campaign can access the IBM® Watson Marketing Assistant application. The administrator must provide this value.

##### Default value

`http://localhost:9090/static/unica.js`

##### Example

`http://host:port/static/unica.js`

#### Watson Assistant Credential Holder

##### Configuration category

Campaign | partitions | partition[n] | WatsonAssistant



**Description**

Specify Campaign user whose data source is going to be used for connecting to WatsonAssistant.

**Default value****Example****Watson Assistant Credential DataSource****Configuration category**

Campaign | partitions | partition[n] | WatsonAssistant

**Description**

Data source that will hold the value of shared secret, which is used to sign the JWT token

EBA supports HS256 and RS256 as JWT signing algorithms. Campaign - Watson Marketing assistant integration is implemented based on HS256 algorithm. Value for the parameter UNICA\_JWT\_ALGORITHM must be set to HS256 under EBA environment configuration.

**Default value**

WatsonAssistant\_DS

**Example****tokenExpirationDuration****Configuration category**

Campaign | partitions | partition[n] | WatsonAssistant

**Description**

Data source that will hold the value of shared secret, which is used to sign the JWT token

**Default value**

10

**Example****Campaign | partitions | partition[n] | Deliver | contactAndResponseHistTracking**

Use the properties in this category to configure Unica Deliver offer integration with Unica Campaign for the current partition.

**etlEnabled****Description**

Unica Campaign uses its own ETL process to extract, transform, and load offer response data from the Unica Deliver tracking tables into the Unica Campaign contact and response history tables.

The ETL process coordinates information across the necessary tables, including `UA_UsrResponseType` (Unica Campaign response types) and `UA_RespTypeMapping` (mapping of response types between Unica Campaign and Unica Deliver).

Setting the value to `Yes` ensures that information about Unica Deliver offer contact and response history is coordinated between Unica Campaign and Unica Deliver. For example, email response data will be included in Unica Campaign reports.



**Note:** You must also set `Unica Campaign | partitions | partition[n] | server | internal | deliverInstalled` to `Yes` for this partition or the ETL process will not run.



**Tip:** If you want to monitor the progress of the ETL, enable `Unica Campaign | monitoring | monitorEnabledForDeliver`.

#### Default value

No

#### Valid values

Yes | No

### runOnceADay

#### Description

Indicate whether the ETL process should run only once a day.

If the value is `Yes`: You must specify a **startTime**; the ETL job then runs until all records are processed; and the **sleepIntervalInMinutes** is ignored.

If the value is `No`: The ETL job starts as soon as the Campaign web server starts. The ETL job stops after all of the records are processed, then waits for the time specified by **sleepIntervalInMinutes**.

#### Default value

No

#### Valid values

Yes | No

### batchSize

#### Description

The ETL process uses this parameter to fetch records that have been downloaded by the RCT into the local Unica Deliver system tables. Because large values can impact performance, the list of available values is restricted to the valid values shown below. If you anticipate large volumes of records, adjust the **batchSize** together with the **sleepIntervallnMinutes** to process records at regular intervals.

#### Default value

100

#### Valid values

100 | 200 | 500 | 1000

### sleepIntervallnMinutes

#### Description

Specify the interval in minutes between ETL jobs. This option determines the wait time after a job finishes. The ETL process waits for this duration before starting the next job. Multiple jobs can run synchronously and there may be multiple ETL jobs per partition.

If **runOnceADay** is *Yes*, you cannot set a sleep interval.

#### Default value

60

#### Valid values

Positive integers

### startTime

#### Description

Specify a time to start the ETL job. You must use the English locale format to specify the start time.

#### Default value

12:00:00 AM

#### Valid values

Any valid time in the format `hh:mm:ss AM/PM`

### notificationScript

#### Description

An optional executable or script file that is run after each ETL job is done. For example, you might want to be notified of the success or failure of each ETL job, for monitoring purposes. The notification script runs every time the ETL job for a given partition finishes running.

The parameters passed in to this script are fixed and cannot be changed. The following parameters can be used by the script:

- etlStart: The start time of ETL in number of milliseconds.
- etlEnd: The end time of ETL in number of milliseconds.
- totalCHRecords: Total number of contact records processed.
- totalRHRecords: Total number of response history records processed.
- executionStatus: Execution status of the ETL with value 1 (failed) or 0 (succeeded).

#### Default value

No default value defined.

#### Valid values

Any valid path that the Unica Campaign server can access with Read and Execute permissions. For example: `D:\myscripts\scriptname.exe`

## Campaign | partitions | partition[n] | Deliver | sendTimeOptimization

Use the property in this category to run Deliver STO ETL which will determine the best time to send a message to users when they are most likely to open the message. Deliver STO ETL job determines optimal send time of all the recipients contacted based on their email open responses.

### isDeliverSTOETLEnabled

#### Description

Setting the value to `Yes` ensures that Deliver STO ETL job runs periodically to process optimal send time of each recipient contacted..



**Note:** You must also set `Unica Campaign | partitions | partition[n] | server | internal | deliverInstalled` to `Yes` for this partition or the ETL process will not run.



**Tip:** If you want to monitor the progress of the ETL, enable `Unica Campaign | monitoring | monitorEnabledForDeliver`.

#### Default value

No

#### Valid values

Yes | No

## runFrequency

#### Description

Indicates the frequency of running this process. It can be run either on a monthly or a weekly basis.

**Default value**

Weekly

**Valid values**

Monthly | Weekly

**runWeek****Description**

Indicates the frequency of running this ETL process on a weekly basis, in case runFrequency is chosen as Monthly, otherwise this is ignored.

**Default value**

1

**Valid values**

1 | 2 | 3 | 4

**runDay****Description**

Indicates the day the ETL process should be run.

**Default value**

Sunday

**Valid values**

Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday

**startTime****Description**

Specify a time to start the ETL job. You must use the English locale format to specify the start time.

**Default value**

12:00:00 AM

**Valid values**

Any valid time in the format `hh:mm:ss AM/PM`

**pastDaysToConsiderForSTO****Description**

Stores the past days of customer to calculate the STO.

**Default value**

90

**Valid values**

Positive integers

## Campaign | partitions | partition[n] | Acoustic Campaign

These properties control authentication and data exchange between Unica Campaign and Engage if the products are integrated.

To access these properties, choose **Settings > Configuration**. If your Unica Campaign installation has multiple partitions, set these properties for each partition that uses the integration.

### Service URL

**Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `Service URL` indicates the URL where Unica Campaign can access the Acoustic Campaign application. The Acoustic Campaign Org Admin must provide this value.

**Default value**

<none>

**Example**

`https://engageapi.abc01.com/`

### OAuth URL Suffix

**Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `OAuth URL Suffix` specifies the authentication token for the Acoustic Campaign APIs.

**Default value**

oauth/token

### API URL Suffix

**Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `API URL Suffix` is set to `XMLAPI` to ensure that Unica Campaign uses the Acoustic Campaign XML APIs. Best practice is to leave this set to the default value.

**Default value**

XMLAPI

**Platform User with Data Sources for Acoustic Campaign Credentials****Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `Platform User with Data Sources for Acoustic Campaign Credentials` indicates the name of the Unica Platform user account that is allowed to connect to the Acoustic Campaign server. This user account contains the data sources that provide Acoustic Campaign credentials. Typically, `asm_admin` is used.

**Default value**

No default value defined.

**Valid values**

The Unica Platform user account that contains the data sources for Acoustic Campaign integration credentials.

**Data Source for Client ID****Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `Data Source for Client ID` value must exactly match the name of the Acoustic Campaign Client ID data source that was created for the user account that connects to the Acoustic Campaign server (**Platform User with Data Sources for Acoustic Campaign Credentials**). In other words, the value must match what is set up as the `datasource` for the Unica Platform user. Best practice is to leave this set to the default value.

**Default value**

ENGAGE\_CLIENT\_ID\_DS

**Data Source for Client Secret****Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `Data Source for Client Secret` value must exactly match the name of the Acoustic Campaign Client Secret data source that was created for the user account that connects to the Acoustic Campaign server (**Platform User with Data Sources for Acoustic Campaign Credentials**). Best practice is to leave this set to the default value.

**Default value**

ENGAGE\_CLIENT\_SECRET\_DS

## Data Source for Client Refresh Token

**Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `Data Source for Client Refresh Token` value must exactly match the name of the Acoustic Campaign Client Refresh Token data source that was created for the user account that connects to the Acoustic Campaign server (**Platform User with Data Sources for Acoustic Campaign Credentials**). Best practice is to leave this set to the default value.

**Default value**

ENGAGE\_CLIENT\_REF\_TOK\_DS

## Data Source for File Transfer Credentials

**Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `Data Source for File Transfer Credentials` indicates the name of the data source that provides the credentials for FTP communication between Campaign and Acoustic Campaign. This value must exactly match the name of the Acoustic Campaign FTP data source that was created for the user account that connects to the Acoustic Campaign server (**Platform User with Data Sources for Acoustic Campaign Credentials**). Best practice is to leave this set to the default value.

**Default value**

ENGAGE\_FTP\_DS

## Host Name for File Transfer

**Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**



The `Host Name for File Transfer` indicates the host name of the Acoustic Campaign FTP server where Campaign uploads the contact list in TSV format. This file gets deleted automatically after it is uploaded into a contact list.

**Default value**

<none>

**Valid values**

Any valid address in the list of IBM® Marketing Cloud FTP addresses.. For example:

transfer2.silverpop.com

## Port Number for File Transfer

**Configuration category**

Campaign | partitions | partition[n] | Acoustic Campaign

**Description**

The `Port Number for File Transfer` indicates the port number for the FTP server that is specified in **Host Name for File Transfer**.

**Default value**

22

**Valid values**

Any valid FTP port number

## Use proxy for ServiceURL

**Description**

Determine if you use proxy for ServiceURL. If you select `Yes`, your connection uses the proxy server. Proxy server details can be configured under Campaign | proxy. If you select `No`, a proxy server is not used to connect to Acoustic Campaign.

**Default value**

No

**Valid values**

Yes, No

## Use proxy for FTP

**Description**

Determine if you use proxy for FTP. If you select `Yes`, your connection to the Acoustic Campaign FTP server uses the proxy server. Proxy server details can be configured under Campaign | proxy. If you select `No`, a proxy server is not used to connect to the Acoustic Campaign FTP server.

**Default value**

No

**Valid values**

Yes, No

**allowableFailurePercentage**

**Description**

When a campaign contact list is processed in Acoustic Campaign, an error is returned to Campaign if a contact fails for any reason. Use the `allowableFailurePercentage` property to specify the percentage of failed records that are allowed before the campaign process fails. If the percentage of failed records is greater than the configured `allowableFailurePercentage`, the process fails.

If the percentage of failed records is less than or equal to the configured `allowableFailurePercentage`, the process does not fail. Duplicate records are considered as valid records and so they do not affect the `allowableFailurePercentage` for a process box. All valid records are imported into Acoustic Campaign.

The percentage of failed records is calculated as  $([TOTAL\_ROWS - TOTAL\_VALID] / TOTAL\_ROWS) * 100$ .

Any warnings and errors for the process are logged in the `ImportErrors` log file. You can delete this file at regular intervals. The `ImportErrors` log file is located in the `<Campaign_home>/logs` folder.

**Default value**

0

**Valid values**

Integers between 0-100.

**Campaign | partitions | partition[n] | Acoustic Campaign | contactAndResponseHistTracking**

These properties specify the ETL of events that are downloaded from UBX in the Unica Campaign history tables.

To access these properties, choose **Settings > Configuration**. If your Unica Campaign installation has multiple partitions, set these properties for each partition that uses the integration.

**etlEnabled**

**Description**

Determine whether you want to enable the ETL transfer of data from the events table in the Unica Campaign history table.

**Default value**

No

**Valid values**

Yes, No

## runOnceADay

### Description

Determine whether the ETL runs once a day. It can run repeatedly if you specify the `sleepIntervallInMinutes` property. If `runOnceADay` is set to `yes`, ETL runs once a day at the specified time.

### Valid values

Yes, No

## batchSize

### Description

The number of records that are processed in one ETL cycle.

### Default value

100

### Valid values

100, 200, 500, 1000, 10000, 100000

## sleepIntervallInMinutes

### Description

Specify the number of minutes the ETL waits it runs again. This value is used when `runOnceADay` is set to `No`.

### Default value

60

### Valid values

Positive integers.

## startTime

### Description

When `runOnceADay` is set to `Yes`, this property determines the ETL run start.

### Default value

12:00:00 AM

### Valid values

Any valid time in the format `hh:mm:ss AM/PM`.

## notificationScript

### Description

Enter any script that can run after the ETL execution is complete.

#### Default value

No default value defined.

#### Valid values

Any valid path that the Campaign server can access with Read and Execute permissions. Example: `D:\myscripts\scriptname.exe`

## Campaign | partitions | partition[n] | Coremetrics

The properties in this category specify integration settings for Digital Analytics and Unica Campaign for the selected partition.

If your Unica Campaign installation has multiple partitions, set these properties for each partition that you want to affect. For these properties to take effect, `UC_CM_integration` must be set to `Yes` for the partition (under `partitions | partition[n] | server | internal`).

### ServiceURL

#### Description

The `ServiceURL` specifies the location of the Digital Analytics integration service that provides the integration point between Digital Analytics and Unica Campaign. Note that the default port for https is 443.

#### Default value

`https://export.coremetrics.com/eb/segmentapi/1.0/api.do`

#### Valid values

The only supported value for this release is the default value shown above.

### CoremetricsKey

#### Description

Unica Campaign uses the `CoreMetricsKey` to map IDs exported from Digital Analytics to the corresponding Audience ID in Unica Campaign. The value defined for this property must exactly match the value used in the translation table.

#### Default value

`registrationid`

#### Valid values

The only supported value for this release is `registrationid`.

## ClientID

### Description

Set this value to the unique Digital Analytics Client ID assigned to your company.

### Default value

No default value defined.

## TranslationTableName

### Description

Specify the name of the translation table being used to translate Digital Analytics keys to Unica Campaign Audience IDs. For example, `Cam_CM_Trans_Table`. If you do not specify a table name, an error will occur if users run a flowchart that uses Digital Analytics segments as input, because without the table name, Unica Campaign does not know how to map IDs from one product to the other.



**Note:** When you map or re-map a translation table, the **Table Name** assigned in the Table Definition dialog must exactly match (including case) the `TranslationTableName` defined here.

### Default value

No default value defined.

## ASMUserForCredentials

### Description

The `ASMUserForCredentials` property specifies which Unica account is allowed to access the Digital Analytics integration service. See below for additional information.

If no value is specified, Unica Campaign checks the currently logged-in user's account to see if the `ASMDatasourceForCredentials` value is specified as a data source. If it is, then access is allowed. If not, access is denied.

### Default value

`asm_admin`

## ASMDatasourceForCredentials

### Description

The `ASMDatasourceForCredentials` property identifies the data source assigned to the Unica Platform account specified in the **ASMUserForCredentials** setting. The default is `UC_CM_ACCESS`. This "data source for credentials" is the mechanism that Unica Platform uses to store the credentials that provide access to the integration service.

Although a default value of `UC_CM_ACCESS` is supplied, a data source of that name is not provided, nor do you have to use that name.



**Important:** You must choose **Settings > Users**, select the user specified in

`ASMUserForCredentials`, click the **Edit Data Sources** link, and add a new data source whose name exactly matches the value defined here (for example, `UC_CM_ACCESS`). For Data Source Login and Data Source Password, use the credentials associated with your Digital Analytics Client ID. For information about data sources, user accounts, and security, see the *Unica Platform Administrator's Guide*

#### Default value

`UC_CM_ACCESS`

## Campaign | partitions | partition[n] | reports

The **Campaign | partitions | partition[n] | reports** property defines the different types of folders for reports.

### offerAnalysisTabCachedFolder

#### Description

The `offerAnalysisTabCachedFolder` property specifies the location of the folder that contains the specification for bursted (expanded) offer reports listed on the Analysis tab when you reach it by clicking the Analysis link on the navigation pane. The path is specified by using the XPath notation.

#### Default value

```
/content/folder[@name='Affinium Campaign - Object Specific Reports']/folder[@name='offer']/
folder[@name='cached']
```

### segmentAnalysisTabOnDemandFolder

#### Description

The `segmentAnalysisTabOnDemandFolder` property specifies the location of the folder that contains the segment reports listed on the Analysis tab of a segment. The path is specified by using the XPath notation.

#### Default value

```
/content/folder[@name='Affinium Campaign - Object Specific Reports']/folder[@name='segment']/
folder[@name='cached']
```

### offerAnalysisTabOnDemandFolder

#### Description

The `offerAnalysisTabOnDemandFolder` property specifies the location of the folder that contains the offer reports listed on the Analysis tab of an offer. The path is specified by using the XPath notation.

**Default value**

```
/content/folder[@name='Affinium Campaign - Object Specific Reports']/folder[@name='offer']
```

## segmentAnalysisTabCachedFolder

**Description**

The `segmentAnalysisTabCachedFolder` property specifies the location of the folder that contains the specification for bursted (expanded) segment reports listed on the Analysis tab when you reach it by clicking the Analysis link on the navigation pane. The path is specified by using the XPath notation.

**Default value**

```
/content/folder[@name='Affinium Campaign - Object Specific Reports']/folder[@name='segment']
```

## analysisSectionFolder

**Description**

The `analysisSectionFolder` property specifies the location of the root folder where report specifications are stored. The path is specified by using the XPath notation.

**Default value**

```
/content/folder[@name='Affinium Campaign']
```

## campaignAnalysisTabOnDemandFolder

**Description**

The `campaignAnalysisTabOnDemandFolder` property specifies the location of the folder that contains the campaign reports listed on the Analysis tab of a campaign. The path is specified by using the XPath notation.

**Default value**

```
/content/folder[@name='Affinium Campaign - Object Specific Reports']/folder[@name='campaign']
```

## campaignAnalysisTabCachedFolder

**Description**

The `campaignAnalysisTabCachedFolder` property specifies the location of the folder that contains the specification for bursted (expanded) campaign reports listed on the Analysis tab when you reach it by clicking the Analysis link on the navigation pane. The path is specified by using the XPath notation.

**Default value**

```
/content/folder[@name='Affinium Campaign - Object Specific Reports']/folder[@name='campaign']/  
folder[@name='cached']
```

## campaignAnalysisTabDeliverOnDemandFolder

### Description

The `campaignAnalysisTabDeliverOnDemandFolder` property specifies the location of the folder that contains the Unica Deliver reports listed on the Analysis tab of a campaign. The path is specified by using the XPath notation.

### Default value

```
/content/folder[@name='Affinium Campaign']/folder[@name='Deliver Reports']
```

## campaignAnalysisTabInteractOnDemandFolder

### Description

Report server folder string for Unica Interact reports.

### Default value

```
/content/folder[@name='Affinium Campaign']/folder[@name='Interact Reports']
```

### Availability

This property is applicable only if you install Unica Interact.

## interactiveChannelAnalysisTabOnDemandFolder

### Description

Report server folder string for Interactive Channel analysis tab reports.

### Default value

```
/content/folder[@name='Affinium Campaign - Object Specific Reports']/folder[@name='interactive  
channel']
```

### Availability

This property is applicable only if you install Unica Interact.

## Campaign | partitions | partition[n] | validation

The Validation Plugin Development Kit (PDK), delivered with Unica Campaign, allows third parties to develop custom validation logic for use in Unica Campaign. Properties in the `partition[n] > validation` category specify the classpath and class name of the custom validation program, and an optional configuration string.

### validationClass

#### Description



The `validationClass` property specifies the name of the class used for validation in Unica Campaign. The path to the class is specified in the `validationClasspath` property. The class must be fully qualified with its package name.

For example:

```
com.unica.campaign.core.validation.samples.SimpleCampaignValidator
```

indicates the `SimpleCampaignValidator` class from the sample code.

This property is undefined by default, which causes Unica Campaign to perform no custom validation.

#### Default value

No default value defined.

### validationConfigString

#### Description

The `validationConfigString` property specifies a configuration string that is passed into the validation plugin when Unica Campaign loads it. The use of the configuration string may vary, depending on the plugin used.

This property is undefined by default.

#### Default value

No default value defined.

### validationClasspath

#### Description

The `validationClasspath` property specifies the path to the class used for custom validation in Unica Campaign.

- Use either a full path or a relative path. If the path is relative, the behavior depends on the application server that is running Unica Campaign. WebLogic uses the path to the domain work directory, which by default is `c:\bea\user_projects\domains\mydomain`.
- If the path ends in a slash (forward slash / for UNIX™ or backslash \ for Windows™), Unica Campaign assumes that it points to the location of the Java™ plug-in class that should be used.
- If the path does not end in a slash, Unica Campaign assumes that it is the name of a .jar file that contains the Java™ class. For example, the value `/<CAMPAIGN_HOME>/devkits/validation/lib/validator.jar` is the path on a UNIX™ platform that points to the JAR file that is provided with the plug-in developer's kit.

This property is undefined by default, which causes the property to be ignored.

#### Default value

No default value defined.

## Campaign | partitions | partition[n] | audienceLevels | audienceLevel

Do not edit properties in this category. These properties are created and populated when a user creates audience levels on the Administration page in Unica Campaign.

### numFields

#### Description

This property indicates the number of fields in the audience level. Do not edit this property.

#### Default value

No default value defined.

### audienceName

#### Description

This property indicates the audience name. Do not edit this property.

#### Default value

No default value defined.

## Campaign | partitions | partition[n] | audienceLevels | audienceLevel | field[n]

Properties in this category define an audience level field. These properties are populated when a user creates audience levels on the Administration page in Unica Campaign. You should not edit properties in this category.

### type

#### Description

The `partition[n] > audienceLevels > audienceLevel > field[n] > type` property is populated when a user creates audience levels on the Administration page in Unica Campaign. You should not edit this property.

#### Default value

No default value defined.

### name

#### Description

The `partition[n] > audienceLevels > audienceLevel > field[n] > name` property is populated when a user creates audience levels on the Administration page in Unica Campaign. You should not edit this property.

#### Default value

No default value defined.

## Campaign | partitions | partition[n] | dataSources

The properties in `Campaign|partitions|partition[n]|dataSources` determine how Unica Campaign interacts with databases, including its own system tables, for the specified partition.

These properties specify the databases that Unica Campaign can access and they control many aspects of how queries are formed.

Each data source that you add in Unica Campaign is represented by a category under `Campaign|partitions|partition[n]|dataSources|<data-source-name>`.



**Note:** The Unica Campaign system tables data source for each partition must be named `UA_SYSTEM_TABLES` in Unica Platform, and every Unica Campaign partition must have a **dataSources | UA\_SYSTEM\_TABLES** category on the Configuration page.

### New category name

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

Use the `New category name` field when you create a data source by clicking on one of the provided templates. The provided templates are indicated by italics and parentheses, for example (DB2® Template). Enter a category name to identify the data source, such as `DB2_Customers`. After you save a new category, it appears in the navigation tree. You can change its properties as needed. The properties that are available depend on which template you selected. All of the possible properties for all templates are listed below in alphabetical order.

### AccessLibrary

#### Description

Unica Campaign chooses its data source access library according to the data source type. For example, `libora4d.so` is used for Oracle connectivity, while `libdb24d.so` is used for DB2® connectivity. In most cases, the default selections are appropriate. However, the `AccessLibrary` property can be changed if the default value proves to be incorrect in your Unica Campaign environment. For example, 64-bit Unica Campaign provides two ODBC access libraries: one appropriate for ODBC data sources compatible with the unixODBC implementation (`libodb4d.so`) and the other compatible with the DataDirect implementation (`libodb4dDD.so`, used by Unica Campaign to access, for example, Teradata).

### AliasPrefix

#### Description

The `AliasPrefix` property specifies the way Unica Campaign forms the alias name that Unica Campaign creates automatically when using a dimension table and writing to a new table.

Note that each database has a maximum identifier length; check the documentation for the database you are using to be sure that the value you set does not exceed the maximum identifier length for your database.

#### Default value

A

### Additional libraries for AIX

#### Description

Unica Campaign includes two additional libraries for AIX ODBC driver managers that support the ODBC ANSI API rather than the ODBC Unicode API:

- `libodb4dAO.so` (32- and 64-bit): ANSI-only library for unixODBC-compatible implementations
- `libodb4dDDAO.so` (64-bit only): ANSI-only library for DataDirect-compatible implementations

If you determine that the default access library must be overridden, set this parameter (for example, to `libodb4dDD.so`, overriding the default selection of `libodb4d.so`).

#### Default value

No default value defined.

### AllowBaseJoinsInSelect

#### Description

This property determines whether Unica Campaign attempts to do a SQL join of base tables (from the same data source) used in a Select process; otherwise, the equivalent join is done on the Unica Campaign server.

#### Default value

TRUE

#### Valid Values

TRUE | FALSE

### AllowSegmentUsingSQLCase

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

This property specifies whether the Segment process consolidates multiple SQL statements into a single SQL statement, when specific configuration conditions are met.

Setting this property to `TRUE` results in significant performance improvements when all following conditions are met:

- Segments are mutually exclusive.
- All segments come from a single table.
- Criteria for each segment are based on the macro language.

In this case, Unica Campaign generates a single SQL `CASE` statement to perform segmentation, followed by segment-by-field processing on the Unica Campaign application server.

#### Default value

`TRUE`

#### Valid Values

`TRUE` | `FALSE`

## AllowTempTables

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

This property specifies whether Unica Campaign creates temporary tables in the database. Creating temporary tables can significantly improve the performance of campaigns.

When the value is `TRUE`, temporary tables are enabled. Each time a query is issued against the database (for example, by the Segment process), the resulting IDs are written to a temporary table in the database. When an additional query is issued, Unica Campaign can use that temporary table to retrieve rows from the database.

A number of Unica Campaign operations, such as `useInDbOptimization`, rely on the ability to create temp tables. If temporary tables are not enabled, Unica Campaign retains the selected IDs in the Unica Campaign server memory. The additional query retrieves IDs from the database and matches them to the IDs in server memory. This can negatively impact performance.


You must have appropriate privileges to write in the database to use temporary tables. Privileges are determined by the database login that you provide when you connect to the database.

#### Default value

`TRUE`



**Note:** Typically, you set **AllowTempTables** to `TRUE`. To override the value for a specific flowchart, open the

flowchart in Edit mode, select **Admin**  > **Advanced settings**, click the **Server optimization** tab, and select **Disallow use of temp tables for this flowchart**.

## ASMSaveDBAuthentication

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `ASMSaveDBAuthentication` property specifies whether, when you log in to Unica Campaign and map a table in a data source you did not previously log in to, Unica Campaign saves your user name and password in Unica.

If you set this property to `TRUE`, Unica Campaign does not prompt you for a user name and password when you log in to the data source. If you set this property to `FALSE`, Unica Campaign prompts you for a user name and password each time you log in to the data source.

### Default value

`TRUE`

### Valid Values

`TRUE | FALSE`

## ASMUserForDBCredentials

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `ASMUserForDBCredentials` property specifies the Unica user name that is assigned to the Unica Campaign system user (required to access the Unica Campaign system tables).

This property must be the same user that was created as the Unica Campaign system user during installation. This property is undefined by default.

### Default value

No default value defined.

## BulkInsertBlockSize

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

**Description**

This property defines the maximum size of a data block, in number of records, that Unica Campaign passes to the database at a time.

**Default value**

100

**BulkInsertRequiresColumnType****Configuration category**

*Campaign | partitions | partition[n] | dataSources | dataSourceName*

**Description**

The `BulkInsertRequiresColumnType` property is required to support DataDirect ODBC data sources only. Set this property to TRUE for DataDirect ODBC data sources when you use bulk (array) inserts. Set the property to FALSE to be compatible with most other ODBC drivers.

**Default value**

FALSE

**BulkReaderBlockSize****Configuration category**

*Campaign | partitions | partition[n] | dataSources | dataSourceName*

**Description**

The `BulkReaderBlockSize` property defines the size of a data block, in number of records, that Unica Campaign reads from the database at a time.

**Default value**

2500

**ConditionalSQLCloseBracket****Configuration category**

*Campaign | partitions | partition[n] | dataSources | dataSourceName*

**Description**

The `ConditionalSQLCloseBracket` property specifies the type of bracket that is used to indicate the end of a conditional segment in raw SQL custom macros. Conditionalized segments that are enclosed in the specified open and close bracket type are used only if temp tables exist. They are ignored if there are no temp tables.

**Default value**

} (closing curly brace)

## ConditionalSQLOpenBracket

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourceName*

### Description

The `ConditionalSQLOpenBracket` property specifies the type of bracket used to indicate the start of a conditional segment in raw SQL custom macros. Conditionalized segments enclosed within the brackets specified by the `ConditionalSQLOpenBracket` and `ConditionalSQLCloseBracket` properties are used only if temp tables exist, and are ignored if there are no temp tables.

### Default value

{ (opening curly brace)

## ConnectionCacheSize

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourceName*

### Description

The `ConnectionCacheSize` property specifies the number of connections that Unica Campaign maintains in a cache for each data source.

By default ( $N=0$ ), Unica Campaign establishes a new connection to a data source for each operation; if Unica Campaign maintains a cache of connections and a connection is available for reuse, Unica Campaign uses the cached connection rather than establishing a new connection.

If the setting is not 0, when a process is done with a connection, Unica Campaign keeps up to the specified number of connections open for an amount of time that is specified by the `InactiveConnectionTimeout` property. After this time expires, the connections are removed from the cache and closed.

### Default value

0 (zero)

## DateFormat

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourceName*

### Description

Unica Campaign uses the value of the `DateFormat` property to determine how to parse data in `date` formats when using the Unica Campaign macro language or when interpreting data from date columns.



Set the value of the `DateFormat` property to the format in which Unica Campaign expects to receive dates from this data source. The value must match the format that your database uses to display dates on select. For most databases, this setting is the same as the setting for the `DateOutputFormatString` property.



**Note:** If you use the multi-locale feature, do not use date formats that contain 3-letter months (MMM), %b (abbreviated month name), or %B (full month name). Instead, use a delimited or fixed format with a numeric value for the month.

To determine the date format that your database uses, select a date from the database. For additional information, see the following table.

**Table 50. Date formats**

Database	To determine the correct setting
DB2®	<p>Connect to the database from a machine that is running the Unica Campaign server. Use <code>db2test</code> in the <code>Campaign\bin</code> directory to connect and issue the following command:</p> <pre>values current date</pre> <p>If your operating system does not provide the <code>db2test</code> utility, use the <code>cxntest</code> utility to test connections to the target database.</p>
Hive-based Hadoop big data	<p>All Date strings (Date, DateFormat, DateTimeFormat, DateTimeOutputFormatString) must use the dash "-" character to format dates. Hive does not support any other characters for dates. Example: %Y-%m-%d %H:%M:%S</p>
Netezza®	<p>Connect to the database from a machine that is running the Unica Campaign server. Use <code>odbctest</code>, in the <code>Campaign\bin</code> directory, to connect and issue the following command:</p> <pre>CREATE TABLE date_test (f1 DATE); INSERT INTO date_test values (current_date); SELECT f1 FROM date_test;</pre> <p>Another way to select date format is to run following command:</p> <pre>SELECT current_date FROM ANY_TABLE limit 1;</pre> <p>where ANY_TABLE is the name of any existing table.</p>

**Table 50. Date formats (continued)**

Database	To determine the correct setting
Oracle	<p>Log in to the database from the machine that is running the Unica Campaign server. Use SQL *Plus to connect and issue the following command:</p> <pre>SELECT sysdate FROM dual</pre> <p>The current date is returned in NLS_DATE_FORMAT for that client.</p>
SQL Server	<p>Connect to the database from a machine that is running the Unica Campaign listener. Use <code>odbcetest</code>, in the <code>Campaign\bin</code> directory, to connect and issue the following command:</p> <pre>SELECT getdate()</pre> <p>If the <b>Use regional settings when outputting currency, numbers, dates, and times</b> option is not checked in the ODBC data source configuration, then you cannot reset the date format. In general, it is easier to leave this setting cleared so that the date format configuration does not change for each language.</p>
Teradata	<p>Teradata allows you to define the date format on a per-column basis. In addition to <code>dateFormat</code> and <code>dateOutputFormatString</code>, you must set <code>SuffixOnCreateDateField</code>. To be consistent with the system table settings, use:</p> <ul style="list-style-type: none"> <li>• <code>SuffixOnCreateDateField = FORMAT 'YYYY-MM-DD'</code></li> <li>• <code>DateFormat = DELIM_Y_M_D</code></li> <li>• <code>DateOutputFormatString = %Y-%m-%d</code></li> </ul>

**Default value**

DELIM\_Y\_M\_D

**Valid Values**Any of the formats that are specified in the `DATE` macro**DateOutputFormatString****Configuration category**

Campaign|partitions|partition[n]|dataSources|dataSourceName

**Description**

The `DateOutputFormatString` property specifies the format of the date datatype to be used when Unica Campaign writes any date, such as a campaign start or end date, to a database. Set the value of the `DateOutputFormatString` property to the format that the data source expects for columns of the type `date`. For most databases, this setting is the same as the setting for the `[data_source_name] > DateFormat` property.

The `DateOutputFormatString` property can be set to any of the formats that are specified for `format_string` in the `DATE_FORMAT` macro. The `DATE_FORMAT` macro accepts two different kinds of formats. One is an identifier (for example, `DELIM_M_D_Y`, `DDMMYYYY`, the same as accepted by the `DATE` macro), while the other is a format string. The value of the `DateOutputFormatString` property must be a format string - it must not be one of the `DATE` macro identifiers. Typically, use one of the delimited formats.

You can verify whether you selected the correct format by creating a table and inserting a date in the format you selected, as described in the following procedure.

#### To verify `DateOutputFormatString`

1. Connect to the database using the appropriate tool, as described in the table for "Selecting a date by database".

Do not use the query tools that come with the database (such as SQL Server's Query Analyzer) to verify that dates are being sent to the database correctly. These query tools might convert the date format to something other than what Unica Campaign actually sent to the database.

2. Create a table and insert a date in the format you selected. For example, if you selected `%m/%d/%Y`:

```
CREATE TABLE date_test (F1 DATE)
INSERT INTO date_test VALUES ('03/31/2004')
```

If the database allows the `INSERT` command to complete successfully, then you selected the correct format.

#### Default value

`%Y/%m/%d`

## DateTimeFormat

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

The value of the `<data-source-name> DateTimeFormat` property specifies the format in which Unica Campaign expects to receive `datetime`/`timestamp` data from a database. It must match the format that your database uses to display `datetime`/`timestamp` data on select. For most databases, this setting is the same as the setting for `DateTimeOutputFormatString`.

Typically, you set the `DateTimeFormat` by prepending your `DateFormat` value with `DT_` after determining the `DateFormat` value as described previously.



**Note:** If you use the multi-locale feature, do not use date formats that contain 3-letter months (MMM), %b (abbreviated month name), or %B (full month name). Instead, use a delimited or fixed format with a numeric value for the month.

### Default value

DT\_DELIM\_Y\_M\_D

### Valid Values

Only delimited formats are supported, as follows:

- DT\_DELIM\_M\_D
- DT\_DELIM\_M\_D\_Y
- DT\_DELIM\_Y\_M
- DT\_DELIM\_Y\_M\_D
- DT\_DELIM\_M\_Y
- DT\_DELIM\_D\_M
- DT\_DELIM\_D\_M\_Y

## DateTimeOutputFormatString

### Configuration category

`Campaign|partitions|partition[n]|dataSources|datasourcename`

### Description

The `DateTimeOutputFormatString` property specifies the format of the datetime datatype to be used when Unica Campaign writes any datetime, such as a campaign start or end date and time, to a database. Set the value of the `DateTimeOutputFormatString` property to the format that the data source expects for columns of the type `datetime`. For most databases, this setting is the same as the setting for the `[data_source_name] > DateTimeFormat` property.

See `DateOutputFormatString` for a method for verifying that the format you select is correct.

### Default value

`%Y/%m/%d %H:%M:%S`

## DB2NotLoggedInitially

### Configuration category

`Campaign|partitions|partition[n]|dataSources|datasourcename`

### Description

This property determines whether Unica Campaign uses the `not logged initially` SQL syntax when populating temporary tables in DB2®.

A value of `TRUE` disables logging for inserts into temp tables, which improves performance and decreases database resource consumption. When set to `TRUE`, if a temp table transaction fails for any reason, the table will become corrupted and must be dropped. All data previously contained in the table will be lost.

If your version of DB2® does not support the `not logged initially` syntax, set this property to `FALSE`.

If you are using a DB2® 11 user database on z/OS®, set this property to `FALSE`. If you are using DB2® 10.5 with the BLU feature ON for a user database, set both **DB2NotLoggedInitially** and **DB2NotLoggedInitiallyUserTables** to `FALSE`.

#### Default value

`TRUE`

#### Valid Values

`TRUE` | `FALSE`

## DB2NotLoggedInitiallyUserTables

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

The `DB2NotLoggedInitiallyUserTables` property determines whether Unica Campaign uses the `not logged initially` SQL syntax for inserts into DB2® user tables.

A value of `TRUE` disables logging for inserts into the user tables, which improves performance and decreases database resource consumption. When set to `TRUE`, if a user table transaction fails for any reason, the table will become corrupted and must be dropped. All data previously contained in the table will be lost.

If you are using DB2® 10.5 with the BLU feature ON for a user database, set both **DB2NotLoggedInitially** and **DB2NotLoggedInitiallyUserTables** to `FALSE`.



**Note:** The **DB2NotLoggedInitiallyUserTables** property is not used for the Unica Campaign system tables.

#### Default value

`FALSE`

#### Valid Values

TRUE | FALSE

## DefaultScale

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `DefaultScale` property is used when Unica Campaign creates a database field to store numeric values from a flat file or a derived field, when using the Snapshot or Export process.

This property is not used for numeric values originating in a database table, unless the database field omits information about precision and scale. (Precision indicates the total number of digits allowed for the field. Scale indicates the number of digits allowed to the right of the decimal point. For example, 6.789 has a precision of 4 and a scale of 3. Values obtained from a database table include information about precision and scale, which Unica Campaign uses when creating the field.)

Example: Flat files do not indicate precision and scale so you can use `DefaultScale` to specify how many places to the right of the decimal point to define for the field that is created, as shown below:

- `DefaultScale=0` creates a field with no places to the right of the decimal point (only whole numbers can be stored).
- `DefaultScale=5` creates a field with a maximum of 5 values to the right of the decimal point.

If the value set for `DefaultScale` exceeds the field's precision, `DefaultScale=0` is used for those fields. For example, if the precision is 5, and `DefaultScale=6`, a value of zero is used.

### Default value

0 (zero)

## DefaultTextType

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `DefaultTextType` property is intended for ODBC data sources. This property tells Unica Campaign how to create text fields in the destination data source if the source text fields are from a different data source type. For example, the source text fields might be from a flat file or from a different type of DBMS. If the source text fields are from the same type of DBMS, this property is ignored and the text fields are created in the destination data source using the data types from the source text fields.

### Default value

VARCHAR

**Valid Values**

VARCHAR | NVARCHAR

**DeleteAsRecreate****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

**Description**

The `DeleteAsRecreate` property specifies whether, when an output process is configured to `REPLACE TABLE` and if `TRUNCATE` is not supported, Unica Campaign drops and recreates the table or only deletes from the table.

When the value is `TRUE`, Unica Campaign drops the table and recreates it.

When the value is `FALSE`, Unica Campaign executes a `DELETE FROM` from the table.

**Default value**

`FALSE`

**Valid Values**

`TRUE` | `FALSE`

**DeleteAsTruncate****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

**Description**

The `DeleteAsTruncate` property specifies whether, when an output process is configured to `REPLACE TABLE`, Unica Campaign uses `TRUNCATE TABLE` or deletes from the table.

When the value is `TRUE`, Unica Campaign runs a `TRUNCATE TABLE` from the table.

When the value is `FALSE`, Unica Campaign runs a `DELETE FROM` from the table.

The default value depends on the database type.

**Default value**

- `TRUE` for Netezza®, Oracle, and SQLServer.
- `FALSE` for other database types.

**Valid Values**

`TRUE` | `FALSE`

## DisallowTempTableDirectCreate

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

This property is used by Oracle, Netezza®, and SQL Server data sources and is ignored for all other data sources.

This property specifies the way Unica Campaign adds data to a temp table.

When set to `FALSE`, Unica Campaign performs direct create-and-populate SQL syntax using one command. For example: `CREATE TABLE <table_name> AS ...` (for Oracle and Netezza) and `SELECT <field_names> INTO <table_name> ...` (for SQL Server).

When set to `TRUE`, Unica Campaign creates the temp table and then populates it directly from table to table using separate commands.

### Default value

`FALSE`

### Valid Values

`TRUE | FALSE`

## DSN

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Set this property to the data source name (DSN) as assigned in your ODBC configuration for this Unica Campaign data source. For SQL server, set this property to the DSN (data source name) that was created during installation. For Oracle and DB2®, set this property to the database name or the SID (service) name. This value is undefined by default.

Using the Unica Campaign data source configuration properties, you can specify multiple logical data sources that refer to the same physical data source. For example, you can create two sets of data source properties for the same data source, one with `AllowTempTables = TRUE` and the other with `AllowTempTables = FALSE`. Each of these data sources has a different name in Unica Campaign, but if they refer to the same physical data source they have the same DSN value.

### Default value

No default value defined.



## DSNUsingOSAuthentication

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `DSNUsingOSAuthentication` property applies only when an Unica Campaign data source is SQL Server. Set the value to `TRUE` when the DSN is configured to use Windows Authentication mode.

### Default value

`FALSE`

### Valid Values

`TRUE | FALSE`

## EnableBaseDimSelfJoin

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `EnableBaseDimSelfJoin` property specifies whether the Unica Campaign database behavior will perform self-joins when the Base and Dimension tables are mapped to the same physical table and the Dimension is not related to the Base table on the Base table's ID field(s).

By default, this property is set to `FALSE`, and when the Base and Dimension tables are the same database table and the relationship fields are the same (for example, `AcctID` to `AcctID`), Unica Campaign assumes that you do not want to perform a join.

### Default value

`FALSE`

## EnableSelectDistinct

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `EnableSelectDistinct` property specifies whether the internal lists of IDs for Unica Campaign are de-duplicated by the Unica Campaign server or by the database.

When the value is `TRUE`, the database performs de-duplication, and SQL queries generated against the database then have the form (when appropriate):

```
SELECT DISTINCT key FROM table
```

When the value is `FALSE`, the Unica Campaign server performs de-duplication, and SQL queries generated against the database have the form:

```
SELECT key FROM table
```

Leave the default value of `FALSE` if:

- Your database is constructed so that unique identifiers (primary keys of base tables) are already guaranteed to be de-duped.
- You want the Unica Campaign application server to perform de-duplication to reduce resource consumption/burden on the database.

Regardless of what value you specify for this property, Unica Campaign automatically ensures that keys are de-duplicated as required. This property merely controls where the de-duplication effort occurs (on the database or on the Unica Campaign server).

#### Default value

`TRUE`

#### Valid Values

`TRUE` | `FALSE`

## EnableSelectOrderBy

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourceName`

#### Description

The `EnableSelectOrderBy` property specifies whether the internal lists of IDs for Unica Campaign are sorted by the Unica Campaign server or by the database.

When the value is `TRUE`, the database performs the sorting, and SQL queries generated against the database have the form:

```
SELECT <key> FROM <table> ORDER BY <key>
```

When the value is `FALSE`, the Unica Campaign server performs the sorting, and SQL queries generated against the database have the form:

```
SELECT <key> FROM <table>
```



**Note:** Only set this property to `FALSE` if the audience levels used are text strings on a non-English database. All other scenarios can use the default of `TRUE`.

#### Default value

TRUE

**Valid Values**

True | False

**ExcludeFromTableDisplay****Configuration category**`Campaign|partitions|partition[n]|dataSources|dataSourcename`**Description**

The `ExcludeFromTableDisplay` parameter allows you to limit the database tables that are displayed during table mapping in Unica Campaign. It does not reduce the number of table names retrieved from the database. Table names that match the specified patterns are not displayed. Values for this parameter are case-sensitive.

Example: If you set the value to `sys.*`, tables with names that begin with all lower case `sys.` are not displayed.

Example: `UAC_*` (the default value for SQL Server data sources) excludes temp tables and Extract tables, when the `ExtractTablePrefix` property's value is the default value.

Example: To exclude the Unica Platform system tables, as they are not relevant when working with user data:

```
DF_*,USM_*,OLS_*,QRTZ*,USCH_*,UAR_*
```

Using Oracle as an example, the complete value would be:

```
UAC_*,PUBLIC.*,SYS.*,SYSTEM.*,DF_*,USM_*,OLS_*,QRTZ*,USCH_*,UAR_*
```

**Default value**

```
UAC_*,PUBLIC.*,SYS.*,SYSTEM.* (for an Oracle data source)
```

```
UAC_* (for a SQL Server data source)
```

```
UAC_*,SYSCAT.*,SYSIBM.*,SYSSTAT.* (for a DB2® data source)
```

**ExtractTablePostExecutionSQL****Configuration category**`Campaign|partitions|partition[n]|dataSources|dataSourcename`**Description**

Use the `ExtractTablePostExecutionSQL` property to specify one or more complete SQL statements that run immediately after the creation and population of an Extract table.

Tokens available to `ExtractTablePostExecutionSQL` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Extract table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Extract table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Extract table was created.
<DBUSER>	This token is replaced with the database user name for the database where the Extract table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Extract table creation.
<KEYCOLUMNS>	This token is replaced with the Extract table column name(s).
<TABLENAME>	This token is replaced with the Extract table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

#### Default value

Not defined

#### Valid Values

A valid SQL statement

### ExtractTablePrefix

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourceName`

#### Description

The `ExtractTablePrefix` property specifies a string that is automatically prepended to all Extract table names in Unica Campaign. This property is useful when two or more data sources point to the same database. For details, see the `TempTablePrefix` description.

**Default value**

UAC\_EX

**ForceNumeric****Configuration category**`Campaign|partitions|partition[n]|dataSources|dataSourcename`**Description**

The `ForceNumeric` property specifies whether Unica Campaign retrieves numeric values as the data type `double`. When the value is set to `TRUE`, Unica Campaign retrieves all numeric values as the data type `double`.

**Default value**

FALSE

**Valid Values**

TRUE | FALSE

**HiveQueryMode****Configuration category**`Campaign|partitions|partition[n]|dataSources|dataSourcename`**Description**

This property is used only for Hive-based Hadoop data sources (`BigDataODBCHiveTemplate`). It provides a way to switch between the `DataDirect` and `Cloudera` drivers. For `DataDirect`, select `Native`. For `Cloudera`, select `SQL`.

**Valid Values**

Native | SQL

**InactiveConnectionTimeout****Configuration category**`Campaign|partitions|partition[n]|dataSources|dataSourcename`**Description**

The `InactiveConnectionTimeout` property specifies the number of seconds an inactive Unica Campaign database connection is left open before it is closed. Setting the value to `0` disables the timeout, leaving the connection open.

**Default value**

120

## InsertLogSize

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `InsertLogSize` property specifies when a new entry is entered in the log file while the Unica Campaign Snapshot process is running. Every time the number of records written by the Snapshot process reaches a multiple of the number specified in the `InsertLogSize` property, a log entry is written. The log entries can help you determine how far a running Snapshot process has progressed. Setting this value too low may create large log files.

### Default value

100000 (one hundred thousand records)

### Valid Values

Positive integers

## JndiName

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `JndiName` property is used only when configuring the Unica Campaign system tables (not for user data sources). Set its value to the Java Naming and Directory Interface (JNDI) data source that you created in your application server (WebSphere® or WebLogic) to connect to this data source.

### Default value

campaignPartitionLDS

## LoaderCommand

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

This property specifies the command issued to invoke your database load utility in Unica Campaign. If you set this property, Unica Campaign enters the database loader utility mode for all output files from the Snapshot process that are used with the **Replace all records** settings. This property also invokes the database loader utility mode when Unica Campaign uploads ID lists into temp tables.

The valid value for this property is any full path name either to the database load utility executable or to a script that launches the database load utility. Using a script allows you to perform additional setup before invoking the load utility.



**Note:** If you use Unica Optimize and you are configuring loader settings for the UA\_SYSTEM\_TABLES datasource, there are important considerations to take into account. For example, you must use absolute paths for **LoaderCommand** and **LoaderCommandForAppend**. Read about setting up Unica Campaign to use database load utilities in the *Unica Campaign Administrator's Guide*.

Most database load utilities require several arguments to be launched successfully. These arguments can include specifying the data file and control file to load from and the database and table to load into. Unica Campaign supports the following tokens, which are replaced by the specified elements when the command is run. Consult your database load utility documentation for the correct syntax to use when invoking your database load utility.

This property is undefined by default.

Tokens available to **LoaderCommand** are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart being run.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart being run.
<CONTROLFILE>	This token is replaced with the full path and file name to the temporary control file that Unica Campaign generates according to the template that is specified in the <b>LoaderControlFileTemplate</b> property.
<DATABASE>	This token is replaced with the name of the data source that Unica Campaign is loading data into. This is the same data source name used in the category name for this data source.
<DATAFILE>	This token is replaced with the full path and file name to the temporary data file created by Unica Campaign during the loading process. This file is in the Unica Campaign Temp directory, UNICA_ACTMPDIR.
<DBUSER>	This token is replaced with the database user name for the database.

Token	Description
<DSN>	This token is replaced with the value of the DSN property. If the <code>DSN</code> property is not set, the <DSN> token is replaced by the data source name used in the category name for this data source (the same value used to replace the <DATABASE> token).
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart being run.
<NUMFIELDS>	This token is replaced with the number of fields in the table.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<TABLE>	This token is obsolete. Use <TABLENAME> instead.
<TABLENAME>	This token is replaced with the database table name that Unica Campaign is loading data into. This is the target table from your Snapshot process or the name of the Temp Table being created by Unica Campaign.
<USER>	This token is replaced with the database user from the current flowchart connection to the data source.

**Default value**

No default value defined.

**Valid Values**

Any full path name either to the database load utility executable or to a script that launches the database load utility.

**LoaderCommandForAppend****Configuration category**

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

**Description**

This property specifies the command issued to invoke your database load utility for appending records to a database table in Unica Campaign. If you set this property, Unica Campaign enters database



loader utility mode for all output files from the Snapshot process that are used with the **Append Records** settings.

This property is specified as a full path name either to the database load utility executable or to a script that launches the database load utility. Using a script allows you to perform additional setup before invoking the load utility.

Most database load utilities require several arguments to be successfully launched. These can include specifying the data file and control file to load from and the database and table to load into. The tokens are replaced by the specified elements when the command is run.

Consult your database load utility documentation for the correct syntax to use when invoking your database load utility.

This property is undefined by default.

Tokens available to **LoaderCommandForAppend** are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart being run.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart being run.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart being run.
<CONTROLFILE>	This token is replaced with the full path and file name to the temporary control file that Unica Campaign generates according to the template that is specified in the <b>LoaderControlFileTemplate</b> property.
<DATABASE>	This token is replaced with the name of the data source that Unica Campaign is loading data into. This is the same data source name used in the category name for this data source.
<DATAFILE>	This token is replaced with the full path and file name to the temporary data file created by Unica Campaign during the loading process. This file is in the Unica Campaign Temp directory, UNICA_-\ACTMPDIR.

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<DSN>	This token is replaced with the value of the DSN property. If the <code>DSN</code> property is not set, the <code>&lt;DSN&gt;</code> token is replaced by the data source name used in the category name for this data source (the same value used to replace the <code>&lt;DATABASE&gt;</code> token).
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<NUMFIELDS>	This token is replaced with the number of fields in the table.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<TABLE>	This token is obsolete. Use <code>&lt;TABLENAME&gt;</code> instead.
<TABLENAME>	This token is replaced with the database table name that Unica Campaign is loading data into. This is the target table from your Snapshot process or the name of the Temp Table being created by Unica Campaign.
<USER>	This token is replaced with the database user from the current flowchart connection to the data source.

**Default value**

No default value defined.

**LoaderControlFileTemplate****Configuration category**

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

**Description**

This property specifies the full path and file name to the control file template that is configured for Unica Campaign. The path to the template is relative to the current partition. For example:

`loadscript.db2.`

When this property is set, Unica Campaign dynamically builds a temporary control file based on the specified template. The path and name of this temporary control file is available to the `<CONTROLFILE>` token that is available to the **LoaderCommand** property.

Before you use Unica Campaign in the database loader utility mode, you must configure the control file template that is specified by this parameter. The control file template supports the following tokens, which are dynamically replaced when the temporary control file is created by Unica Campaign.

For the correct syntax required for your control file, see your database loader utility documentation.

This property is undefined by default.

Tokens available to **LoaderControlFileTemplate** are the same as those described for the **LoaderCommand** property, plus the following special tokens, which are repeated once for each field in the outbound table.

Token	Description
<code>&lt;DBCOLUMNNUMBER&gt;</code>	This token is replaced with the column ordinal in the database.
<code>&lt;FIELDLENGTH&gt;</code>	This token is replaced with the length of the field being loaded into the database.
<code>&lt;FIELDNAME&gt;</code>	This token is replaced with the name of the field being loaded into the database.
<code>&lt;FIELDNUMBER&gt;</code>	This token is replaced with the number of the field being loaded into the database.
<code>&lt;FIELDTYPE&gt;</code>	This token is replaced with the literal <code>CHAR ( )</code> . The length of this field is specified between the parentheses <code>( )</code> . If your database does not understand the field type <code>CHAR</code> , you can manually specify the appropriate text for the field type and use the <code>&lt;FIELDLENGTH&gt;</code> token. For example, for <code>SQLSVR</code> and <code>SQL2000</code> you would use <code>SQLCHAR (&lt;FIELDLENGTH&gt;)</code> .
<code>&lt;NATIVETYPE&gt;</code>	This token is replaced with the actual database type that this field is loaded into.
<code>&lt;xyz&gt;</code>	This token places the specified character(s) on all fields being loaded into the database, except the last. A typical use is <code>&lt;,&gt;</code> which repeats a comma for all fields except the last.
<code>&lt;-xyz&gt;</code>	This token places the specified characters only on the last repeated line.
<code>&lt;!xyz&gt;</code>	This token places the specified character(s), including the angle brackets <code>&lt; &gt;</code> , on all lines.

### Default value

No default value defined.

## LoaderControlFileTemplateForAppend

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

### Description

This property specifies the full path and file name to the control file template that is configured in Unica Campaign. The path to the template is relative to the current partition. For example: `loadappend.db2`

When this property is set, Unica Campaign dynamically builds a temporary control file based on the specified template. The path and name of this temporary control file is available to the `<CONTROLFILE>` token that is available to the **LoaderCommandForAppend** property.

Before you use Unica Campaign in the database loader utility mode, you must configure the control file template that is specified by this property. See your database loader utility documentation for the correct syntax required for your control file.

The available tokens are the same as the tokens for the **LoaderControlFileTemplate** property.

This property is undefined by default.

### Default value

No default value defined.

## LoaderDelimiter

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

### Description

This property specifies whether the temporary data file is a fixed-width or delimited flat file, and, if it is delimited, the characters that Unica Campaign uses as delimiters.


If the value is undefined, Unica Campaign creates the temporary data file as a fixed width flat file.

If you specify a value, it is used when the loader is invoked to populate a table that is known to be empty. Unica Campaign creates the temporary data file as a delimited flat file, using the value of this property as the delimiter. The delimiter is a character such as comma (,) or semi-colon (;) that separates fields in the temporary data files that are loaded into the user data source.



**Important:** The following fields, if used, must use the same character as specified for **LoaderDelimiter**: **SuffixOnTempTableCreation**, **SuffixOnSegmentTableCreation**,

 **SuffixOnSnapshotTableCreation, SuffixOnExtractTableCreation, SuffixOnUserBaseTableCreation, SuffixOnUserTableCreation.**

 **Important:** For big data, such as Hadoop Hive or Amazon Redshift, the delimiter value must match the ROW format delimiter that was used when the big data database table was created. In this example, a comma is used: **ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';**

This property is undefined by default.

#### Default value

No default value defined.

#### Valid Values

Characters, which can be enclosed in double quotation marks, if wanted. Hive-based Hadoop big data does not support the Tab (/t) character.

### LoaderDelimiterAtEnd

#### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

#### Description

Some external load utilities require that the data file be delimited and that each line end with the delimiter. To accommodate this requirement, set the **LoaderDelimiterAtEnd** value to `TRUE`, so that when the loader is invoked to populate a table that is known to be empty, Unica Campaign uses delimiters at the end of each line. For example, DB2® on Unix expects each record to be terminated by a line feed character only; Unica Campaign Unica Campaign on Windows uses carriage return and line feed characters. Putting a delimiter at the end of every record ensures that the last column in the data file will load properly.

`FALSE`

#### Default value

`FALSE`

#### Valid Values

`TRUE | FALSE`

### LoaderDelimiterAtEndForAppend

#### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

#### Description

Some external load utilities require that the data file be delimited and that each line end with the delimiter. To accommodate this requirement, set the **LoaderDelimiterAtEndForAppend** value to `TRUE`, so that when the loader is invoked to populate a table that is not known to be empty, Unica Campaign uses delimiters at the end of each line. For example, DB2® on Unix expects each record to be terminated by a line feed character only; Unica Campaign on Windows uses carriage return and line feed characters. Putting a delimiter at the end of every record ensures that the last column in the data file will load properly.

#### Default value

`FALSE`

#### Valid Values

`TRUE` | `FALSE`

## LoaderDelimiterForAppend

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourceName`

#### Description

This property specifies whether the temporary Unica Campaign data file is a fixed-width or delimited flat file, and, if it is delimited, the character or set of characters used as delimiters.

If the value is undefined, Unica Campaign creates the temporary data file as a fixed width flat file.

If you specify a value, it is used when the loader is invoked to populate a table that is not known to be empty. Unica Campaign creates the temporary data file as a delimited flat file, using the value of this property as the delimiter.

This property is undefined by default.

#### Default value

No default value defined.

#### Valid Values

Characters, which you may enclose in double quotation marks, if wanted.

## LoaderPostLoadDataFileRemoveCmd

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourceName`

#### Description

This property is used only for Hive-based Hadoop data sources (BigDataODBC HiveTemplate).

This property is used together with LoaderPreLoadDataFileCopyCmd. After data files are

copied from Unica Campaign to the `/tmp` folder on the Hive-based Hadoop system, the `LoaderPostLoadDataFileRemoveCmd` property uses the SSH "rm" command to remove the temporary data file.

For example: `ssh mapr@example.com "rm/tmp/<DATAFILE>"`

For important information, read about exporting data from Unica Campaign to a Hive-based Hadoop system.

#### Default value

none

### LoaderPreLoadDataFileCopyCmd

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

This property is used only for Hive-based Hadoop data sources (BigDataODBCHiveTemplate). This property uses SCP to copy data from Unica Campaign to a temp folder called `/tmp` on your Hive-based Hadoop system. The location must be called `/tmp` and it must be on the Hive server (the file system location, not the HDFS location). You can either specify an SCP command or call a script that specifies the SCP command.

**Example #1:** `scp <DATAFILE> mapr@example.com:/tmp`

**Example #2:** `/opt/HCL/CampaignBigData/bin/copyToHive.sh <DATAFILE>`

In addition to this property, use `LoaderPostLoadDataFileRemove` to remove the temporary data file from the Hive server after it has been copied.

For important information, read about exporting data from Unica Campaign to a Hive-based Hadoop system.

#### Default value

none

### LoaderNULLValueInDelimitedData

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

This property supports null values in delimited data for database loaders, specifically Netezza®. Enter the string that represents a null value for the column.

#### Default value

null

## LoaderUseLocaleDP

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

This property specifies, when Unica Campaign writes numeric values to files to be loaded by a database load utility, whether the locale-specific symbol is used for the decimal point.

Set this value to `FALSE` to specify that the period (.) is used as the decimal point.

Set this value to `TRUE` to specify that the decimal point symbol appropriate to the locale is used.

### Default value

`FALSE`

### Valid Values

`TRUE` | `FALSE`

## MaxItemsInList

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Allows you to specify the maximum number of items that Unica Campaign is allowed to include in a single list in SQL (for example, the list of values following an IN operator in a WHERE clause).

### Default value

1000 (Oracle only), 0 (unlimited) for all other databases

### Valid Values

integers

## MaxQueryThreads

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

This property specifies the upper limit on the number of simultaneous queries allowed to run against each database source from a single Unica Campaign flowchart. Higher values generally improve performance.



Unica Campaign runs database queries using independent threads. Because Unica Campaign processes run in parallel, it is common to have multiple queries running simultaneously against a single data source. If the number of queries to be run in parallel exceeds the `MaxQueryThreads`, the Unica Campaign server limits the number of simultaneous queries to the specified value.

The maximum value is unlimited.



**Note:** If `maxReuseThreads` is set to a non-zero value, it should be greater than or equal to the value of `MaxQueryThreads`.

#### Default value

Varies depending on the database

### MaxRowFetchRecords

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

For performance reasons, it is best to keep this number low.

When the selected number of IDs is less than the value specified by the `MaxRowFetchRecords` property, Unica Campaign passes the IDs to the database one at a time, in separate SQL queries. This process can be very time-consuming. If the number of selected IDs is greater than the value specified by this property, Unica Campaign uses temporary tables (if allowed on the database source), or it pulls down all the values from the table, not including any unnecessary values.

#### Default value

100

### MaxTempTableJoinPctSelectAll

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

When a query is issued, Unica Campaign creates a temporary table on the database containing the exact list of IDs, as a result of the query. When an additional query that selects all records is issued against the database, the `MaxTempTableJoinPctSelectAll` property specifies whether a join is performed with the temporary table.

If the relative size of the temporary table (specified as a percentage) is greater than the value of the `MaxTempTableJoinPctSelectAll` property, no join is performed. All records are selected first, then unwanted records are discarded.

If the relative size of the temporary table (specified as a percentage) is less than or equal to the value of `MaxTempTableJoinPctSelectAll` property, the join is performed with the temporary table first, and then the resulting IDs are retrieved to the server.

This property is applicable only if the value of the `AllowTempTables` property is set to `TRUE`. This property is ignored if the `useInDbOptimization` property is set to `YES`.

### Default value

90

### Valid Values

Integers between 0-100. A value of 0 means that temporary table joins are never used; a value of 100 means that table joins are always used, regardless of the size of the temporary table.

### Example

Assume that `MaxTempTableJoinPctSelectAll` is set to 90. First, you might want to select customers (`CustID`) with account balances (`Accnt_balance`) greater than \$1,000 from the database table (`Customer`).

The corresponding SQL expression generated by the Select process may look like this:

```
SELECT CustID FROM Customer
WHERE Accnt_balance > 1000
```

The Select process may retrieve 100,000 IDs from the total table size of 1,000,000, which is 10%. If temporary tables are allowed, Unica Campaign writes the selected IDs (`TempID`) into a temporary table (`Temp_table`) in the database.

Then, you might want to snapshot the selected IDs (`CustID`) together with the actual balance (`Accnt_balance`). Since the relative size of the temporary table (`Temp_table`) is less than 90 percent (`MaxTempTableJoinPctSelectAll`), the join is done with the temporary table first. The SQL expression generated by the Snapshot process may look like this:

```
SELECT CustID, Accnt_balance FROM Customer, Temp_table WHERE CustID = TempID
```

If the Select process retrieves more than 90 percent, the subsequent Snapshot process retrieves all the records, and matches them with the first set of IDs, discarding the unnecessary ones.

The SQL expression generated by the Snapshot process may look like this:

```
SELECT CustID, Accnt_balance FROM Customer
```

## MaxTempTableJoinPctWithCondition

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

When a query is issued, Unica Campaign creates a temporary table on the database containing the exact list of IDs, as a result of the query. When an additional query, selecting records with limitation conditions is issued against the database, the `MaxTempTableJoinPctWithCondition` property specifies whether a join should be performed with the temporary table.

If the relative size of the temporary table (specified as a percentage) is greater than the value of `MaxTempTableJoinPctWithCondition`, no join is performed. This avoids the overhead in the database where it may not be needed. In this case, the query is issued against the database, the resulting list of IDs retrieved, and then unwanted records are discarded as they are matched to the list in server memory.

If the relative size of the temporary table (in percentage) is less than or equal to the value of `MaxTempTableJoinPctWithCondition`, the join is done with the temporary table first, and then the resulting IDs are retrieved to the server.

This property is applicable only if the value of the `AllowTempTables` property is set to `TRUE`.

#### Default value

20

#### Valid Values

Integers between 0-100. A value of 0 means that temporary table joins are never used; a value of 100 means that table joins are always used, regardless of the size of the temporary table.

## MinReqForLoaderCommand

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourceName`

#### Description

Use this property to set the threshold for using the bulk loader. Unica Campaign invokes the script assigned to the `LoaderCommand` property when the number of unique IDs in the input cell exceeds the value defined here. The value of this property does not represent the number of records that will be written.

If this property is not configured, Unica Campaign assumes that the value is the default value (zero). If this property is configured but a negative value or non-integer value is set as the value, a value of zero is assumed.

#### Default value

0 (zero)

#### Valid Values

Integers

## MinReqForLoaderCommandForAppend

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use this property to set the threshold for using the bulk loader. Unica Campaign invokes the script assigned to the `LoaderCommandForAppend` parameter when the number of unique IDs in the input cell exceeds the value defined here. The value of this property does not represent the number of records that will be written.

If this property is not configured, Unica Campaign assumes that the value is the default value (zero). If this property is configured but a negative value or non-integer value is set as the value, a value of zero is assumed.

### Default value

0 (zero)

### Valid Values

Positive integers

## NumberOfRetries

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `NumberOfRetries` property specifies the number of times Unica Campaign automatically retries a database operation on failure. Unica Campaign automatically resubmits queries to the database this number of times before reporting a database error or failure.

### Default value

0 (zero)

## ODBCTableTypes

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

This property is empty by default, which is appropriate for all currently supported data sources.

### Default value

Not defined

### Valid Values

(empty)

## ODBCUnicode

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `ODBCUnicode` property specifies the type of encoding used in Unica Campaign ODBC calls. It is used only with ODBC data sources and is ignored when used with Oracle or DB2® native connectivity.



**Important:** If this property is set to `UTF-8` or `UCS-2`, the data source's `StringEncoding` value must be set to either `UTF-8` or `WIDEUTF-8`, otherwise the `ODBCUnicode` property's setting is ignored.

### Default value

`disabled`

### Valid Values

Possible values for this property are:

- `Disabled`: Unica Campaign uses ANSI ODBC calls.
- `UTF-8`: Unica Campaign uses Unicode ODBC calls and assumes that a `SQLWCHAR` is a single byte. This is compatible with DataDirect ODBC drivers.
- `UCS-2`: Unica Campaign uses Unicode ODBC calls and assumes that a `SQLWCHAR` is 2 bytes. This is compatible with Windows™ and unixODBC ODBC drivers.

## ODBCv2

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use the `ODBCv2` property to specify which ODBC API specification Unica Campaign should use for the data source.

The default value of `FALSE` allows Unica Campaign to use the v3 API specification, while a setting of `TRUE` causes Unica Campaign to use the v2 API specification. Set the `ODBCv2` property to `TRUE` for data sources that do not support the ODBC v3 API specification.

When the `ODBCv2` property is set to `TRUE`, Unica Campaign does not support the ODBC Unicode API, and values other than `disabled` for the `ODBCUnicode` property are not recognized.

### Default value

`FALSE`

### Valid Values

TRUE | FALSE

## OwnerForTableDisplay

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use this property to limit the table mapping display in Unica Campaign to tables in a specified schema. For example, to specify tables in the schema "dbo", set OwnerForTableDisplay=dbo.

### Default value

No default value defined.

## PadTextWithSpaces

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

When set to TRUE, the PadTextWithSpaces property causes Unica Campaign to pad text values with spaces until the string is the same width as the database field.

### Default value

FALSE

### Valid Values

TRUE | FALSE

## PostExtractTableCreateRunScript

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use this property to specify a script or executable for Unica Campaign to run after an Extract table has been created and populated.

Tokens available to **PostExtractTableCreateRunScript** are described below.

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the Extract table was created.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Extract table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Extract table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Extract table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Extract table creation.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<KEYCOLUMNS>	This token is replaced with the Extract table column name(s).

**Default value**

Not defined

**Valid Values**

File name of a shell script or executable

**PostSegmentTableCreateRunScript****Configuration category***Campaign|partitions|partition[n]|dataSources|dataSourcename***Description**

Specifies a script or executable that Unica Campaign runs after a Segment temp table has been created and populated.

Tokens available to **PostSegmentTableCreateRunScript** are described below.

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the Segment temp table was created.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Segment temp table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Segment temp table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Segment temp table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Segment temp table creation.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<KEYCOLUMNS>	This token is replaced with the Segment temp table column name(s).

**Default value**

Not defined

**Valid Values**

File name of a script or executable

**PostSnapshotTableCreateRunScript****Configuration category***Campaign|partitions|partition[n]|dataSources|dataSourceName***Description**

Use the **PostSnapshotTableCreateRunScript** property to specify a script or executable that Unica Campaign runs after a Snapshot table has been created and populated. The property is only called when snapshot process writes to "Extract Tables"

Tokens available to **PostSnapshotTableCreateRunScript** are described below.



Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the Snapshot table was created.
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Snapshot table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Snapshot table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Snapshot table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Snapshot table creation.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<KEYCOLUMNS>	This token is replaced with the Snapshot table column name(s).

**Default value**

Not defined

**Valid Values**

File name of a shell script or executable

**PostTempTableCreateRunScript****Configuration category***Campaign|partitions|partition[n]|dataSources|dataSourcename***Description**

Use the **PostTempTableCreateRunScript** property to specify a script or executable for Unica Campaign to run after a temp table has been created and populated in a user data source or in the system tables database.

Tokens available to **PostTempTableCreateRunScript** are described below.

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<KEYCOLUMNS>	This token is replaced with the temp table column name(s).

**Default value**

No default value defined.

**PostUserTableCreateRunScript****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

**Description**

Specifies a script or executable that Unica Campaign runs after a User table has been created and populated.

Tokens available to `PostUserTableCreateRunScript` are described below.

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the User table was created.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the User table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the User table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the User table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the User table creation.
<PASSWORD>	This token is replaced with the database password from the current flowchart connection to the data source.
<KEYCOLUMNS>	This token is replaced with the User table column name(s).

**Default value**

Not defined

**Valid Values**

File name of a script or executable

**PrefixOnSelectSQL****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

**Description**

Use the `PrefixOnSelectSQL` property to specify a string that is automatically prepended to all `SELECT` SQL expressions generated by Unica Campaign.

This property applies only to SQL generated by Unica Campaign, and does not apply to SQL in raw SQL expressions used in the Select process.

This property is automatically added to the `SELECT` SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression.

This property is undefined by default.

Tokens available to **PrefixOnSelectSQL** are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

## QueryThreadSleep

#### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

#### Description

The `QueryThreadSleep` property affects the CPU utilization of the Unica Campaign server process (`UNICA_ACSVR`). When the value is `TRUE`, the thread that the Unica Campaign server process uses to check for query completion sleeps between checks. When the value is `FALSE`, the Unica Campaign server process checks continuously for query completion.

#### Default value

`TRUE`

## ReaderLogSize

#### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

**Description**

The `ReaderLogSize` parameter defines when Unica Campaign makes a new entry in the log file when reading data from the database. Every time the number of records read from the database reaches a multiple of the number defined by this parameter, a log entry is written in the log file.

This parameter can help you determine how far a process has progressed in its run. Setting this value too low may create large log files.

**Default value**

1000000 (one million records)

**Valid Values**

Integers

**SegmentTablePostExecutionSQL****Configuration category**

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

**Description**

Use the `SegmentTablePostExecutionSQL` property to specify a complete SQL statement that Unica Campaign runs after a Segment temp table has been created and populated.

Tokens available to `SegmentTablePostExecutionSQL` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Segment temp table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Segment temp table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Segment temp table was created.
<DBUSER>	This token is replaced with the database user name for the database where the Segment temp table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Segment temp table creation.

Token	Description
<KEYCOLUMNS>	This token is replaced with the Segment temp table column name(s).
<TABLENAME>	This token is replaced with the Segment temp table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

**Default value**

Not defined

**Valid Values**

A valid SQL statement

**SegmentTempTablePrefix****Description**

Sets the prefix for Segment tables created by the CreateSeg process in this data source. This property is useful when two or more data sources point to the same database. For details, see the `TempTablePrefix` description.

**Default value**

UACS

**SnapshotTablePostExecutionSQL****Configuration category**

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

**Description**

Use the `SnapshotTablePostExecutionSQL` property to specify one or more complete SQL statements to run immediately after a Snapshot table has been created and populated. This property is invoked only when a Snapshot process box writes out to an extract table.

Tokens available to `SnapshotTablePostExecutionSQL` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Snapshot table was created.

Token	Description
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Snapshot table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Snapshot table was created.
<DBUSER>	This token is replaced with the database user name for the database where the Snapshot table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Snapshot table creation.
<KEYCOLUMNS>	This token is replaced with the Snapshot table column name(s).
<TABLENAME>	This token is replaced with the Snapshot table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

**Default value**

Not defined

**Valid Values**

A valid SQL statement

**SQLOnConnect****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

**Description**

The `SQLOnConnect` property defines a complete SQL statement that Unica Campaign runs immediately after each database connection.

The SQL statement generated by this property is automatically passed to your database without checking its syntax. If you use this property, make sure that it is a legal expression. The string may be enclosed in quotation marks, but this is not required.

This property is undefined by default.

Tokens available to `SQLOnConnect` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.
<session_token::x>	<p>Users should specify any additional header variable names under Affinium suite security login-Modes siteMinderPartitionLogin instead of using a generic placeholder like "x variable". This token gets replaced with the value of the additional header variable you have specified.</p> <p>Example:</p> <p>Insert into sample_table values ('&lt;session_token::x-auth-channel&gt;','&lt;session_token::x-auth-location&gt;')</p>

#### Default value

No default value defined.

#### StringEncoding

##### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

##### Description



The `StringEncoding` property specifies the character encoding of the database. When Unica Campaign retrieves data from the database, the data is transcoded from the encoding specified to the internal encoding of Unica Campaign (`UTF-8`). When Unica Campaign sends a query to the database, character data is transcoded from the internal encoding of Unica Campaign (`UTF-8`) to the encoding specified in the `StringEncoding` property.

The value of this property must match the encoding used on the database client.

Do not leave this value blank although it is undefined by default.

If you use ASCII data, set this value to `UTF-8`.

If your database client encoding is `UTF-8`, the preferred setting for this value is `WIDEUTF-8`. The `WIDE-UTF-8` setting works only if your database client is set to `UTF-8`.

If you use the `partitions > partition[n] > dataSources > data_source_name > ODBCUnicode` property, set the `StringEncoding` property to either `UTF-8` or `WIDEUTF-8`. Otherwise, the `ODBCUnicode` property value is ignored.

For a list of supported encodings, see Character encodings in Unica Campaign in the Unica Campaign Administrator's Guide.



**Important:** See the following sections for important exceptions and additional considerations.

#### Default value

No default value defined.

#### Database-specific considerations

This section describes how to set the correct values for DB2®, SQL Server, or Teradata databases.

##### DB2®

Identify the DB2® database code page and code set. For localized environments, the DB2® database must have the following configuration:

- Database code set = UTF-8
- Database code page = 1208

Set the `StringEncoding` property values in Unica Campaign to the DB2® database code set value.

Set the `DB2CODEPAGE` DB2® environment variable to the DB2® database code page value:

- On Windows™: Add the following line to the Unica Campaign Listener startup script (`<CAMPAIGN_HOME>\bin\cmpServer.bat`):

```
db2set DB2CODEPAGE=1208
```

- On UNIX™: After DB2® is started, the system administrator must type the following command from the DB2® instance user:

```
$ db2set DB2CODEPAGE=1208
```

Then start the Unica Campaign listener by running this command:

```
./rc.unica_ac start
```

This setting affects all DB2® data sources and can affect other running programs.

### SQL Server

For SQL Server, use a code page instead of an iconv encoding. To determine the correct the value for the `StringEncoding` property with a SQL Server database, look up the code page that corresponds to the regional settings of the server's operating system.

For example, to use code page 932 (Japanese `shift-JIS`):

```
StringEncoding=CP932
```

### Teradata

For Teradata, you must override some default behavior. Teradata supports per-column character encoding, while Unica Campaign supports only per-data source encoding. `UTF-8` cannot be used with Unica Campaign due to a bug in the Teradata ODBC driver. Teradata sets a default character encoding for each login. You can override this using a parameter in the ODBC data source configuration on Windows™ or in the `odbc.ini` on UNIX™ platforms as follows:

```
CharacterSet=UTF8
```

The default encoding for a Teradata table is `LATIN`. Teradata has very few built-in encodings, but it supports user-defined encodings.

The default value of the `StringEncoding` property is `ASCII`.



**Important:** For many situations involving a UTF-8 database, you should use `WIDEUTF-8` pseudo-encoding, described in the `WIDEUTF-8` section.

### WIDEUTF-8

Unica Campaign is normally responsible for transcoding between its internal encoding, `UTF-8`, and the encoding of the database. When the database is encoded in `UTF-8`, the value `UTF-8` can be specified for `StringEncoding` (except for `SQLServer`), and no transcoding will be needed. Traditionally, these have been the only viable models for Unica Campaign to access non-English data within a database.

In the 7.0 version of Unica Campaign, a new database encoding called `WIDEUTF-8` was introduced as a value for the `StringEncoding` property. By using this encoding, Unica Campaign still uses `UTF-8` to communicate with the database client, but allows the client to perform the task of transcoding between `UTF-8` and the encoding of the actual database. This enhanced version of `UTF-8` is needed to alter the widths of table column mappings so that they will be wide enough for transcoded text.



**Note:** The `WIDEUTF-8` pseudo-encoding may be used only in the database configuration. It should not be used for any other purpose.



**Note:** Oracle does not support transcoding through the client.

## SuffixOnAllOtherSQL

### Configuration category

`Campaign|partitions|partition[n]|dataSources|datasourcename`

### Description

The `SuffixOnAllOtherSQL` property specifies a string that is automatically appended to every SQL expression, generated by Unica Campaign, which are not covered by the `SuffixOnInsertSQL`, `SuffixOnSelectSQL`, `SuffixOnTempTableCreation`, `SuffixOnUserTableCreation`, or `SuffixOnUserBaseTableCreation` properties.

This property applies only to SQL generated by Unica Campaign, and does not apply to SQL in raw SQL expressions used in the Select process.

`SuffixOnAllOtherSQL` is used for the following expression types, when generated by Unica Campaign:

```
TRUNCATE TABLE table
DROP TABLE table
DELETE FROM table [WHERE ...]
UPDATE table SET ...
```

This property is automatically added to the SQL expression without checking its syntax. If you use this parameter, make sure that it is a legal expression. The string may be enclosed in quotation marks, but this is not required.

This property is undefined by default.

Tokens available to `SuffixOnAllOtherSQL` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

**Default value**

No default value defined.

**SuffixOnCreateDateField****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

**Description**

The `SuffixOnCreateDateField` property specifies a string that Unica Campaign automatically appends to any `DATE` fields in the `CREATE TABLE` SQL statement.

For example, you might set this property as follows:

```
SuffixOnCreateDateField = FORMAT 'YYYY-MM-DD'
```

If this property is undefined (the default), the `CREATE TABLE` command is unchanged.



**Note:** See the table in the description of the `DateFormat` property.

**Default value**

No default value defined.

## SuffixOnExtractTableCreation

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use the `SuffixOnExtractTableCreation` property to specify a string that is automatically appended to the SQL expression generated by Unica Campaign when an Extract table is created.

Tokens available to `SuffixOnExtractTableCreation` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Extract table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Extract table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Extract table was created.
<DBUSER>	This token is replaced with the database user name for the database where the Extract table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Extract table creation.
<KEYCOLUMNS>	This token is replaced with the Extract table column name(s).
<TABLENAME>	This token is replaced with the Extract table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

### Default value

Not defined

### Valid Values

Valid SQL

## SuffixOnInsertSQL

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `SuffixOnInsertSQL` property specifies a string that is automatically appended to all `INSERT SQL` expressions generated by Unica Campaign. This property applies only to SQL generated by Unica Campaign, and does not apply to SQL in raw SQL expressions used in the Select process.

`SuffixOnInsertSQL` is used for the following expression type, when generated by Unica Campaign:

```
INSERT INTO table ...
```

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. The string may be enclosed in quotation marks, but this is not required.

This property is undefined by default.

Tokens available to `SuffixOnInsertSQL` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

### Default value

No default value defined.

## SuffixOnSegmentTableCreation

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourceName*

### Description

Specifies a string that is automatically appended to the SQL expression generated by Unica Campaign when a Segment temp table is created.

Tokens available to **SuffixOnSegmentTableCreation** are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Segment temp table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Segment temp table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Segment temp table was created.
<DBUSER>	This token is replaced with the database user name for the database where the Segment temp table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Segment temp table creation.
<KEYCOLUMNS>	This token is replaced with the Segment temp table column name(s).
<TABLENAME>	This token is replaced with the Segment temp table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

### Default value

Not defined

### Valid Values

Valid SQL

## SuffixOnSelectSQL

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourceName*

### Description

The `SuffixOnSelectSQL` property specifies a string that is automatically appended to all `SELECT` SQL expressions generated by Unica Campaign. This property applies only to SQL generated by Unica Campaign, and does not apply to SQL in "raw SQL" expressions used in the Select process.

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. The string may be enclosed in quotation marks, but this is not required.

This property is undefined by default.

Tokens available to `SuffixOnSelectSQL` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

### Default value

No default value defined.



## SuffixOnSnapshotTableCreation

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use the `SuffixOnSnapshotTableCreation` property to specify a string that is automatically appended to the SQL expression generated by Unica Campaign when a Snapshot table is created.

Tokens available to `SuffixOnSnapshotTableCreation` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the Snapshot table was created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the Snapshot table was created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the Snapshot table was created.
<DBUSER>	This token is replaced with the database user name for the database where the Snapshot table was created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the Snapshot table creation.
<TABLENAME>	This token is replaced with the Snapshot table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

### Default value

Not defined

### Valid Values

Valid SQL

## SuffixOnTempTableCreation

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use this property to specify a string that is automatically appended to the SQL expression generated by Unica Campaign when a temp table is created. This property applies only to SQL generated by Unica Campaign, and does not apply to SQL in "raw SQL" expressions used in the Select process. To use this property, the `AllowTempTables` property must be set to `TRUE`.

You may want to use tokens to substitute the table name and the column name(s) (`<TABLENAME>` and `<KEYCOLUMNS>`) in this SQL statement, since these are generated dynamically during the execution of the campaign.

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. The string may be enclosed in quotation marks, but this is not required.

This property is undefined by default.



**Note:** For Oracle databases, the configuration parameter is appended to the temp table creation SQL expression after the table name.

Tokens available to `SuffixOnTempTableCreation` are described below.

Token	Description
<code>&lt;AMUSER&gt;</code>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<code>&lt;CAMPAIGNCODE&gt;</code>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<code>&lt;CAMPAIGNNAME&gt;</code>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<code>&lt;DBUSER&gt;</code>	This token is replaced with the database user name for the database where the temp tables were created.
<code>&lt;FLOWCHARTNAME&gt;</code>	This token is replaced with the name of the flowchart associated with the temp table creation.
<code>&lt;KEYCOLUMNS&gt;</code>	This token is replaced with the temp table column name(s).

Token	Description
<TABLENAME>	This token is replaced with the temp table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

**Default value**

No default value defined.

**SuffixOnUserBaseTableCreation****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourceName*

**Description**

Use the `SuffixOnUserBaseTableCreation` property to specify a string that is automatically appended to the SQL expression that Unica Campaign generates when a user creates a Base table (for example, in an Extract process). This property applies only to SQL generated by Unica Campaign, and does not apply to SQL in "raw SQL" expressions used in the Select process.

You may want to use tokens to substitute the table name and the column name(s) (<TABLENAME> and <KEYCOLUMNS>) in this SQL statement, since these are generated dynamically during the execution of the campaign.

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. The string may be enclosed in quotation marks, but this is not required.

This property is undefined by default.

Tokens available to **SuffixOnUserBaseTableCreation** are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<KEYCOLUMNS>	This token is replaced with the temp table column name(s).
<TABLENAME>	This token is replaced with the temp table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

**Default value**

No default value defined.

**SuffixOnUserTableCreation****Configuration category**

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

**Description**

Use the `SuffixOnUserTableCreation` property to specify a string that is automatically appended to the SQL expression that Unica Campaign generates when a user creates a General table (for example, in a Snapshot process). This property applies only to SQL generated by Unica Campaign, and does not apply to SQL in "raw SQL" expressions used in the Select process.

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. The string may be enclosed in quotation marks, but this is not required.

This property is undefined by default.

Tokens available to `SuffixOnUserTableCreation` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.

Token	Description
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<TABLENAME>	This token is replaced with the temp table name.

**Default value**

No default value defined.

**SystemTableSchema****Configuration category**

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

**Description**

Specifies the schema used for Unica Campaign system tables.

The default value is blank. This parameter is only relevant for the `UA_SYSTEM_TABLES` data source.

This property is not required for SQL server. For other data sources, set this property to the user of the database to which you are trying to connect.

You can leave this value blank unless the `UA_SYSTEM_TABLES` data source contains multiple schemas (for example, an Oracle database used by multiple groups). In this context, "schema" indicates the initial portion of a "qualified" table name of the form `x.y`, where `x` is the schema and `y` is the unqualified table name. For example: `dbo.UA_Folder`. This terminology for this syntax differs among the different database systems supported by Unica Campaign.)

If multiple schemas exist in the system tables database, set this value to the name of the schema in which the Unica Campaign system tables were created.

**Default value**

No default value defined.

## TableListSQL

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use the `TableListSQL` property to specify the SQL query to use to include synonyms in the list of tables available to map.

The default value is blank. This property is required if your data source is SQL Server and you want to be able to map synonyms in the returned table schema. This property is optional if you want to use a specific SQL query with other data sources in place of, or in addition to, the table schema information retrieved using the standard methods (such as an ODBC call or native connection).



**Note:** To ensure that Campaign works with SQL Server synonyms, you must set the `UseSQLToRetrieveSchema` property to `TRUE` in addition to setting this property as described here.

If you set this property with a valid SQL query, Unica Campaign issues the SQL query to retrieve the list of tables for mapping. If the query returns one column, it is treated as a column of names; if the query returns two columns, the first column is assumed to be a column of owner names, and the second column is considered to be a column of table names.

If the SQL query does not begin with an asterisk (\*), Unica Campaign merges this list with the list of tables that are normally retrieved (such as through ODBC calls or native connections).

If the SQL query begins with an asterisk (\*), the list returned by the SQL replaces the normal list, rather than being merged with it.

### Default value

None

### Valid Values

A valid SQL query

### Example

If the data source is SQL Server, under normal circumstances the ODBC API call that Unica Campaign uses returns a list of tables and views, but no synonyms. To include the list of synonyms as well, set `TableListSQL` similar to the following example:

```
select B.name AS oName, A.name AS tName
from sys.synonyms A LEFT OUTER JOIN sys.schemas B
on A.schema_id = B.schema_id ORDER BY 1, 2
```

To retrieve the list of tables, views, and synonyms, avoiding the ODBC API completely, set `TableListSQL` similar to the following example:

```
*select B.name AS oName, A.name AS tName from
  (select name, schema_id from sys.synonyms UNION
  select name, schema_id from sys.tables UNION select name,
  schema_id from sys.views) A LEFT OUTER JOIN sys.schemas B on
  A.schema_id = B.schema_id ORDER BY 1, 2
```

If the data source is Oracle, you can use a query similar to the following to retrieve the list of tables, views, and synonyms in place of the data retrieved using the native connection method that looks at the `ALL_OBJECTS` view:

```
*select OWNER, TABLE_NAME from (select OWNER, TABLE_NAME
  from ALL_TABLES UNION select OWNER, SYNONYM_NAME AS TABLE_NAME
  FROM ALL_SYNONYMS UNION select OWNER,
  VIEW_NAME AS TABLE_NAME from ALL_VIEWS) A ORDER BY 1, 2
```

## TempTablePostExecutionSQL

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

Use this property to specify a complete SQL statement that Unica Campaign runs immediately after the creation of a temporary table in a user data source or in the system tables database. For example, to improve performance, you can create an index on a temporary table immediately after its creation (see examples below). To enable the creation of temporary tables in a data source, the `AllowTempTables` property must be set to `TRUE`.

You can use tokens to substitute the table name (`<TABLENAME>`) and column names (`<KEYCOLUMNNS>`) in the SQL statement, because the values are generated dynamically when the campaign runs.

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. You can enclose the string in quotation marks, but this is not required.

This property treats semicolons as delimiters to run multiple SQL statements. If your SQL statement contains semicolons and you want it to run as one statement, use a backslash as an escape character before the semicolons.



**Note:** If you are using stored procedures with this property, be sure that you use the correct syntax for your database.

Tokens available to **TempTablePostExecutionSQL** are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<KEYCOLUMNS>	This token is replaced with the temp table column name(s).
<TABLENAME>	This token is replaced with the temp table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

#### Examples

The following value creates an index on the temp table just after its creation, to improve the data retrieval process: `CREATE INDEX IND_<TABLENAME> ON <TABLENAME> (<KEYCOLUMNS>)`

The following example for Oracle calls a stored procedure and uses backslashes to escape the semicolon: `begin dbms_stats.collect_table_stats()\; end\;`

## TempTablePrefix

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description



This property specifies a string that is automatically prepended to the names of all temporary tables created by Unica Campaign. Use this property to help identify and manage temp tables. You also can use this property to cause temp tables to be created in a particular location.

For example, if the user token corresponds to a schema, you can set

```
TempTablePrefix="<USER>"
```

and all temp tables will be created in the schema of whatever user is connected to the data source.

If two or more data sources point to the same database, errors and incorrect search results can occur during flowchart runs due to usage of the same temporary tables by different process boxes and flowcharts. This situation can also occur with Extract process tables and Strategic Segment tables. To avoid this situation, use `TempTablePrefix` (or `ExtractTablePrefix` for Extract tables) to define different schemas for each data source. This approach ensures that the initial part of the name is different, so the table names will always be different.

For example, give each data source a unique `TempTablePrefix` such as `UAC_DS1` and `UAC_DS2` to distinguish between temp tables for each data source. The same concept applies if you are sharing data source schemas. For example, the following prefixes allow the temp tables to be unique for both data sources that write temp tables to the same database:

DS1 `TempTablePreFix`: `schemaA.UAC_DS1`

DS2 `TempTablePreFix`: `schemaA.UAC_DS2`

The following table describes the tokens that are available to **TempTablePrefix**.



**Note:** You must make sure that the final temp table name after resolving tokens does not exceed any database-specific name length restrictions.



**Note:** In tokens used for `TempTablePrefix`, any characters that are not valid for database table names will be stripped. After tokens are resolved, the resulting temp table prefixes must start with an alphabetic character, and must contain only alphanumeric characters or underscore characters. Illegal characters will be removed silently. If any resulting temp table prefix does not begin with an alphabetic character, Unica Campaign prepends the letter "U" to the prefix.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.

Token	Description
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

**Default value**

UAC

**TempTablePreTruncateExecutionSQL****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

**Description**

**Note:** This property is supported only for Teradata data sources. For all other supported databases, this property should not be set.

Use this property to specify a SQL query to run before a temp table is truncated. The query that you specify can be used to negate the effect of a SQL statement specified in the **TempTablePostExecutionSQL** property.

For example, with the **TempTablePostExecutionSQL** property, you could specify the following SQL statement to create an index:

```
CREATE INDEX <TABLENAME>Idx_1 (<KEYCOLUMNS>) ON <TABLENAME>
```

Then, specify the following query in the **TempTablePreTruncateExecutionSQL** property to drop the index:

```
DROP INDEX <TABLENAME>Idx_1 ON <TABLENAME>
```

**Default value**

Not defined

#### Valid Values

A valid SQL query

### TempTablePreTruncateRunScript

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourceName`

#### Description



**Note:** This property is supported only for Teradata data sources. For all other supported databases, this property should not be set.

Use this property to specify a script or executable to run before a temp table is truncated. The script that you specify can be used to negate the effect of a SQL statement specified in the **PostTempTableCreateRunScript** property.

For example, with the **PostTempTableCreateRunScript** property, you could specify a script that includes the following SQL statement to create an index:

```
CREATE INDEX <TABLENAME>Idx_1 (<KEYCOLUMNS>) ON <TABLENAME>
```

Then, specify another script with the following statement in the **TempTablePreTruncateRunScript** property to drop the index:

```
DROP INDEX <TABLENAME>Idx_1 ON <TABLENAME>
```

#### Default value

Not defined

#### Valid Values

File name of a shell script or executable

### TeradataDeleteBeforeDrop

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourceName`

#### Description

This property applies only to Teradata data sources. It specifies whether records are deleted before a table is dropped.

Set the value to `TRUE` to delete all records from a table before dropping the table.



**Note:** If Unica Campaign is unable to delete the records for any reason, it will not drop the table.

Set the value to `FALSE` to drop a table without first deleting all records.

#### Default value

`TRUE`

## TruncateSQL

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

This property is available for use with DB2® data sources, and allows you to specify alternate SQL for table truncation. This property applies only when **DeleteAsTruncate** is `TRUE`. When **DeleteAsTruncate** is `TRUE`, any custom SQL in this property is used to truncate a table. When this property is not set, Unica Campaign uses the `TRUNCATE TABLE <TABLENAME>` syntax.

This property is undefined by default.

Tokens available to **TruncateSQL** are described below.

Token	Description
<TABLENAME>	This token is replaced with the database table name that Unica Campaign is truncating.

#### Default value

No default value defined.

## Type

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

This property specifies the database type of this data source.

#### Default value

The default value depends on the database template that was used to create the data source configuration.

#### Valid Values

Valid Types for system tables are:

- DB2
- DB2ODBC
- ORACLE
- ORACLE8
- ORACLE9
- SQLServer

Valid Types for customer tables are:

- BigDataODBC\_Hive
- DB2
- DB2ODBC
- NETEZZA
- ORACLE
- ORACLE8
- ORACLE9
- PostgreSQL
- SQLServer
- TERADATA

## UOSQLOnConnect

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

The `SQLOnConnect` property defines a complete SQL statement that Unica Campaign runs immediately after each database connection. The `UOSQLOnConnect` property is similar to this, but specifically applicable to Unica Optimize.

The SQL statement generated by this property is automatically passed to your database without checking its syntax. If you use this property, make sure that it is a legal expression. The string may be enclosed in quotation marks, but this is not required.

This property is undefined by default.

Tokens available to `UOSQLOnConnect` are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.

Token	Description
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

**Default value**

No default value defined.

**UseAliasForPredicate****Configuration category**

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

**Description**

This property is used only for Hive-based Hadoop data sources (BigDataODBC HiveTemplate). Set the value to TRUE if you are connecting to IBM BigInsight Hadoop instance. Set to FALSE if you are connecting to any other Hive based Hadoop instance.



**Note:** If you are upgrading to version 11.1 and you have already configured and are using a Hive based Hadoop data source, you do not need to make any changes for the existing instance.

**Default value**

FALSE

**Valid Values**

TRUE | FALSE

**UseExceptForMerge****Configuration category**

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

### Description

When Unica Campaign performs exclusions in the Merge process or in the Segment process, by default it uses `NOT EXISTS` syntax, as:

```
SELECT IncludeTable.ID FROM IncludeTable WHERE NOT EXISTS
(SELECT * FROM ExcludeTable WHERE IncludeTable.ID = ExcludeTable.ID)
```

If `UseExceptForMerge` is `TRUE` and you cannot use `NOT IN` (because `UseNotInForMerge` is disabled or because the audience level consists of multiple fields and the data source is not Oracle), then the syntax is altered as follows:

### Oracle

```
SELECT IncludeTable.ID FROM IncludeTable
MINUS (SELECT ExcludeTable.ID FROM ExcludeTable)
```

### Others

```
SELECT IncludeTable.ID FROM IncludeTable
EXCEPT (SELECT ExcludeTable.ID FROM ExcludeTable)
```

For Hive-based Hadoop big data, this property must be `FALSE`. Hive does not support the `EXCEPT` clause, so a setting of `TRUE` can result in process failures.

### Default value

`FALSE`

### Valid Values

`TRUE` | `FALSE`

## UseGroupByForDistinct

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

### Description

The `UseGroupByForDistinct` property is available for the Teradata datasource template. By default, this property is `FALSE`. When this property is enabled, <select query> uses a `GROUP BY` statement instead of `DISTINCT`.

The purpose of this property: Flowcharts execute the query "select DISTINCT <audience id> from <table>" to fetch unique records when a table is not mapped as a normalized table in Unica Campaign. When such queries are submitted against Teradata, it causes additional sorting of data on the database and high CPU consumption. The Teradata DBA recommendation is to use GROUP BY instead of DISTINCT, because GROUP BY can take advantage of Teradata multi AMP processing architecture.

#### Default value

FALSE

#### Valid Values

TRUE | FALSE

### UseMergeForTrack

#### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

#### Description

This property implements SQL MERGE syntax to improve the performance of the Track process in flowcharts. This property can be set to `TRUE` for DB2®, Oracle, SQL Server 2008, and Teradata 12. It can also be used with other databases that support the SQL MERGE statement.

#### Default value

TRUE (DB2 and Oracle) | FALSE (all others)

#### Valid Values

TRUE | FALSE

### UseNonANSIJoin

#### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

#### Description

The `UseNonANSIJoin` property specifies whether this data source uses non-ANSI join syntax. If the data source type is set to Oracle7 or Oracle8, and the value of `UseNonANSIJoin` is set to `TRUE`, the data source uses non-ANSI join syntax appropriate for Oracle.

#### Default value

FALSE

#### Valid Values

TRUE | FALSE



## UseNotInForMerge

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

### Description

When Unica Campaign performs exclusions in the Merge process or in the Segment process, by default it uses `NOT EXISTS` syntax, as:

```
SELECT IncludeTable.ID FROM IncludeTable WHERE NOT EXISTS (SELECT *
FROM ExcludeTable WHERE IncludeTable.ID = ExcludeTable.ID)
```

If **UseNotInForMerge** is enabled and either (1) the audience level is composed of a single ID field, or (2) the data source is Oracle, then the syntax is altered as follows:

```
SELECT IncludeTable.ID FROM IncludeTable WHERE IncludeTable.ID NOT IN
(SELECT ExcludeTable.ID FROM ExcludeTable)
```

### Default value

FALSE

### Valid Values

TRUE | FALSE

## UseNotInToDeleteCH

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

### Description

This property affects the Unica Campaign system table data source (UA\_SYSTEM\_TABLES). It affects the SQL query syntax for how the MailList and CallList processes remove records from the Unica Campaign system tables.

The default value of FALSE typically improves database performance. The default behavior uses EXISTS / NOT EXISTS when removing Contact History records (either after a failed run or in response to the user's action in the GUI). The removal process involves deleting from UA\_OfferHistAttrib and updating UA\_OfferHistory.

You can change this value to TRUE if you prefer to use the SQL syntax of IN / NOT IN. Prior versions of Unica Campaign used IN / NOT IN.

### Default value

FALSE

## Valid Values

TRUE | FALSE

## UserBaseTablePostExecutionSQL

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

### Description

This property is invoked when a process box is configured to write to a **New Mapped Table > Base Record Table > Create New Table in Selected Database**. This property is invoked only when the table is created (during the creation and mapping process). This property is not invoked during process box runtime.

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. You can enclose the string in quotation marks, but this is not required.

This property treats semicolons as delimiters to run multiple SQL statements. If your SQL statement contains semicolons and you want it to run as one statement, use a backslash as an escape character before the semicolons.



**Note:** If you are using stored procedures with this property, be sure to use the correct syntax for your database. The following example for Oracle calls a stored procedure and uses backslashes to escape the semicolon: `begin dbms_stats.collect_table_stats()\; end\;`

You can use tokens to substitute the `<TABLENAME>` in this SQL statement because the name is generated dynamically when the campaign runs. For available tokens, see **UserTablePostExecutionSQL**.

## UserTablePostExecutionSQL

### Configuration category

*Campaign|partitions|partition[n]|dataSources|dataSourcename*

### Description

Use this property to specify a complete SQL statement that Unica Campaign runs immediately after the creation of a user table in a user data source or in the system tables database. This property is invoked when a process box writes to one of the following tables:

- **New mapped table > General table > Create new table in selected datasource:** The property is invoked during the creation/mapping process; not during Snapshot runtime.
- **New mapped table > Dimension table > Create new table in selected database:** The property is invoked during the creation/mapping process; not during Snapshot runtime.
- **Database table:** The property is invoked during process box runtime.

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. You can enclose the string in quotation marks, but this is not required.

This property treats semicolons as delimiters to run multiple SQL statements. If your SQL statement contains semicolons and you want it to run as one statement, use a backslash as an escape character before the semicolons.



**Note:** If you are using stored procedures with this property, be sure to use the correct syntax for your database. The following example for Oracle calls a stored procedure and uses backslashes to escape the semicolon: `begin dbms_stats.collect_table_stats()\; end\;`

You can use tokens to substitute the `<TABLENAME>` in this SQL statement, because the name is generated dynamically when the campaign runs.

Tokens available to **UserTablePostExecuteSQL** are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which the user tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which the user tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which the user tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the user tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the user table creation.
<TABLENAME>	This token is replaced with the user table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

## UseSQLToProfile

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

This property allows you to configure Unica Campaign to submit the SQL query `GROUP BY` to the database to compute profiles (using `SELECT field, count(*) FROM table GROUP BY field`), rather than fetching records.

- A value of `FALSE` (the default) causes Unica Campaign to profile a field by retrieving the field value for all records in the table and to track the count of each distinct value.
- A value of `TRUE` causes Unica Campaign to profile a field by issuing a query similar to the following:

```
SELECT field, COUNT(*) FROM table GROUP BY field
```

which pushes the burden to the database.

### Default value

`FALSE`

### Valid Values

`TRUE | FALSE`

## UseSQLToRetrieveSchema

### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

### Description

This property determines whether Unica Campaign uses a SQL query, rather than an ODBC or native API call, to retrieve the schema to use as the table schema for this data source.

The default value is `FALSE`, indicating that Unica Campaign should use its standard method (ODBC or native connection, for example) to retrieve the schema. Setting this property to `TRUE` causes Unica Campaign to prepare a SQL query similar to `select * from <table>` to retrieve the table schema.

This can provide advantages that are specific to each data source. For example, some data sources (Netezza®, SQL Server) do not properly report SQL synonyms (alternative names for database objects, defined using the `create synonym` syntax) through the default ODBC or native connections. By setting this property to `TRUE`, SQL synonyms are retrieved for data mapping within Unica Campaign.

The following list describes the behavior of this setting for a number of data sources:

- Hive-based Hadoop big data: Use the default setting of FALSE.
- Netezza®: Set this property to TRUE to allow support for SQL synonyms. No other settings or values are needed to support synonyms in Netezza® data sources.
- SQL Server: To allow support for synonyms, set this property to TRUE **and** enter valid SQL in the `TableListSQL` property for this data source. See the description for the `TableListSQL` property for more details.
- Oracle: Set this property to TRUE to tell Unica Campaign to prepare the SQL query to retrieve the table schema. The result set identifies `NUMBER` fields (no precision/scale specified, which may cause issues in Unica Campaign) as `NUMBER(38)`, which avoids those possible issues.
- For other data sources, you can optionally set this property to TRUE to use the default SQL select query described above, or to specify valid SQL in the `TableListSQL` property to use instead of, or in addition to, the ODBC API or native connection that is used by default. See the description for the `TableListSQL` property for more details.

#### Default value

FALSE

#### Valid Values

TRUE | FALSE

#### Example

To allow Unica Campaign to work with Netezza® or SQL Server synonyms:

```
UseSQLToRetrieveSchema=TRUE
```

## UseTempTablePool

#### Configuration category

`Campaign|partitions|partition[n]|dataSources|dataSourcename`

#### Description

When `UseTempTablePool` is set to `FALSE`, temp tables are dropped and re-created every time a flowchart is run. When the property is set to `TRUE`, temp tables are not dropped from the database. Temp tables are truncated and reused from the pool of tables maintained by Unica Campaign. The temp table pool is most effective in environments where you rerun flowcharts many times, such as during a design and test phase.

#### Default value

FALSE

#### Valid Values

TRUE | FALSE

## MakeStringColumnAsBlobBeyondRowSize

### Configuration category

`Campaign|partitions|partition[n]|dataSources|datasourcename`

### Description

This is MariaDB specific configuration parameter, to be used in deliver OLT creation flow. MYSQL database has a limitation of maximum row size to be 65535 bytes. If the OLT table row size limit exceeds limitation, then the deliver PB fails with error Deliver-1515. Set the property to TRUE for the CLOB column in Deliver PB to be mapped to TEXT column in Deliver OD side and OLT table gets created with CLOB columns so that OLT table row size does not exceed allowed size.

### Default value

FALSE

## Campaign | partitions | partition[n] | systemTableMapping

Properties in the systemTableMapping category are populated automatically if you remap any system tables or map Contact or Response history tables. You should not edit properties in this category.

## Campaign | partitions | partition[n] | server

This category contains properties to configure the Unica Campaign server for the selected partition.

## Campaign | partitions | partition[n] | server | systemCodes

Properties in this category specify, for Unica Campaign, whether variable length codes are allowed, the format and generator of the campaign and cell codes, whether offer codes are displayed, and the offer code delimiter.

### offerCodeDelimiter

#### Configuration category

`Campaign|partitions|partition[n]|server|systemCodes`

#### Description

The `offerCodeDelimiter` property is used internally to concatenate multiple code parts (for example, to output the OfferCode field in Unica Campaign Generated Fields) and for incoming offer codes in the Unica Campaign Response process, to split the offer code into multiple parts. The value must be only a single character.

Older versions of Unica Campaign included a `NumberOfOfferCodesToUse` parameter. However, in more recent versions, this value comes from the offer template (every offer template can have a different number of offer codes).

#### Default value

-

## allowVariableLengthCodes

### Configuration category

Campaign | partitions | partition[n] | server | systemCodes

### Description

The `allowVariableLengthCodes` property specifies whether variable length codes are allowed in Unica Campaign.

If the value is `TRUE`, and if the trailing part of the code format is `x`, the length of the code can vary. For example, if the code format is `nnnnxxxx`, then the code can be from 4 to 8 characters long. This applies to campaign, offer, version, tracking, and cell codes.

If the value is `FALSE`, variable length codes are not allowed.

### Default value

`FALSE`

### Valid Values

`TRUE` | `FALSE`

## displayOfferCodes

### Configuration category

Campaign | partitions | partition[n] | server | systemCodes

### Description

The `displayOfferCodes` property specifies whether to show offer codes beside their names in the Unica Campaign GUI.

If the value is `TRUE`, offer codes are displayed.

If the value is `FALSE`, offer codes are not displayed.

### Default value

`FALSE`

### Valid Values

`TRUE` | `FALSE`

## cellCodeFormat

### Configuration category

Campaign | partitions | partition[n] | server | systemCodes

### Description

The `cellCodeFormat` property is used by the campaign code generator to define the format of the cell code that is automatically created by the default cell code generator. For a list of valid values, see `campCodeFormat`.

#### Default value

Annnnnnnnn

### campCodeFormat

#### Configuration category

Campaign|partitions|partition[n]|server|systemCodes

#### Description

The `campCodeFormat` property is used by the campaign code generator to define the format of the campaign code that is automatically generated by the default campaign code generator when you create a campaign.

#### Default value

Cnnnnnnnnn

#### Valid Values

The possible values are:

- `A-Z` or any symbol - treated as a constant
- `a` - random letters A-Z (upper case only)
- `c` - random letters A-Z or numbers 0-9
- `n` - random digit 0-9
- `x` - any single ASCII character from `0-9` or `A-Z`. You can edit the generated campaign code and replace the ASCII character that Unica Campaign substituted for the `x` with any ASCII character, and Unica Campaign will use that character instead.

### cellCodeGenProgFile

#### Configuration category

Campaign|partitions|partition[n]|server|systemCodes

#### Description

The `cellCodeGenProgFile` property specifies the name of the cell code generator. The properties that control the format of the code generated are set in the `cellCodeFormat` property. See `campCodeGenProgFile` for a list of supported options.



If you write your own cell code generator, replace the default value with the absolute path of your custom program, including the file name and extension, and using forward slashes (/) for UNIX™ and backslashes (\) for Windows™.

#### Default value

`uaccampcodegen` (the code generator supplied by Unica Campaign)

### campCodeGenProgFile

#### Configuration category

`Campaign|partitions|partition[n]|server|systemCodes`

#### Description

This property specifies the name of the campaign code generator. The properties that control the format of the generated code are set in the `campCodeFormat` property.

If you write your own campaign code generator, replace the default value with the absolute path of your custom program, including the file name and extension, using forward slashes (/) for UNIX™ and backslashes (\) for Windows™.

The default campaign code generator can be called with the following options:

- `-y` Year (four integers)
- `-m` Month (one or two integers, cannot exceed value of twelve)
- `-d` Day (one or two integers, cannot exceed value of 31)
- `-n` Campaign name (any string, cannot exceed 64 characters)
- `-o` Campaign owner (any string, cannot exceed 64 characters)
- `-u` Campaign code (any integer). Allows you to specify the exact campaign ID rather than having the application generate one for you.
- `-f` Code format if overriding the default. Takes the values specified in `campCodeFormat`.
- `-i` Other integer.
- `-s` Other string.

#### Default value

`uaccampcodegen` (the code generator supplied by Unica Campaign)

### cellCodeBulkCreation

#### Configuration category

`Campaign|partitions|partition[n]|server|systemCodes`

### Description

A value of TRUE improves performance of the cell code generation utility during bulk creation of cell codes, because multiple cell codes are generated with a single invocation of the cell code generator. This is more efficient and is the recommended setting. A value of TRUE also improves performance when copying flowcharts, templates, and process boxes.

When the value is FALSE, the cell code generator is invoked once for each cell code generation. If cell code generation seems to take a long time for Segment, Sample, and Decision process boxes, or for the target cell spreadsheet, set this value to TRUE.

The default setting is FALSE to support existing customized implementations. If you are using a legacy custom-made cell code generation utility, leave this setting at its default value of FALSE until you implement a new custom utility. Then you can change its value to TRUE.

If you are not using a custom cell code generation utility, change the value to TRUE to take advantage of the efficiency improvements.

### Default value

FALSE

### Valid Values

TRUE | FALSE

## Campaign | partitions | partition[n] | server | encoding

The property in this category specifies the text encoding for values written to files, to support non-English data.

### stringEncoding

#### Description

The `partition[n] > server > encoding > stringEncoding` property how Unica Campaign reads in and writes out flat files. It should match the encoding used for all flat files. If not configured elsewhere, this is the default setting for flat file encoding.



**Note:** `WIDEUTF-8` is not supported for this setting.

By default, no value is specified, and outgoing text files are encoded as UTF-8, which is the default encoding for Unica Campaign.

It is a best practice to explicitly set this value to an encoding appropriate for your system, even if the value is UTF-8, the same as the implicit default.



**Note:** If you do not set the value of the `StringEncoding` property for data sources in the `dataSources` category, the value of this `stringEncoding` property is used as the default value. This can cause unnecessary confusion – you should always explicitly set the `StringEncoding` property in the `dataSources` category.

See the Unica Campaign Administrator's Guide for a list of supported encodings.

#### Default value

No default value defined.

### forceDCTOneBytePerChar

#### Description

The `forceDCTOneBytePerChar` property specifies whether Unica Campaign should use the original field width for output files, rather than the potentially expanded width reserved to allow sufficient space for transcoding into UTF-8.

A text value may have different lengths, depending on the encoding used to represent it. When the text value comes from a data source whose `stringEncoding` property is neither ASCII nor UTF-8, Unica Campaign reserves three times the field width in order to ensure sufficient space for transcoding into UTF-8. For example, if the `stringEncoding` property is set to `LATIN1`, and the field in the database is defined as `VARCHAR(25)`, Unica Campaign will reserve 75 bytes to hold the transcoded UTF-8 value. Set the `forceDCTOneBytePerChar` property to `TRUE` if you want to use the original field width.

#### Default value

FALSE

#### Valid Values

TRUE | FALSE

### Campaign | partitions | partition[n] | server | timeout

The properties in this category specify the number of seconds an Unica Campaign flowchart waits, after the user has disconnected and all runs have completed, before exiting, and the Unica Campaign server process waits for a response from external servers before reporting an error.

### waitForGracefulDisconnect

#### Description

The `waitForGracefulDisconnect` property specifies whether the Unica Campaign server process continues to run until the user gracefully disconnects, or exits regardless of whether the user intended to disconnect.

If the value is `yes`, the default, the server process continues to run until it can determine that the user wants it to exit. This option prevents changes from being lost, but can result in server processes accumulating.

If the value is `no`, the server process shuts down and server processes are prevented from accumulating, but users can lose work if a network interruption occurs or if they do not follow the recommended sequence of actions to exit gracefully.

**Default value**

`TRUE`

**Valid Values**

`TRUE | FALSE`

**urlRequestTimeout**

**Description**

The `urlRequestTimeout` property specifies the number of seconds the Unica Campaign server process waits for a response from external servers. Currently, this applies to requests to Unica servers and Unica Deliver components that operate with Unica Campaign.

If the Unica Campaign server process does not receive a response within this period, a communication timeout error is reported.

**Default value**

`60`

**delayExitTimeout**

**Description**

The `delayExitTimeout` property specifies the number of seconds an Unica Campaign flowchart waits, after the user has disconnected and all runs have completed, before exiting.

Setting this property to a non-0 value enables subsequent Unica Campaign flowcharts to make use of existing instances rather than starting a new instance.

**Default value**

`10`

Campaign | partitions | partition[n] | server | collaborate

This category applies to Unica Collaborate.

**collaborateInactivityTimeout**

**Configuration category**

`Campaign|partitions|partition[n]|server|collaborate`

### Description

The `collaborateInactivityTimeout` property specifies the number of seconds the `unica_acsvr` process waits after it finishes servicing a Unica Collaborate request before it exits. This waiting period allows the process to remain available in the typical scenario in which Unica Collaborate makes a series of requests prior to running the flowchart.

The minimum value is 1. Setting this property to 0 causes it to default to 60.

### Default value

60

## logToSeparateFiles

### Configuration category

`Campaign|partitions|partition[n]|server|collaborate`

### Description

This property was introduced in v8.6.0.6. By default and upon upgrade, the value of this parameter is False.

When True, flowchart logs for runs initiated from Unica Collaborate are logged to separate log files. Log files are created under folders with the current date to avoid an excessive number of log files in a single folder. The format of the folder name is "FlowchartRunLogs\_<YYYYMMDD>".

The format of the log file name is: <CAMP\_NAME>\_<CAMP\_CODE>\_<FC\_NAME>\_<PID>\_<LIST\_CODE>\_<DATE>\_<TIMESTAMP>.log, where PID is the Unica Campaign server process ID that ran the flowchart. LIST\_CODE is the object code of the Unica Collaborate list, ONDC or corporate campaign from which the flowchart was run.

All user variables that are passed to the flowchart run process are logged for troubleshooting purposes.



**Note:** When a flowchart is opened, it initially logs to the traditional flowchart log file. When a flowchart run is initiated from Unica Collaborate, if `logToSeparateFiles` is True, logging is done in the new directory and file at that time.

### Default value

False

### Valid Values

True | False

## Campaign | partitions | partition[n] | server | spss

Properties in this category affect IBM SPSS Modeler Advantage Enterprise Marketing Management Edition integration for the specified partition in Unica Campaign.

### SharedDirectoryPathCampaign

#### Description

The path to the directory that is used to transfer data between Unica Campaign and SPSS Modeler Server, as seen from Unica Campaign.

- Unica Campaign puts input data files to IBM SPSS Modeler Advantage Enterprise Marketing Management Edition in this directory.
- IBM SPSS Modeler Advantage Enterprise Marketing Management Edition puts output data files in this directory to be read and processed by Unica Campaign.

#### Default value

None

#### Valid values

Any valid Windows path (such as `z:\SPSS_Shared`) or a mount directory (for UNIX).

### SharedDirectoryPathSPSS

#### Description

The path to the directory that is used to transfer data between Unica Campaign and SPSS Modeler Server, as seen from SPSS Modeler Server. This is the same shared directory referenced by `SharedDirectoryPathCampaign` but is the local directory path used by SPSS Modeler Server.

For example, Unica Campaign might be installed on Windows with `SharedDirectoryPathCampaign = z:\SPSS_Shared`, where `z:\SPSS_Shared` is a mapped network drive, while SPSS® Modeler Server is installed on UNIX with a mount to that directory defined as `SharedDirectoryPathSPSS = /share/CampaignFiles`.

#### Default value

None

#### Valid values

Any valid Windows™ path (such as `z:\SPSS_Shared`) or a mount directory (such as `/share/CampaignFiles`) for UNIX.

### C&DS\_URL

#### Description

The URL for the IBM SPSS Collaboration and Deployment Services repository.

**Default value**

`http://localhost:7001/cr-ws/services/ContentRepository`

**Valid values**

The URL for the IBM SPSS Collaboration and Deployment Services repository.

**SPSS\_Integration\_Type****Description**

This property determines the type of integration between Unica Campaign and IBM SPSS Modeler Advantage Enterprise Marketing Management Edition.

**Default value**

None

**Valid values**

- `None`: No integration
- `SPSS MA Marketing Edition`: Full integration of modeling and scoring. This option is only available if IBM SPSS Modeler Advantage Enterprise Marketing Management Edition is installed and configured.
- `Scoring only`: Scoring, but not modeling, is enabled.

## Campaign | partitions | partition[n] | server | permissions

The properties in this category specify the permissions for folders that are created by Unica Campaign and the UNIX™ group and permissions for files in the `profile` directory.

**userFileGroup (UNIX™ only)****Description**

This property specifies a group associated with user-generated Unica Campaign files. The group will be set only if the user is a member of the specified group.

This property is undefined by default.

**Default value**

No default value defined.

**createFolderPermissions****Description**

The `createFolderPermissions` parameter specifies the permissions of directories that are created by Unica Campaign on the Unica Campaign server (partition[n] location) by using the Create Folder icon on the Open Data Source File dialog in the table mapping.

**Default value**

755 (owner has read/write/execute access, group and world have execute/read access)

**catalogFolderPermissions**

**Description**

The `catalogFolderPermissions` property specifies the permissions of directories created by Unica Campaign through the Stored table catalogs > Create folder window.

**Default value**

755 (owner has read/write/execute access, group and world have execute/read access)

**templateFolderPermissions**

**Description**

The `templateFolderPermissions` property specifies the permissions of template directories created by Unica Campaign through the **Stored templates > Create folder** window.

**Default value**

755 (owner has read/write/execute access, group and world have read/execute access)

**adminFilePermissions (UNIX™ only)**

**Description**

The `adminFilePermissions` property specifies a permission bit mask for the files contained in the `profile` directory.

**Default value**

660 (owner and group have read/write access only)

**userFilePermissions (UNIX™ only)**

**Description**

The `userFilePermissions` property specifies a permission bit mask for user generated Unica Campaign files (for example, log files, summary files, exported flat files).

**Default value**

666 (everyone can read and write files created by Unica Campaign in the server)

**adminFileGroup (UNIX™ only)**

**Description**

The `adminFileGroup` property specifies a UNIX™ admin group associated with files contained in the `profile` directory.



This property is undefined by default.

#### Default value

No default value defined.

## Campaign | partitions | partition[n] | server | flowchartConfig

Properties in this category specify the behavior of the Unica Campaign Generated Field, whether duplicate cell codes are allowed, and whether the Log to Contact History option defaults to enabled.

### allowDuplicateCellcodes

#### Description

The `allowDuplicateCellcodes` property specifies whether the cell codes in the Unica Campaign Snapshot process can have duplicate values.

If the value is `FALSE`, the Unica Campaign server enforces unique cell codes.

If the value is `TRUE`, the Unica Campaign server does not enforce unique cell codes.

#### Default value

`TRUE`

#### Valid Values

`TRUE` | `FALSE`

### allowResponseNDaysAfterExpiration

#### Description

The `allowResponseNDaysAfterExpiration` property specifies the maximum number of days after all offer expiration dates that responses can be tracked. These late responses can be included in performance reports.

#### Default value

90

### agfProcessnameOutput

#### Description

The `agfProcessnameOutput` property specifies the output behavior of the Unica Campaign Generated Field (UCGF) in the List, Optimize, Response, and Snapshot processes.

If the value is `PREVIOUS`, the UCGF contains the process name associated with the incoming cell.

If the value is `CURRENT`, the UCGF holds the process name of the process in which it is used.

### Default value

PREVIOUS

### Valid Values

PREVIOUS | CURRENT

## logToHistoryDefault

### Description

The `logToHistoryDefault` property specifies whether the Log to Contact History and Tracking Tables option in the Log tab of the Unica Campaign contact processes defaults to enabled.

If the value is `TRUE`, the option is enabled.

If the value is `FALSE`, the option is disabled in any newly created contact processes.

### Default value

TRUE

### Valid Values

TRUE | FALSE

## overrideLogToHistory

### Description

This property controls whether users with the appropriate permissions can change the Log to Contact History Tables setting when they configure a contact or Track process. To make all flowchart production runs always write to contact history, enable `logToHistoryDefault` and disable `overrideLogToHistory`

### Default value

TRUE

### Valid Values

TRUE | FALSE

## defaultBehaviorWhenOutputToFile

### Description

Specifies the behavior for contact processes in Unica Campaign when outputting to a file. This property applies only within the current partition. This default behavior (if set) is only applied for processes when they are newly added to flowcharts; once a process is added to a flowchart, the output behavior can be changed in the process configuration.

### Default value

Replace All Records

### Valid Values

- Append to Existing Data
- Create New File
- Replace All Records

## defaultBehaviorWhenOutputToDB

### Description

Specifies the behavior for contact processes in Unica Campaign when outputting to a database table. This property applies only within the current partition. This default behavior (if set) is only applied for processes when they are newly added to flowcharts; once a process is added to a flowchart, the output behavior can be changed in the process configuration.

### Default value

Replace All Records

### Valid Values

- Append to Existing Data
- Replace All Records

## replaceEmbeddedNames

### Description

When `replaceEmbeddedNames` is `TRUE`, Unica Campaign replaces user variable and UCGF names embedded in query text with actual values, although these names must be separated by a non-alphanumeric character, such as an underscore (for example, `ABC_UserVar.v1` will be substituted but `ABCUserVar.v1` will not). Set this property to `TRUE` for backwards compatibility with Unica Campaign 7.2 and earlier.

When set to `FALSE`, Unica Campaign replaces only distinct user variable and UCGF names with actual values (in both Unica and raw SQL expressions). Set this property to `FALSE` for backwards compatibility with Unica Campaign 7.3 and higher.

### Default value

FALSE

### Valid Values

TRUE | FALSE

## legacyMultifieldAudience

### Description

In most cases, you can leave this property set to the default value of `FALSE`. Unica Campaign v8.5.0.4 and newer name multifield Audience ID fields according to the audience definition, regardless of the source of the fields. When you configure processes to use multifield Audience ID fields, you now see the new Audience ID naming convention for multifield audiences. Already-configured processes in flowcharts created in previous Unica Campaign versions should continue to work. However, if old flowcharts fail because of the change in the naming convention, you can revert Unica Campaign behavior by setting this property to `TRUE`.

### Default value

`FALSE`

### Valid Values

`TRUE` | `FALSE`

## disableReplaceAll

### Description

In Unica Campaign the behaviour for contact process when outputting data to database are:

- Append to Existing data
- Replace All Records

If the value is `TRUE`, 'Replace All Records' behaviour won't be available when outputting data to database, user can only choose 'Append to Existing data' behaviour. All the existing configured contact processes will be unconfigured if user has chosen 'Replace All Records' behaviour and set this property value to `TRUE`. If the value is `FALSE` both behaviour available to user. The value of this property will be ignored when outputting data to file in contact process.

### Default value

`FALSE`

### Valid Values

`TRUE` | `FALSE`

## Campaign | partitions | partition[n] | server | flowchartSave

The properties in this category specify the default settings for the auto-save and checkpoint properties of a new Unica Campaign flowchart.

## checkpointFrequency

### Description

The `checkpointFrequency` property specifies (in minutes) the default setting for a new Unica Campaign flowchart's checkpoint property, configurable for each flowchart through the client-side Advanced Settings window. The checkpoint feature provides the ability to capture a snapshot of a running flowchart for recovery purposes.

**Default value**

0 (zero)

**Valid Values**

Any integer

## autosaveFrequency

**Description**

The `autosaveFrequency` property specifies (in minutes) the default setting for a new Unica Campaign flowchart's auto-save property, configurable for each flowchart through the client-side Advanced Settings window. The auto-save function performs a forced save of flowcharts during editing and configuration.

**Default value**

0 (zero)

**Valid Values**

Any integer

## Campaign | partitions | partition[n] | server | dataProcessing

Properties in this category specify how Unica Campaign handles string comparisons and empty fields in flat files, and the behavior of the macro `STRING_CONCAT`.

## longNumericIdsAsText

**Description**

The `longNumericIdsAsText` property specifies whether the Unica Campaign macro language will treat numeric IDs longer than 15 digits as text. This property affects ID fields. It has no effect on non-ID fields. This property is useful if you have numeric ID fields with more than 15 digits AND you want to include ID values in criteria.

- Set the value to `TRUE` to specify that numeric IDs longer than 15 digits will be treated as text.
- When the value is `FALSE`, numeric IDs longer than 15 digits are treated as numeric values (and thus might lose precision or uniqueness if truncated or rounded). If you do anything that treats the ID values as numeric (such as profiling, or using in a Derived Field), the text is converted to numeric, and precision beyond the 15th digit is lost.



**Note:** For non-ID numeric fields, if you do anything that treats the value as numeric (such as profiling, rounding, or using in a Derived Field), precision beyond 15 digits is lost.

This setting is ignored if the `partitions > partition[n] > dataSources > [data_source_name] > ForceNumeric` property is set to `TRUE` for fields coming from this data source.

#### Default value

`FALSE`

#### Valid Values

`TRUE` | `FALSE`

## stringConcatWithNullsNull

### Description

The `stringConcatWithNullsNull` property controls the behavior of the Unica Campaign macro `STRING_CONCAT`.

When the value is `TRUE`, `STRING_CONCAT` returns `NULL` if any of its inputs is `NULL`.

When the value is `FALSE`, `STRING_CONCAT` returns the concatenation of all of its non-`NULL` properties; in this case, `STRING_CONCAT` returns `NULL` only if all of its inputs are `NULL`.

#### Default value

`TRUE`

#### Valid Values

`TRUE` | `FALSE`

## performCaseInsensitiveComparisonAs

### Description

The `performCaseInsensitiveComparisonAs` property specifies how Unica Campaign compares data values when the `compareCaseSensitive` property is set to `no` (that is, during case-insensitive comparisons). This property is ignored if the value of `compareCaseSensitive` is `yes`.

When the value is `UPPER`, Unica Campaign converts all data to upper case before comparing.

When the value is `LOWER`, Unica Campaign converts all data to lower case before comparing.

#### Default value

`LOWER`

#### Valid Values

UPPER | LOWER

## upperAllowsDate

### Description

The `upperAllowsDate` property specifies whether the `UPPER` database function allows a `DATE/DATETIME` parameter, and therefore whether the operation may be performed in the database or must be performed by the Unica Campaign server.

Set the value to `TRUE` if the database is SQL Server or Oracle. The `UPPER` function in these databases allows a `DATE/DATETIME` parameter.

Set the value to `FALSE` if the database is DB2® or Teradata. The `UPPER` function in these databases does not allow a `DATE/DATETIME` parameter.

Note that this setting is global, not per data source. If a value of `no` is recommended for any data source in use, set the value to `no`. If a value of `yes` is recommended for all data sources in use, set the value to `yes`.

### Default value

TRUE

### Valid Values

TRUE | FALSE

## compareCaseSensitive

### Description

The `compareCaseSensitive` property specifies whether the Unica Campaign data comparisons are sensitive to alphabetic case (UPPER vs. lower).

When the value is `FALSE`, Unica Campaign ignores case differences when comparing data values and sorts textual data in a binary, case-insensitive manner. This setting is strongly recommended when English data is used.

When the value is `TRUE`, Unica Campaign distinguishes data values based on case differences, performing a true binary-value comparison of each character. This setting is strongly recommended when non-English data is used.

### Default value

FALSE

### Valid Values

TRUE | FALSE

## lowerAllowsDate

### Description

The `lowerAllowsDate` property specifies whether the `LOWER` database function allows a `DATE/DATETIME` parameter, and therefore whether the operation may be performed in the database or must be performed by the Unica Campaign server.

Set the value to `TRUE` if the database is SQL Server or Oracle. The `LOWER` function in these databases allows a `DATE/DATETIME` parameter.

Set the value to `FALSE` if the database is DB2® or Teradata. The `LOWER` function in these databases does not allow a `DATE/DATETIME` parameter.

Note that this setting is global, not per data source. If a value of `no` is recommended for any data source in use, set the value to `no`. If a value of `yes` is recommended for all data sources in use, set the value to `yes`. Typically, only one database type is in use at a customer site, but there are some installations in which multiple database types are in use.

### Default value

`TRUE`

### Valid Values

`TRUE` | `FALSE`

## substrAllowsDate

### Description

The `substrAllowsDate` property specifies whether the `SUBSTR/SUBSTRING` database function allows a `DATE/DATETIME` parameter, and therefore whether the operation may be performed in the database or must be performed by the Unica Campaign server.

Set the value to `TRUE` if the database is Oracle or Teradata. The `SUBSTR/SUBSTRING` function in these databases allows a `DATE/DATETIME` parameter.

Set the value to `FALSE` if the database is SQL Server or DB2®. The `SUBSTR/SUBSTRING` function in these databases does not allow a `DATE/DATETIME` parameter.

Note that this setting is global, not per data source. If a value of `no` is recommended for any data source in use, set the value to `no`. If a value of `yes` is recommended for all data sources in use, set the value to `yes`.

### Default value

`TRUE`

### Valid Values



TRUE | FALSE

## ltrimAllowsDate

### Description

The `ltrimAllowsDate` property specifies whether the `LTRIM` database function allows a `DATE/DATETIME` parameter, and therefore whether the operation may be performed in the database or must be performed by the Unica Campaign server.

Set the value to `TRUE` if the database is SQL Server, Oracle, or Teradata. The `LTRIM` function in these databases allows a `DATE/DATETIME` parameter.

Set the value to `FALSE` if the database is DB2®. The `LTRIM` function in this database does not allow a `DATE/DATETIME` parameter.

Note that this setting is global, not per data source. If a value of `no` is recommended for any data source in use, set the value to `no`. If a value of `yes` is recommended for all data sources in use, set the value to `yes`. Typically, only one database type is in use at a customer site, but there are some installations in which multiple database types are in use.

### Default value

TRUE

### Valid Values

TRUE | FALSE

## rtrimAllowsDate

### Description

The `rtrimAllowsDate` property specifies whether the `RTRIM` database function allows a `DATE/DATETIME` parameter, and therefore whether the operation may be performed in the database or must be performed by the Unica Campaign server.

Set the value to `TRUE` if the database is SQL Server, Oracle, or Teradata. The `RTRIM` function in these databases allows a `DATE/DATETIME` parameter.

Set the value to `FALSE` if the database is DB2®. The `RTRIM` function in this database does not allow a `DATE/DATETIME` parameter.

Note that this setting is global, not per data source. If a value of `no` is recommended for any data source in use, set the value to `no`. If a value of `yes` is recommended for all data sources in use, set the value to `yes`.

### Default value

TRUE

## Valid Values

TRUE | FALSE

## likeAllowsDate

### Description

The `likeAllowsDate` property specifies whether the `LIKE` database function allows a `DATE/DATETIME` parameter, and therefore whether the operation may be performed in the database or must be performed by the Unica Campaign server.

Set the value to `TRUE` if the database is SQL Server or Oracle. The `LIKE` function in these databases allows a `DATE/DATETIME` parameter.

Set the value to `FALSE` if the database is DB2® or Teradata. The `LIKE` function in these databases does not allow a `DATE/DATETIME` parameter.



**Note:** This setting is global, not per data source. If a value of `no` is recommended for any data source in use, set the value to `no`. If a value of `yes` is recommended for all data sources in use, set the value to `yes`.

### Default value

TRUE

## Valid Values

TRUE | FALSE

## fileAllSpacesIsNull

### Description

The `fileAllSpacesIsNull` property controls how Unica Campaign interprets an empty field in a mapped flat file by specifying whether an all-spaces value in a flat file should be considered to be a `NULL` value.

When the value is `TRUE`, an all-spaces value is considered to be a `NULL` value. Unica Campaign matches queries such as `<field> is null`, but fails queries such as `<field> = ""`.

When the value is `FALSE`, an all-spaces value is treated as a non-`NULL` empty string. Unica Campaign matches queries such as `<field> = ""`, but fails `<field> is null`.

### Default value

TRUE

## Valid Values

TRUE | FALSE

## Campaign | partitions | partition[n] | server | optimization

Properties in this category control Unica Campaign server optimization for each partition.



**Note:** This category is not related to Unica Optimize.

### maxVirtualMemory

#### Configuration category

Campaign|partitions|partition[n]|server|optimization

#### Description

It is used for internal locking of memory which prevents it from being swapped as temporary files.

Set a value equal to  $(80\% \times \text{available memory}) / (\text{number of expected concurrent flowcharts})$ . For example:

If available virtual memory on server = 32 GB


Number of concurrent flowcharts = 10

Set virtual Memory =  $(80\% \times 32) / 10 = \text{approximately } 2.5 \text{ GB} / \text{flowchart}$

#### Default value

128 (MB)

**maxVirtualMemory** is a global configuration setting. To override the value for a specific flowchart, open the flowchart

in Edit mode, select **Advanced settings** from the **Admin** menu , select the **Server optimization** tab, and change the **Campaign virtual memory usage** value.

### useInDbOptimization

#### Configuration category

Campaign|partitions|partition[n]|server|optimization

#### Description

This property specifies whether Unica Campaign tries to perform as many operations as possible in the database instead of in the Unica Campaign server.

Setting the value to `TRUE` can improve flowchart performance. When the value is `TRUE`, Unica Campaign avoids pulling the ID lists if possible.

When the value is `FALSE`, Unica Campaign maintains lists of IDs in the Unica Campaign server at all times.

#### Default value

FALSE

### Valid Values

TRUE | FALSE

## maxReuseThreads

### Configuration category

Campaign | partitions | partition[n] | server | optimization

### Description

This property specifies the number of operating system threads that are cached by the server process (`unica_acsvr`) for reuse. By default, the cache is disabled.

It is a best practice to use the cache when you want to reduce the overhead of thread allocation, or with operating systems that exhibit an inability to release threads when asked to do so by an application.

If the **maxReuseThreads** property is a non-zero value, set it to be greater than or equal to the value of **MaxQueryThreads**.

### Default value

0 (zero), which disables the cache

## threadStackSize

### Configuration category

Campaign | partitions | partition[n] | server | optimization

### Description

This property determines the number of bytes allocated for each thread's stack. Do not change this property except under guidance from HCL. The minimum value is 128 K. The maximum value is 8 MB.

### Default value

1048576

## tempTableDataSourcesForSegments

### Configuration category

Campaign | partitions | partition[n] | server | optimization

### Description

This property defines the list of data sources where persistent Segment temp tables can be created by the Create Seg process. This list is comma-separated. By default, this property is blank.

### Default value

No default value defined.

## doNotCreateServerBinFile

### Configuration category

`Campaign|partitions|partition[n]|server|optimization`

### Description

To improve performance, set this property to `TRUE`. When this property is `TRUE`, strategic segments create Segment temp tables in the data source rather than creating binary files on the Unica Campaign server. You must specify at least one data source in the Create Segment (CreateSeg) process configuration dialog to hold the temp tables. Also, you must set the `AllowTempTables` property to `TRUE` to enable the creation of temporary tables in a data source.

### Default value

`FALSE`

### Valid Values

`TRUE | FALSE`

## forceViewForPreOptDates

### Configuration category

`Campaign|partitions|partition[n]|server|optimization`

### Description

The default value (`TRUE`) forces creation of a parameterized offer attribute view in a Mail List process whose offers are assigned from Optimize. A value of `FALSE` causes the parameterized offer attribute view to be created only if the Mail List exports at least one parameterized offer attribute.

If this value is set to `FALSE`, a Mail List process that is configured to get its input from an Extract process (whose source is an Optimize session) may write NULL values for EffectiveDate and ExpirationDate into the `UA_Treatment` table, even when the offer includes parameterized Effective and Expiration Dates. In this case, set it back to `TRUE`.

### Default value

`TRUE`

### Valid Values

`TRUE | FALSE`

## httpCompressionForResponseLength

### Configuration category

`Campaign|partitions|partition[n]|server|optimization`

**Description**

This property enables and configures compression for HTTP responses from the Unica Campaign web application to the client browser for flowchart-specific messages. The Unica Campaign web application reads this property only once for each partition. If you modify this property, you must restart the web application for the change to take effect.

Compression can improve page load and interaction times by reducing the amount of data that is sent over HTTP.

All responses that have a data length greater than or equal to the `httpCompressionForResponseLength` value (in KB) are candidates for compression. Any other responses are not compressed.

Compression reduces network transfer, but it requires resources on the server side. Therefore, compression makes sense only for large amounts of data, when sufficient server-side resources are available. If you typically have network delays that can slow large data transfers, you can analyze how much time it takes to load a given amount of data. For example, suppose that some of your HTTP requests are <100 KB in size, but most are 300 to 500 KB. In this case, you would increase the value of this property to 500 KB so that only responses  $\geq$  500 KB in size are compressed.

To disable compression, set the value to 0.

**Default value**

100 (KB)

**Valid Values**

0 (disables compression) or higher

**cacheSystemDSQueries****Configuration category**

`Campaign|partitions|partition[n]|server|optimization`

**Description**

To improve performance, set this value to TRUE. When set to TRUE, this property reduces multiple execution of queries on the Unica Campaign system tables by caching the query results. When set to FALSE, query results are not cached.

**Default value**

TRUE

**Valid Values**

TRUE | FALSE

## Campaign | partitions | partition[n] | server | logging

Properties in this category affect flowchart logging behavior for the specified partition on the Unica Campaign server.


### enableWindowsEventLogging

#### Description

This property enables or disables Unica Campaign server logging to the Windows™ event log.

If the value is `TRUE`, logging to the Windows™ event log is enabled.

If the value is `FALSE`, logging to the Windows™ event log is disabled, and the **windowsEventLoggingLevel** and **windowsEventLoggingCategory** settings are ignored.

 **Attention:** Windows™ Event logging can cause issues with flowchart runs. Avoid enabling this feature unless advised by Technical Support.

#### Default value

`FALSE`

#### Valid Values

`TRUE` | `FALSE`

### logFileBufferSize

#### Configuration category

`Campaign|partitions|partition[n]|server|logging`

#### Description

This property is used when **keepFlowchartLogOpen** is `TRUE`. Specify a value to indicate the number of messages to buffer before writing to the log. If the value is `1`, every log message is written immediately to file, effectively disabling buffering but having a negative impact on performance.

This property is ignored if **keepFlowchartLogOpen** is `FALSE`.

#### Default value

`5`

### keepFlowchartLogOpen

#### Configuration category

`Campaign|partitions|partition[n]|server|logging`

#### Description

This property specifies whether Unica Campaign opens and closes the flowchart log file each time a line is written to the log file.

A value of `TRUE` can improve performance of real-time interactive flowcharts. When the value is `TRUE`, Unica Campaign opens the flowchart log file only once, and closes it when the flowchart's server process exits. A side effect of using the `TRUE` value is that recently-logged messages may not be immediately visible in the log file, as Unica Campaign flushes the log messages to file only when its internal buffer becomes full or when the number of logged messages equals the value of the `logFileBufferSize` property.

If the value is `FALSE`, Unica Campaign opens and closes the flowchart log file.

#### Default value

`FALSE`

#### Valid Values

`TRUE` | `FALSE`

### logProcessId

#### Configuration category

`Campaign|partitions|partition[n]|server|logging`

#### Description

This property controls whether the process ID (pid) of the Unica Campaign Server process is included in the log file.

If the value is `TRUE`, the process ID is logged.

If the value is `FALSE`, the process ID is not logged.

#### Default value

`TRUE`

#### Valid Values

`TRUE` | `FALSE`

### logMaxBackupIndex

#### Configuration category

`Campaign|partitions|partition[n]|server|logging`

#### Description

This property specifies the number of backup Unica Campaign server log files that are kept before the oldest is erased.



If the value is 0 (zero), no backup files are created, and the log file is truncated when it reaches the size specified by the `logFileMaxSize` property.

For a value of `n`, where `n` is greater than zero, the files `{File.1, ..., File.n-1}` are renamed to `{File.2, ..., File.n}`. Also, `File` is renamed `File.1` and closed. A new `File` is created to receive further log output.

#### Default value

1 (creates one backup log file)

## loggingCategories

#### Configuration category

`Campaign|partitions|partition[n]|server|logging`

#### Description

This property specifies the category of messages written to the Unica Campaign server flowchart log file. This property works in conjunction with **loggingLevels**, which determines the severity of messages to log for all selected categories.

Specify one or more categories in a comma-separated list. Use `ALL` as shorthand to indicate that you want to log all categories.

The values that you specify determine which events are logged by default for all flowcharts. Users can override the default selections by opening a flowchart for editing and choosing **Options > Logging options**. The corresponding Logging Options are indicated below in parentheses, after each configuration value.

#### Default value

`ALL`

#### Valid Values

`ALL`  
`BAD_ORDER` (Log ID ordering errors)  
`CELL_ACCESS` (Cell level operations)  
`CONFIG` (Log config settings at start of run)  
`DATA_ERRORS` (Log data conversion errors)  
`DBLOAD` (External DB loader operations)  
`FILE_ACCESS`(File operations)  
`GENERAL` (Others)  
`COMMANDS` (External interface)  
`MEMORY` (Memory allocation)  
`PROCRUN` (Process Run)  
`QUERY` (Queries issues against user tables)  
`SORT` (Log data sorting progress)

SYSQUERY (Queries issued against system tables)  
TABLE\_ACCESS (Table level operations)  
TABLE\_MAPPING (Log table mapping info at start of run )  
TABLE\_IO (Log data I/O process)  
WEBPROC (Web server interface)

## loggingLevels

### Configuration category

`Campaign|partitions|partition[n]|server|logging`

### Description

The **loggingLevels** property controls the amount of detail written to the Unica Campaign server log file, based on severity.

### Default value

MEDIUM

### Valid Values

LOW: represents the least detail (the most severe errors only)  
MEDIUM  
HIGH  
ALL: includes trace messages and is intended primarily for diagnostic purposes



**Note:** You may want to set **loggingLevels** to **ALL** during configuration and testing. This value generates a large amount of data and therefore may not be advisable for production operation. Setting any logging level higher than its default can adversely affect performance.

You can adjust these settings from within a flowchart by using **Tools > Logging options**.

## windowsEventLoggingCategories

### Configuration category

`Campaign|partitions|partition[n]|server|logging`

### Description

This property specifies the category of messages written to the Windows™ event log for the Unica Campaign server. This property works in conjunction with **windowsEventLoggingLevels**, which determines which messages are logged based on severity (for all selected categories).

You can specify multiple categories in a comma-separated list. The category `all` provides a shorthand for specifying all logging categories.

### Default value

ALL

### Valid Values

ALL  
 BAD\_ORDER  
 CELL\_ACCESS  
 CONFIG  
 DATA\_ERRORS  
 DBLOAD  
 FILE\_ACCESS  
 GENERAL  
 COMMANDS  
 MEMORY  
 PROCRUN  
 QUERY  
 SORT  
 SYSQUERY  
 TABLE\_ACCESS  
 TABLE\_MAPPING  
 TABLE\_IO  
 WEBPROC

## logFileMaxSize

### Configuration category

`Campaign|partitions|partition[n]|server|logging`

### Description

This property specifies the maximum size, in bytes, that the Unica Campaign server log file is allowed to reach before being rolled over to backup files.

### Default value

10485760 (10 MB)

## windowsEventLoggingLevels

### Configuration category

`Campaign|partitions|partition[n]|server|logging`

### Description

This property controls the amount of detail written to the Windows™ event log for the Unica Campaign server, based on severity.

### Default value

MEDIUM

### Valid Values

LOW: represents the least detail (the most severe errors only)

MEDIUM

HIGH

ALL: includes trace messages and is intended for diagnostic purposes.

## enableLogging

### Configuration category

`Campaign|partitions|partition[n]|server|logging`

### Description

This property specifies whether Unica Campaign server logging is turned on at session startup.

If the value is `TRUE`, logging is turned on.

If the value is `FALSE`, logging is turned off.

### Default value

`TRUE`

### Valid Values

`TRUE | FALSE`

## AllowCustomLogPath

### Configuration category

`Campaign|partitions|partition[n]|server|logging`

### Description

This property allows users to change the log path for each flowchart that generates flowchart-specific logging information when it is run. By default, all flowchart log files are saved in `Campaign_home/partitions/partition_name/logs`.

A setting of `TRUE` allows users to change the path through the user interface or when using `unica_svradm` to run the flowchart.

A setting of `FALSE` prevents users from changing the path to which the flowchart log file is written.

### Default value

`FALSE`

### Valid Values

TRUE | FALSE

## Campaign | partitions | partition[n] | server | flowchartRun

Properties in this category specify how many errors are allowed in a Unica Campaign Snapshot export, what files are saved when you save a flowchart, and the maximum number of IDs for each top-level process in a test run and whether output is generated during test runs.

### testRunDefaultEnabled

#### Description

The `testRunDefaultEnabled` property specifies the default value for Enable Output checkbox. Applicable while creating a new flowchart. For Old Flowcharts, Enable Output option will be whatever the user saved lastly for a particular flowchart.

#### Default value

TRUE

#### Valid Values

TRUE | FALSE

### maxDataErrorsAllowed

#### Description

The `maxDataErrorsAllowed` property specifies the maximum number of data conversion errors allowed in an Unica Campaign Snapshot export.

#### Default value

0 (zero), which allows no errors

### saveRunResults

#### Description

This property lets you save Unica Campaign flowchart run results in the temporary folder and database temp tables. You can adjust this option for individual flowcharts by using **Admin > Advanced Settings** when editing a flowchart.

For flowcharts that create artifacts that you want to save, you must set **saveRunResults** to `TRUE`. For example, if you have flowcharts that include **CreateSeg** processes, you must save run results. If you do not save run results, strategic segments do not persist.

When the value is `TRUE`, the flowchart ("underscore") files are saved and, if you are using **useInDbOptimization**, database temp tables persist.

When the value is `FALSE`, only the `.ses` file is saved. Therefore, you cannot view intermediate results if you reload the flowchart.

Unica Campaign creates many temporary files in the temporary directory, which can cause file systems to be highly utilized or even full. Setting this property to `FALSE` makes the flowchart clean up those files after run completion. However, using a setting of `FALSE` is not always feasible because it prevents you from doing partial flowchart runs.

To save disk space, you can create your own script to delete files in the temporary folder, but you must never delete files for flowcharts that are currently running. To avoid flowchart failures, never delete any files from the temporary folder that are being updated or created today. For maintenance purposes, you can delete files from the temp folder if they are older than 2 days.

#### **Default value**

`TRUE`

#### **Valid Values**

`TRUE` | `FALSE`

### **testRunDefaultSize**

#### **Description**

The `testRunDefaultSize` property specifies the default maximum number of IDs for each top-level process in a Unica Campaign test run. A value of `0` (zero) removes the limitation on the number of IDs.

#### **Default value**

`0` (zero)

### **enableEmailCommunication**

#### **Description**

The `enableEmailCommunication` property specifies whether the unica campaign will send email notifications to users about flowchart statuses.

#### **Default value**

`TRUE`

#### **Valid Values**

`TRUE` | `FALSE`

### **eventForSendingEmail**

#### **Description**

The `eventForSendingEmail` property allows users to specify the specific flowchart statuses for which they want to receive email notifications.

#### **Default value**

ALL

**Valid Values**

ALL | FINISH | FAILURE | START | PAUSE | CONTINUE | STOP

Campaign | partitions | partition[n] | server | profile

Properties in this category specify the maximum number of categories that are created during profiling for numeric and text values in Unica Campaign.

**profileMaxTextCategories****Description**

The `profileMaxTextCategories` and `profileMaxNumberCategories` properties specify the maximum number of categories created in Unica Campaign during profiling for text and numeric values, respectively.

These values are different from the setting for the number of bins displayed to the user, which can be modified through the user interface.

**Default value**

1048576

**profileMaxNumberCategories****Description**

The `profileMaxNumberCategories` and `profileMaxTextCategories` properties specify the maximum number of categories created in Unica Campaign during profiling for numeric and text values, respectively.

These values are different from the setting for the number of bins displayed to the user, which can be modified through the user interface.

**Default value**

1024

Campaign | partitions | partition[n] | server | internal

Properties in this category specify integration settings and the internalID limits for the selected Unica Campaign partition. If your Unica Campaign installation has multiple partitions, set these properties for each partition that you want to affect.

**internalIdLowerLimit****Configuration category**

Campaign | partitions | partition[n] | server | internal

**Description**

The `internalIdUpperLimit` and `internalIdLowerLimit` properties constrain the Unica Campaign internal IDs to be within the specified range. Note that the values are inclusive: that is, Unica Campaign may use both the lower and upper limit.

**Default value**

0 (zero)

**internalIdUpperLimit**

**Configuration category**

`Campaign|partitions|partition[n]|server|internal`

**Description**

The `internalIdUpperLimit` and `internalIdLowerLimit` properties constrain the Unica Campaign internal IDs to be within the specified range. The values are inclusive: that is, Unica Campaign may use both the lower and upper limit. If Unica Collaborate is installed, set the value to `2147483647`.

**Default value**

4294967295

**deliverInstalled**

**Configuration category**

`Campaign|partitions|partition[n]|server|internal`

**Description**

Indicates that Unica Deliver is installed. When you select `yes`, Unica Deliver features are available in the Unica Campaign interface.

The installer sets this property to `yes` for the default partition in your Unica Deliver installation. For additional partitions where you installed Unica Deliver, you must configure this property manually.

**Default value**

No

**Valid Values**

Yes | No

**interactInstalled**

**Configuration category**

`Campaign|partitions|partition[n]|server|internal`

**Description**



After installing the Unica Interact design environment, this configuration property should be set to `Yes` to enable the Unica Interact design environment in Unica Campaign.

If Unica Interact is not installed, set to `No`. Setting this property to `No` does not remove Unica Interact menus and options from the user interface. To remove menus and options, you must manually unregister Unica Interact using the `configTool` utility.

**Default value**

no

**Valid Values**

yes | no

**Availability**

This property is applicable only if you installed Unica Interact.

## MO\_UC\_integration

**Configuration category**

`Campaign|partitions|partition[n]|server|internal`

**Description**

Enables integration with Unica Plan for this partition, if the integration is enabled in the **Platform** configuration settings. For more information, see the Unica Plan and Unica Campaign Integration Guide.

**Default value**

No

**Valid Values**

Yes | No

## MO\_UC\_BottomUpTargetCells

**Configuration category**

`Campaign|partitions|partition[n]|server|internal`

**Description**

For this partition, allows bottom-up cells for Target Cell Spreadsheets, if **MO\_UC\_integration** is enabled. When set to `Yes`, both top-down and bottom-up target cells are visible, but bottom-up target cells are read-only. For more information, see the Unica Plan and Unica Campaign Integration Guide.

**Default value**

No

**Valid Values**

Yes | No

## Legacy\_campaigns

### Configuration category

Campaign|partitions|partition[n]|server|internal

### Description

For this partition, enables access to campaigns created before Unica Plan and Unica Campaign were integrated. Applies only if **MO\_UC\_integration** is set to `yes`. Legacy campaigns also include campaigns created in Unica Campaign 7.x and linked to Plan 7.x projects. For more information, see the Unica Plan and Unica Campaign Integration Guide.

### Default value

No

### Valid Values

Yes | No

## Unica Plan - Offer integration

### Configuration category

Campaign|partitions|partition[n]|server|internal

### Description

Enables the ability to use Unica Plan to perform offer lifecycle management tasks on this partition, if **MO\_UC\_integration** is enabled for this partition. Offer integration must be enabled in your **Platform** configuration settings. For more information, see the Unica Plan and Unica Campaign Integration Guide.

### Default value

No

### Valid Values

Yes | No

## UC\_CM\_integration

### Configuration category

Campaign|partitions|partition[n]|server|internal

### Description

Enables Digital Analytics online segment integration for a Unica Campaign partition. If you set this value to `yes`, the Select process box in a flowchart provides the option to select **Digital Analytics Segments** as input. To configure the Digital Analytics integration for each partition, choose **Settings > Configuration > Unica Campaign | partitions | partition[n] | Coremetrics**.

**Default value**

No

**Valid Values**

Yes | No

**linkInstalled****Configuration category**`Campaign|partitions|partition[n]|server|internal`**Description**

Indicates that Link is installed. When you select Yes, Manage Link Connections feature is available in the Unica Campaign interface. The installer sets this property to No for the default partition in your Link installation. For additional partitions where you installed Link, you must configure this property manually.

**Default value**

No

**Valid Values**

Yes | No

**numRowsReadToParseDelimitedFile****Configuration category**`Campaign|partitions|partition[n]|server|internal`**Description**

This property is used when mapping a delimited file as a user table. It is also used by the Score process box when importing a score output file from IBM SPSS Modeler Advantage Enterprise Marketing Management Edition. To import or map a delimited file, Unica Campaign needs to parse the file to identify the columns, data types (field types), and column widths (field lengths).

The default value of 100 means Unica Campaign examines the first 50 and the last 50 line entries in the delimited file. Unica Campaign then allocates the field length based on the largest value it finds within those entries. In most cases, the default value is sufficient to determine field lengths. However, in very large delimited files, a later field might exceed the estimated length that Unica Campaign computes, which can cause an error during flowchart runtime. Therefore, if you are mapping a very large file, you can increase this value to make Unica Campaign examine more line entries. For example, a value of 200 makes Unica Campaign examine the first 100 line entries and the last 100 line entries of the file.

A value of 0 examines the entire file. Typically, this is necessary only if you are importing or mapping files that have variable data widths of fields which cannot be identified by reading the first and last few

lines. Reading the entire file for extremely large files can increase the required processing time for table mapping and Score process box runs.

**Default value**

100

**Valid Values**

0 (all lines) or any positive integer

**contactCentralEnabled**

**Configuration category**

Campaign | Affinium | Campaign | partitions | <partitionName> | server | internal

**Description**

This flag control the display of Regional Preference field on Campaign Create/Edit/Display summary screen.

**Default value**

NO

**Valid Values**

Yes - Contact Central Integration is enabled in Campaign

NO - Contact Central Integration is disabled in Campaign

Campaign | partitions | partition[n] | server | fileDialog

Properties in this category specify the default directories for Unica Campaign input and output data files.

**defaultOutputDirectory**

**Description**

The `defaultOutputDirectory` property specifies the path used to initialize the Unica Campaign File Selection dialog. The `defaultOutputDirectory` property is used when an output data file is mapped into Unica Campaign. If no value is specified, the path is read from the environment variable `UNICA_ACDFDIR`.

**Default value**

No default value defined.

**defaultInputDirectory**

**Description**

The `defaultInputDirectory` property specifies the path used to initialize the Unica Campaign File Selection dialog. The `defaultInputDirectory` property is used when an input data file is mapped into Unica Campaign. If no value is specified, the path is read from the environment variable `UNICA_ACDFDIR`.

**Default value**

No default value defined.

**Campaign | partitions | partition[n] | offerCodeGenerator**

Properties in this category specify the class, classpath, and configuration string for the offer code generator, and also the cell code generator used to assign a contact process to a Target Cell Spreadsheet cell.

**offerCodeGeneratorClass****Description**

The `offerCodeGeneratorClass` property specifies the name of the class Unica Campaign uses as its offer code generator. The class must be fully qualified with its package name.

**Default value**

Note that line breaks have been added for print.

```
com.unica.campaign.core.codegenerator.samples.ExecutableCodeGenerator
```

**offerCodeGeneratorConfigString****Description**

The `offerCodeGeneratorConfigString` property specifies a string that is passed into the offer code generator plug-in when it is loaded by Unica Campaign. By default, the `ExecutableCodeGenerator` (shipped with Unica Campaign) uses this property to indicate the path (relative to Unica Campaign application home directory) to the executable to run.

**Default value**

```
./bin
```

**defaultGenerator****Description**

The `defaultGenerator` property specifies the generator for the cell codes that appear in contact-style process boxes and are used to assign cells to Target Control Spreadsheet cells. The Target Control Spreadsheet manages cell and offer mappings for campaigns and flowcharts.

**Default value**

```
uacoffercodegen.exe
```

**offerCodeGeneratorClasspath****Description**

The `offerCodeGeneratorClasspath` property specifies the path to the class Unica Campaign uses as its offer code generator. It can be either a full path or a relative path.

If the path ends in a slash (forward slash `/` for UNIX™ or backslash `\` for Windows™), Unica Campaign assumes it to be a path to a directory that contains the Java™ plug-in class that should be used. If the path does not end in a slash, Unica Campaign assumes it is the name of a `jar` file that contains the Java™ class.

If the path is relative, Unica Campaign assumes it is relative to the Unica Campaign application home directory.

#### Default value

`codeGenerator.jar` (packaged in the `Campaign.war` file)

## Campaign | partitions | partition[n] | UBX

These properties control authentication and data exchange between Unica Campaign, Engage, and UBX if the products are integrated.

To access these properties, choose **Settings > Configuration**. If your Unica Campaign installation has multiple partitions, set these properties for each partition that uses the integration.

### API URL

#### Description

Specify the UBX Server API URL.

### Data Source for UBX Endpoint Authorization key

#### Description

Specify the datasource name that contains the authorization key for the Unica Campaign registered endpoint. For example, `UBX_DS`.

### Platform User with Data Sources for UBX Credentials

#### Description

Specify the Platform user name that contains the datasource with the name specified under the configuration property **Data Source for UBX Endpoint Authorization key**.

### Use proxy for API URL

#### Description

Determine if you want to use a proxy server for the UBX connection. If you select `yes`, the proxy server details are configured under `Campaign | proxy`.

## Campaign | partitions | partition[n] | UBX | Event Download Schedule

These properties specify the schedule for when events are downloaded from UBX into Unica Campaign.

To access these properties, choose **Settings > Configuration**. If your Unica Campaign installation has multiple partitions, set these properties for each partition that uses the integration.

### Event Download Enabled

#### Description

Determine if you want to enable events from UBX to download to the events table in the Unica Campaign system schema.

#### Default value

No

#### Valid values

Yes, No

### runOnceADay

#### Description

Determine if the download should occur on a daily basis. It can run repeatedly if you specify the sleepIntervallnMinutes property.

### sleepIntervallnMinutes

#### Description

Specify the number of minutes the download waits before executing again. This value is used when runOnceADay is set to No.

### startTime

#### Description

When runOnceADay is set to Yes, this property determines when the event download starts.

## Campaign|partitions|partition[n]|Kafka|Journey

This category contains properties to configure the kafka used in Journeys for the selected partition

### KafkaBrokerURL

#### Description

Kafka server being used with Journeys application. Example is IP-0A862D46:9092

### CommunicationMechanism

#### Description

Specify the connection mechanism to connect to Kafka server.

**Possible Values:**

SASL\_PLAINTEXT\_SSL - Use this to connect to kafka with username/password and SSL enabled.

NO\_SASL\_PLAINTEXT\_SSL - Use this to connect kafka with no authentication and no SSL.

SASL\_PLAINTEXT - Use this to connect kafka with username and password only.

SSL - Use this to connect kafka without username/password but with SSL.

**Default value:**

NO\_SASL\_PLAINTEXT\_SSL

### **KafkaKeyFile**

**Description**

Specify the client key file if connection mechanism is using SSL. Example: /opt/Unica/Kafkakeys/client\_key.pem

### **KafkaCertificateFile**

**Description**

Specify the certificate file if connection mechanism is using SSL. Example: /opt/Unica/Kafkakeys/client\_cert.pem

### **CertificateAuthorityFile**

**Description**

It is signed certificate of Kafka Server, it is required when connection mechanism is using SSL. Example - /opt/Unica/Kafkakeys/ca-cert

### **UserForKafkaDataSource**

**Description**

Marketing Platform user contains the datasource credentials for Kafka while connecting with username / password

### **KafkaDataSource**

**Description**

DataSource containing the kafka user credentials.

### **TopicName**

**Description**

Journeys designated topic for Campaign to push data to Journey.



**Default value:**

STREAMING\_IMPORT

**Note:** Please do not change this as it would send data to Kafka topic which is not being used in Journey.

## Campaign | partitions | partition[n] | Link

This category contains properties to configure the Link for the selected partition.

### LinkConnectionURL

**Description**

This configuration parameter is used to specify the Link Connections url. Campaign application would be using this URL to retrieve connections / action details.

### Application

**Description**

Application name which is being used with Link. For example "campaign" when we are using it with Unica Campaign

### AsmUserForLinkCredentials

**Description**

Marketing Platform user which stores the Unica link credentials data source

### DataSourceWithLinkCredentials

**Description**

Data source name which holds the username and password to authenticate with LinkConnectionURL

### LinkProjectName

**Description**

Campaign project name configured in Unica Link. This project will contain the Campaign configurations like authentication, field mapping url etc.

### Application

**Description**

This configuration parameter is used to specify the Link Connections url. Campaign application would be using this URL to retrieve connections / action details.

### LinkSharedLocation

**Description**

Location on Campaign Listener machine – which should be mounted on the HIP REST server on path mentioned under “HIPRestServerSharedLocation”

## LinkRuntimeServerURL

### Description

Link runtime (HIP REST) server url.

## MultiPartition Support for Link

From V12.1.7 onwards, Campaign supports Multi Partition for Link.

### Enabling MultiPartition

To leverage this new feature, it's essential to enable it using a JVM property. Add the following flag when starting your application:

**-DisLinkMultiPartitionEnabled=true**

## Campaign | monitoring

Properties in the this category specify whether the Operational Monitoring feature is enabled, the URL of the Operational Monitoring server, and caching behavior. Operational Monitoring displays and allows you to control active flowcharts.

### cacheCleanupInterval

#### Description

The `cacheCleanupInterval` property specifies the interval, in seconds, between automatic cleanups of the flowchart status cache.

This property is not available in versions of Unica Campaign earlier than 7.0.

#### Default value

600 (10 minutes)

### cacheRunCompleteTime

#### Description

The `cacheRunCompleteTime` property specifies the amount of time, in minutes, that completed runs are cached and display on the Monitoring page.

This property is not available in versions of Unica Campaign earlier than 7.0.

#### Default value

4320

## monitorEnabled

### Description

The `monitorEnabled` property specifies whether the monitor is turned on.

This property is not available in versions of Unica Campaign earlier than 7.0.

### Default value

`FALSE`

### Valid values

`TRUE` | `FALSE`

## serverURL

### Description

The `Campaign > monitoring > serverURL` property specifies the URL of the Operational Monitoring server. This is a mandatory setting; modify the value if the Operational Monitoring server URL is not the default.

If Unica Campaign is configured to use Secure Sockets Layer (SSL) communications, set the value of this property to use HTTPS. For example: `serverURL=https://host:SSL_port/Campaign/OperationMonitor` where:

- `host` is the name or IP address of the machine on which the web application is installed
- `SSL_Port` is the SSL port of the web application.

Note the `https` in the URL.

### Default value

`http://localhost:7001/Campaign/OperationMonitor`

## monitorEnabledForInteract

### Description

If set to `TRUE`, enables Unica Campaign JMX connector server for Unica Interact. Unica Campaign has no JMX security.

If set to `FALSE`, you cannot connect to the Unica Campaign JMX connector server.

This JMX monitoring is for the Unica Interact contact and response history module only.

### Default value

`FALSE`

### Valid Values

TRUE | FALSE

### Availability

This property is applicable only if you have installed Unica Interact.

## protocol

### Description

Listening protocol for the Unica Campaign JMX connector server, if `monitorEnabledForInteract` is set to yes.

This JMX monitoring is for the Unica Interact contact and response history module only.

### Default value

JMXMP

### Valid Values

JMXMP | RMI

### Availability

This property is applicable only if you have installed Unica Interact.

## port

### Description

Listening port for the Unica Campaign JMX connector server, if `monitorEnabledForInteract` is set to yes.

This JMX monitoring is for the Unica Interact contact and response history module only.

### Default value

2004

### Valid Values

An integer between 1025 and 65535.

### Availability

This property is applicable only if you have installed Unica Interact.

## Campaign | ProductReindex

The creator of an offer can specify the products that are associated with that offer. When the list of products available for association with offers changes, the offer/product associations must be updated. Properties in the Unica Campaign > ProductReindex category specify the frequency of these updates and the time of day that the first update runs.

## startTime

### Description

The `startTime` property specifies the time of day when offer/product associations are updated for the first time. The first update occurs on the day after the Unica Campaign server is started, and subsequent updates occur at intervals specified in the `interval` parameter. The format is `HH:mm:ss`, using a 24-hour clock.

Note that when Unica Campaign first starts up, the `startTime` property is used according to the following rules:

- If the time of day specified by `startTime` is in the future, the first offer/product associations update will occur at `startTime` of the current day.
- If `startTime` is in the past for the current day, the first update will occur at `startTime` tomorrow, or at `interval` minutes from the current time, whichever is earlier.

### Default value

12:00:00 (noon)

## interval

### Description

The `interval` property specifies the time, in minutes, between updates of offer/product associations. The update occurs for the first time at the time specified in the `startTime` parameter, on the day after the Unica Campaign server is started.

### Default value

3600 (60 hours)

## Campaign | unicaACLlistener

If you are configuring a single-node listener cluster, use this category, and only this category, to define configuration settings for your non-clustered listener. For clustered listeners, the properties in this category pertain to all listener nodes in the cluster, except for the following properties, which are ignored: `serverHost`, `serverPort`, `useSSLForPort2`, `serverPort2`. (Instead, set these properties for each individual node, under `Campaign|unicaACLlistener|node[n]`.)

These properties must be set only once per instance of Unica Campaign, they do not need to be set for every partition.

## enableWindowsImpersonation

### Configuration category

`Campaign|unicaACLlistener`

### Description

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `enableWindowsImpersonation` property specifies whether Windows™ impersonation is enabled in Unica Campaign.

Set the value to `TRUE` if you are using Windows™ impersonation. You must configure Windows™ impersonation separately if you want to leverage the Windows™-level security permissions for file access.

Set the value to `FALSE` if you are not using Windows™ impersonation.

### Default value

`FALSE`

### Valid Values

`TRUE` | `FALSE`

## enableWindowsEventLogging

### Configuration category

`Campaign|unicaACLlistener`

### Description

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `Campaign | unicaACLlistener | enableWindowsEventLogging` property turns Windows™ event logging on or off for Unica Campaign listener events. Set this property to `TRUE` to log to the Windows™ event log.



**Attention:** Windows™ Event logging can cause issues with flowchart runs. Avoid enabling this feature unless advised by Technical Support.

### Default value

`FALSE`

### Valid Values

`TRUE` | `FALSE`

## serverHost

### Configuration category

`Campaign|unicaACLlistener`

**Description**

If you have a single-node listener configuration, this property identifies the listener. If you have a clustered listener configuration, this property is ignored. (Instead, set this property for each individual node, under Campaign|unicaACLlistener|node[n].)

The `serverHost` property specifies the name or IP address of the machine where the Unica Campaign listener is installed. If the Unica Campaign listener is not installed on the same machine where Unica is installed, change the value to the machine name or IP address of the machine where the Unica Campaign listener is installed.

**Default value**

localhost

**logMaxBackupIndex****Configuration category**

Campaign|unicaACLlistener

**Description**

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `logMaxBackupIndex` property specifies how many backup files can exist before the oldest one is deleted. If you set this property to 0 (zero), Unica Campaign does not create any backup files and the log file stops logging when it reaches the size you specified in the `logMaxFileSize` property.

If you specify a number (`N`) for this property, when the log file (`File`) reaches the size you specified in the `logMaxFileSize` property, Unica Campaign renames the existing backup files (`File.1 ... File.N-1`) to `File.2 ... File.N`, renames the current log file `File.1`, closes it, and starts a new log file named `File`.

**Default value**

1 (creates one backup file)

**logStringEncoding****Configuration category**


Campaign|unicaACLlistener

**Description**

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `logStringEncoding` property controls the encoding used for all log files. This value must match the encoding used on the operating system. For multi-locale environments, UTF-8 is the preferred setting.

If you change this value, you should empty or remove all affected log files to prevent writing multiple encodings into a single file.

 **Note:** `WIDEUTF-8` is not supported for this setting.

#### Default value

`native`

#### Valid Values

See "Character encodings in Unica Campaign" in the Unica Campaign Administrator's Guide.

### systemStringEncoding

#### Configuration category

`Campaign|unicaACLlistener`


#### Description

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `systemStringEncoding` property indicates which encodings Unica Campaign uses to interpret values received from and sent to the operating system, such as file system paths and filenames. In most cases, you can set this value to `native`. For multi-locale environments, use `UTF-8`.

You can specify more than one encoding, separated by commas. For example:

`UTF-8,ISO-8859,CP950`

 **Note:** `WIDEUTF-8` is not supported for this setting.

#### Default value

`native`

#### Valid Values

See Character encodings in Unica Campaign in the Unica Campaign Administrator's Guide.

### loggingLevels

#### Configuration category

`Campaign|unicaACLlistener`

#### Description



This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `Campaign > unicaACLlistener > loggingLevels` property controls the amount of detail written to the log file.

This property applies to both clustered and non-clustered configurations.

#### Default value

MEDIUM

#### Valid Values

- LOW
- MEDIUM
- HIGH

### maxReuseThreads

#### Configuration category

`Campaign|unicaACLlistener`

#### Description

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

This property sets the number of operating system threads cached by the Unica Campaign listener process (`unica_aclsnr`) for reuse.

It is a best practice to use the cache when you want to reduce the overhead of thread allocation, or with operating systems that can exhibit an inability to release threads when asked to do so by an application.

This property applies to both clustered and non-clustered configurations.

#### Default value

0 (zero), which disables the cache

### logMaxFileSize

#### Configuration category

`Campaign|unicaACLlistener`

#### Description

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `logMaxFileSize` property specifies the maximum size, in bytes, that the log file can reach before rolling into the backup file.

This property applies to both clustered and non-clustered configurations.

**Default value**

10485760 (10 MB)

**windowsEventLoggingLevels**

**Configuration category**

Campaign|unicaACLlistener

**Description**

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `windowsEventLoggingLevels` property controls the amount of detail written to the Windows™ event log file based on severity.

This property applies to both clustered and non-clustered configurations.

**Default value**

MEDIUM

**Valid Values**

- LOW
- MEDIUM
- HIGH
- ALL

The `ALL` level includes trace messages intended for diagnostic purposes.

**serverPort**

**Configuration category**

Campaign|unicaACLlistener

**Description**

If you have a single-node listener configuration, this property identifies the listener port. If you have a clustered listener configuration, this property is ignored. (Instead, set this property for each individual node, under Campaign|unicaACLlistener|node[n].)

The `serverPort` property specifies the port where a single (non-clustered) Unica Campaign listener is installed.

**Default value**

4664

**useSSL****Configuration category**

Campaign|unicaACLlistener

**Description**

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

The `useSSL` property specifies whether to use Secure Sockets Layer for communications between the Unica Campaign listener and the Unica Campaign web application.

Also see the description for the `serverPort2` property, in this category.

**Default value**

no

**Valid Values**

yes | no

**serverPort2****Configuration category**

Campaign|unicaACLlistener

**Description**

This property is optional.

This property applies only to a single-node listener configuration. If you have a clustered listener configuration, this property is ignored. (Instead, define `serverPort2` for each individual node, under Campaign|unicaACLlistener|node[n].)

The `serverPort2` property, in conjunction with the `useSSLForPort2` property, also in this category, enables you to specify the use of SSL for communication between the Unica Campaign listener and flowchart processes, separately from the communication between the Unica Campaign web application and listener, which is specified by the `serverPort` and `useSSL` properties in this category.

All communication between Unica Campaign components, (between the web application and listener and between the listener and server) use the mode specified by the `useSSL` property under any of the following conditions.

- `serverPort2` is set to its default value of 0, **or**
- `serverPort2` is set to the same value as `serverPort`, **or**
- `useSSLForPort2` is set to the same value as `useSSL`

In these cases, a second listener port is not enabled, and communication between the Unica Campaign listener and the flowchart (server) processes and communication between the listener and the Unica Campaign web application use the same mode: either both non-SSL or both SSL, depending on the value of the `useSSL` property.

The listener uses two different modes of communication when both of the following conditions exist.

- `serverPort2` is set to a non-0 value different from the value of `serverPort`, **and**
- `useSSLForPort2` is set to a value different from the value of `useSSL`

In this case, a second listener port is enabled, and the listener and flowchart processes use the mode of communication specified by `useSSLForPort2`.

The Unica Campaign web application always uses the communication mode specified by `useSSL` when communicating to the listener.

When SSL is enabled for communication between the Unica Campaign listener and flowchart processes, set the value of this property (`serverPort2`) to an appropriate port.

#### Default value

0

## useSSLForPort2

#### Configuration category

Campaign|unicaACLlistener

#### Description

This property applies only to a single-node listener configuration. If you have a clustered listener configuration, this property is ignored. (Instead, define `useSSLForPort2` for each individual node, under Campaign|unicaACLlistener|node[n].)

For information, see the description for `serverPort2`, in this category.

#### Default value

FALSE

#### Valid Values

TRUE | FALSE

## keepalive

### Configuration category

Campaign | unicaACLlistener

### Description

This property applies to both clustered and non-clustered listener configurations. For a clustered configuration, this property applies to all listener nodes in the cluster.

Use the `keepalive` property to specify, in seconds, the frequency with which the Unica Campaign web application server sends keep alive messages on otherwise-inactive socket connections to the Unica Campaign listener.

Using the `keepalive` configuration parameter enables socket connections to remain open through extended periods of application inactivity in environments configured to close inactive connections between the web application and the listener (for example, a firewall).

When there is activity on a socket, the keep alive period is automatically reset. At the DEBUG logging level in the web application server, the `campaignweb.log` will show evidence of the keep alive messages as they are sent to the listener.

### Default value

0, which disables the keepalive feature

### Valid Values

positive integers

## loggingCategories

### Configuration category

Campaign | unicaACLlistener

### Description

This property specifies the category of messages written to the Unica Campaign listener log file.

The `loggingCategories` property in conjunction with the `loggingLevels` property determines the severity of the messages that are logged for all selected categories.

Specify one or more categories in a comma separated list. Use the `ALL` option to include logs for all categories.

### Default value

ALL

### Valid Values



**Note:** The corresponding Logging Options are indicated in parentheses after each configuration value.

ALL

GENERAL (Others)

COMMANDS (External interface)

SYS CALL (System Call)

UDB (udb)

XML (xml)

## Campaign | unicaACLlistener | node [n]

A non-clustered listener configuration should not have any nodes under this category. Nodes are created and used only for clustered listener configurations. For a clustered listener configuration, configure an individual child node for each listener in the cluster.

If clustering is enabled, you must configure at least one child node or an error occurs during startup.



**Important:** Never remove a node from the configuration unless you first stop all of the clustered listener nodes. Otherwise, any existing sessions on the removed listener continue to run but the master listener will not be able to contact the removed listener node. This can cause unexpected results.

### serverHost

#### Configuration category

Campaign | unicaACLlistener | node[n]

#### Description

This property applies only if you have a clustered listener configuration. This property identifies each individual listener node in a cluster.

For each node, specify the hostname of the machine where the Unica Campaign listener is installed.

#### Default value

No default value assigned.

### serverPort

#### Configuration category

Campaign | unicaACLlistener | node[n]

#### Description

This property applies only if you have a clustered listener configuration. This property identifies the port that is used for communication between each clustered listener node and the Unica Campaign web application server.

The specified port is also used for communication between listener nodes.

#### Default value

No default value assigned.

### useSSLForPort2

#### Configuration category

`Campaign|unicaACLlistener|node[n]`

#### Description

Optional. This property applies only if you have a clustered listener configuration. You can set this property for each clustered listener node. For information about how to use this property, read the description for `Campaign|unicaACLlistener|serverPort2`.

#### Default value

FALSE

#### Valid Values

TRUE|FALSE

### serverPort2

#### Configuration category

`Campaign|unicaACLlistener|node[n]`

#### Description

Optional. This property applies only if you have a clustered listener configuration. You can set this property for each clustered listener node. For information about how to use this property, read the description for `Campaign|unicaACLlistener|serverPort2`.

#### Default value

3

### masterListenerPriority

#### Configuration category

`Campaign|unicaACLlistener|node[n]`

#### Description

This property applies only if you have a clustered listener configuration.

A cluster always includes one master listener. All clients, including the Unica Campaign web server application, the Unica Campaign Server Manager (`unica_svradm`), and utilities such as `unica_acsesutil`, use the `masterListenerPriority` to identify the master listener.

Any node in the cluster can act as the master listener. The `masterListenerPriority` determines which node initially acts as the master listener. It also determines which listener will take over as the master listener in failover situations. Ideally, the listener node with the most processing power should be assigned the highest priority.

Priority 1 is the highest priority. Assign a 1 to the machine that you want to be the master listener. That machine will serve as the master listener unless it goes down or cannot be contacted, for example due to a network issue. Assign a 2 to the next machine, and so on.

You must assign a priority to every listener in the cluster. If you do not want a machine to serve as the master listener, assign it the lowest priority (10). However, you cannot prohibit a listener from being designated as master. In a clustered listener configuration, one listener must always serve as the master.

If the designated master listener cannot be contacted, then the next machine becomes the master listener, based on its assigned priority.

If multiple nodes have the same priority, the system selects the first of those nodes from the list of nodes in this category.



**Note:** After you change the priority, run the `unica_svradm refresh` command to inform the master listener of the change.

#### Default value

No default value assigned.

#### Valid Values

1 (high) through 10 (low)

## loadBalanceWeight

#### Configuration category

`Campaign|unicaACLlistener|node[n]`

#### Description

This property applies only if you have a clustered listener configuration. This property controls load balancing among clustered nodes. Each node in a cluster can process a portion of the total application traffic. Adjust the weight for each listener node to determine how much load the node will get. A higher value assigns a greater proportion of the load, so more transactions are given to that listener node.



Assign higher values to machines that have more processing capacity. Assign lower values to less powerful or more heavily loaded machines. A value of 0 prohibits the listener from processing any transactions and typically is not used. If multiple nodes have the same weight, the system selects the first of those nodes from the list of nodes in this category.



**Note:** After you change the weight, run the `unica_svradm refresh` command to inform the master listener of the change.

### Example

You have four physical hosts: A, B, C and D.

Host A is the most powerful machine and Host D is the least capable, so you assign weights as: A=4, B=3, C=2, D=1. In terms of loads handled, listeners will process loads in these proportions: A - 40%, B - 30%, C - 20%, D - 10%. For each incoming request to handle Campaign flowchart or Optimize session executions, the master listener will try to see that the listeners have loads in above proportion. For example, if after calculating total load across all the listeners A is having 30% of load then the next request will go to A.

### Default value

No default value assigned.

### Valid Values

0 through 10 (highest priority)

### ListenerType

### Configuration Category

Campaign | unicaACLListener | node[n]

### Description

This property applies, if you have a clustered listener configuration. This property controls execution of flowchart and optimize session among clustered nodes. Each node in a cluster can execute either flowchart or optimize session or both. Choose a type for each listener node to determine what a node can execute.

Value-1 shows that this particular node executes only flowcharts. No Optimize session execution request can be redirected to this type of listener node.

Value-2 shows that this particular node only executes Optimize sessions. No flowchart execution request can be redirected to this type of listener node.

Value-3 shows that this particular node executes both flowcharts and optimize sessions. So any request can be redirected to this type of node.



**Note:** After you change the listenerType, run the unica\_svradm refresh command to inform the master listener of the change.

#### Default value

No default value assigned

#### Valid value

1|2|3

## Campaign | campaignClustering

Set these properties if you have a clustered listener configuration. Set these properties once per instance of Unica Campaign; you do not need to set them for every partition.

### enableClustering

#### Configuration category

Campaign|campaignClustering

#### Description

If you have a single listener, leave the value set to `FALSE`. This causes all other properties in this category to be ignored, because they do not apply to a single-node configuration.

For a clustered listener configuration, set the value to `TRUE`, configure the other properties in this category, then configure listener nodes under `Campaign|unicaACLlistener|node[n]`. When the value is `TRUE`, you must define at least one child node. If you do not define at least one child node, an error occurs during startup.

When the value is `TRUE`, the following properties are ignored for `Campaign|unicaACLlistener` and are instead defined for each individual node under `Campaign|unicaACLlistener|node[n]`: `serverHost`, `serverPort`, `serverPort2`, `useSSLForPort2`.

#### Default value

`FALSE`

#### Valid Values

`TRUE` | `FALSE`

### masterListenerLoggingLevel

#### Configuration category

Campaign|campaignClustering

#### Description

This property applies only if `enableClustering` is TRUE. This property controls the amount of detail that is written to the master listener log file (`<campaignSharedHome>/logs/masterlistener.log`).

The default value of LOW provides the least detail (only the most severe error messages are written). ALL includes trace debug messages and is intended for diagnostic purposes.

**Default value**

MEDIUM

**Valid Values**

LOW | MEDIUM | HIGH | ALL

**masterListenerHeartbeatInterval****Configuration category**

Campaign | campaignClustering

**Description**

This property applies only if `enableClustering` is TRUE. This property affects the master listener. Specify how often the master listener tries to connect to all configured listener nodes to identify their availability. When the master listener connects to the nodes for availability, it also sends a heartbeat message to inform them that the master listener is alive. Therefore, this property serves two purposes (1) Heartbeat from master listener (2) Status response from each listener node.

**Default value**

10 seconds

**webServerDelayBetweenRetries****Configuration category**

Campaign | campaignClustering

**Description**

This property applies only if `enableClustering` is TRUE. This property specifies the time delay between retries for the Unica Campaign web application server to attempt to connect to the Unica Campaign listener.

**Default value**

5 seconds

**webServerRetryAttempts****Configuration category**

Campaign | campaignClustering

**Description**

This property applies only if `enableClustering` is TRUE. This property indicates how many times the Unica Campaign web application server attempts to connect to the Unica Campaign listener.

#### Default value

3

## campaignSharedHome

#### Configuration category

Campaign | campaignClustering

#### Description

This property applies only if `enableClustering` is TRUE.

In a clustered configuration, listener nodes share the files and folders indicated below. The shared location is specified at installation time.



**Note:** Ensure that you do not end the `sharedHome` path with `"/`.

```
campaignSharedHome
|--->/conf
|-----> activeSessions.udb
|-----> deadSessions.udb
|-----> etc.
|--->/logs
|-----> masterlistener.log
|-----> etc.
|--->/partitions
|-----> partition[n]
|-----> {similar to <Campaign_home> partition folder structure}
```



**Note:** Each listener also has its own set of folders and files that are not shared, at `<Campaign_home>` (the Unica Campaign application installation directory).

## masterListenerLoggingCategories

#### Configuration category

Campaign | campaignClustering

#### Description

This property specifies the category of messages written to the Unica Campaign master listener log file.

The `masterListenerLoggingCategories` property in conjunction with the `masterListenerLoggingLevel` property determines the severity of the messages that are logged for all selected categories.

Specify one or more categories in a comma separated list. Use the `ALL` option to include logs for all categories.

#### Default value

`ALL`

#### Valid Values



**Note:** The corresponding Logging Options are indicated in parentheses after each configuration value.

`ALL`

`FILE_ACCESS` (File operations)

`GENERAL` (Others)

`COMMANDS` (External interface)

## Campaign | unicaACOOptAdmin

These configuration properties define settings for the unicaACOOptAdmin tool.

### getProgressCmd

#### Description

Specifies a value that is used internally. Do not change this value.

#### Default value

`optimize/ext_optimizeSessionProgress.do`

#### Valid Values

`optimize/ext_optimizeSessionProgress.do`

### runSessionCmd

#### Description

Specifies a value that is used internally. Do not change this value.

#### Default value

`optimize/ext_runOptimizeSession.do`

#### Valid Values

`optimize/ext_runOptimizeSession.do`

## loggingLevels

### Description

The `loggingLevels` property controls the amount of detail that is written to the log file for the Unica Optimize command-line tool, which is based on severity. Available levels are LOW, MEDIUM, HIGH, and ALL, with LOW providing the least detail (that is, only the most severe messages are written). The ALL level includes trace messages and is intended primarily for diagnostic purposes.

### Default value

HIGH

### Valid Values

LOW | MEDIUM | HIGH | ALL

## cancelSessionCmd

### Description

Specifies a value that is used internally. Do not change this value.

### Default value

`optimize/ext_stopOptimizeSessionRun.do`

### Valid Values

`optimize/ext_stopOptimizeSessionRun.do`

## logoutCmd

### Description

Specifies a value that is used internally. Do not change this value.

### Default value

`optimize/ext_doLogout.do`

### Valid Values

`optimize/ext_doLogout.do`

## getProgressWaitMS

### Description

Set this value to the number (integer) of milliseconds between two successive polls to the web application to get progress information. This value is not used if you do not set `getProgressCmd`.

### Default value

1000

**Valid Values**

An integer greater than zero

## Campaign | server

The property in this category specifies a URL that is used internally, and does not need to be changed.

**fullContextPath****Description**

The `fullContextPath` specifies the URL that Unica Campaign flowcharts use to communicate to the application server Listener proxy. This property is undefined by default, which causes the system to determine the URL dynamically. When Unica Platform is integrated with the Tivoli® web access control platform, you must set this property to the Unica Campaign URL in Tivoli®.

**Default value**

No default value defined.

**numRetryServerCommand****Description**

The `numRetryServerCommand` specifies the maximum number of times the Campaign web application can call the Campaign analytical server (listener) command until it receives a successful result. If the Campaign application continues to receive a non-success response after the maximum number of tries, a `Server Busy` error is displayed on the user interface.

Modify this parameter based on the Campaign Analytics server response time, the network speed and latency.

**Default value**

5

## Campaign | logging

This category specifies the location of the Unica Campaign `log4jConfig` properties file.

**log4jConfig****Description**

The Unica Campaign web application uses the Apache log4j utility for logging configuration, debugging, and error information.

The `log4jConfig` property specifies the location of the Unica Campaign log properties file, `campaign_log4j.xml`. Specify the path relative to the Unica Campaign home directory, including the file name. Use forward slashes (/) for UNIX™ and backslashes (\) for Windows™.

**Default value**

`./conf/campaign_log4j.xml`

## Campaign | proxy

The Unica Campaign, Acoustic Campaign, and UBX integration is supported with outbound proxy connections.

To access these properties, choose **Settings > Configuration**.

### Proxy host name

**Description**

Specify the hostname or the IP address of your proxy server.

### Proxy port number

**Description**

Specify the port number of your proxy server.

### Proxy type

**Description**

Select the proxy server type.

**Default value**

HTTP

**Valid value**

HTTP, SOCK5

### Data source for credentials

**Description**

Specify the datasource name that contains the proxy server user name and password details.

### Platform user with data source for proxy credentials

**Description**

Specify the name of the Platform user that has the specified datasource in the **Data source for credentials** property.



**Note:** When you deploy Campaign on a WebLogic server and HTTP proxy is configured, you need to add the variable `DUseSunHttpHandler=true` in `JAVA_OPTION` to the `setDomainEnv.cmd` file.



## Campaign | FlowchartEvents

If you are configuring Unica Director with Unica Campaign, use this category and set "enableEvents" to "Yes". If this property is set to NO, Unica Campaign will not publish any flowchart execution details to Unica Director. These properties must be set only once per instance of Unica Campaign; they do not need to be set for every partition of Unica Campaign.

To access these properties, choose **Settings > Configuration > Unica Campaign > FlowchartEvents**.

### enableEvents

#### Description

Configuration category:

Campaign|FlowchartEvents

The enableEvents property specifies whether Unica Campaign should send flowchart events to ActieMQ, all the Flowchart execution details are sent to Unica Director using Active MQ (version apache-activemq-5.15.7). which in turn are utilized by Unica Director and are used for Monitoring feature available in it, to display the status of flowcharts.

#### Default value

NO

#### Valid value

NO | Yes

## Campaign | FlowchartEvents |ActiveMQ

Unica Campaign application publishes the flowchart information to Unica Director using Message Queue. Data is pushed from Unica Campaign to Message Queue and consumed in Unica Director. Unica Director stores the flowchart information data. Message Queue should be up and running all the time to publish the data to Unica Director. In case Message Queue is not up then flowchart execution data will not be available with Unica Director.

### URL

#### Description

Specify the activeMQ listener url , for eg : tcp://localhost:61616

### Platform User with Data Sources for ActiveMQ Credentials

Provide the Unica Platform user name that holds the DataSource containing ActiveMQ credentials. , for example: asm\_admin

### Data Source For ActiveMQ message broker credentials

Specify the name of the dataSource that holds the credentials. Default value is - ACTIVEMQ\_CRED\_DS

## queueName

This mentions the queue name used to publish the flowchart information from Unica Campaign. Please do not modify this , Default value is "flowchartInfo-campaign"

## Report configuration properties

The report configuration properties for Unica are available at **Settings > Configuration > Reports**.

To generate reports, the Unica suite integrates with Cognos®, a business intelligence application. You use the **Integrations > Cognos** properties to identify your Cognos® system. Then, for Unica Campaign, Unica Deliver, and Unica Interact, you must configure additional properties to set up and customize the reporting schemas. For more details on the configuration properties, see the Cognos Reports Installation and Configuration Guide.

## Reports | Integrations | Cognos [version]

The Unica suite integrates with Cognos to generate reports.

This page displays properties that specify URLs and other parameters that are used by the system.

### Integration Name

#### Description

Read-only. Specifies that IBM Cognos is the third-party reporting or analytical tool that is used by the Unica to display the reports.

#### Default value

Cognos

### Vendor

#### Description

Read-only. Specifies that IBM Cognos is the name of the company that provides the application that is specified by the Integration Name property.

#### Default value

Cognos

### Version

#### Description

Read-only. Specifies the product version of the application that is specified by the Integration Name property.

#### Default value

<version>

## Enabled

### Description

Specifies whether IBM Cognos is enabled for the suite.

### Default value

False

### Valid Values

True | False

## Integration Class Name

### Description

Read-only. Specifies the fully qualified name of the Java class that creates the integration interface that is used to connect to the application specified by the `Integration Name` property.

### Default value

`com.unica.report.integration.cognos.CognosIntegration`

## Domain

### Description

Specifies the fully qualified company domain name in which your Cognos server is running. For example, `myCompanyDomain.com`.

If your company uses subdomains, the value in this field must include the appropriate subdomain as well.

### Default value

[CHANGE ME]

### Valid Values

A string no longer than 1024 characters.

## Portal URL

### Description

Specifies the URL of the IBM Cognos Connection portal. Use a fully qualified host name, including the domain name (and subdomain, if appropriate) that is specified in the **Domain** property. For example:

`http://MyReportServer.MyCompanyDomain.com/cognos<version>/cgi-bin/cognos.cgi`

You can find the URL in IBM Cognos Configuration at: **Local Configuration > Environment** .

### Default value

```
http://[CHANGE ME]/cognos<version>/cgi-bin/cognos.cgi
```

### Valid Values

A well-formed URL.

## Dispatch URL

### Description

Specifies the URL of the IBM Cognos Content Manager. Use a fully qualified host name, including the domain name (and subdomain, if appropriate) specified in the `Domain` property. For example: `http://MyReportServer.MyCompanyDomain.com:9300/p2pd/servlet/dispatch`

You can find the URL in Cognos® Configuration at: **Local Configuration > Environment** .

### Default value

```
http://[CHANGE ME]:9300/p2pd/servlet/dispatch
```

9300 is the default port number for the Cognos Content Manager. Be sure that the port number specified matches that used in the Cognos installation.

### Valid Values

A well-formed URL.

## Authentication mode

### Description

Specifies whether the IBM Cognos application is using the `Authentication Provider`, which means it relies on the Unica Platform for authentication.

### Default value

```
anonymous
```

### Valid Values

- `anonymous`: means that authentication is disabled.
- `authenticated`: means that the communications between the system and the Cognos system are secured at the machine level. You configure a single system user and configure it with the appropriate access rights. By convention, this user is named "cognos\_admin."
- `authenticatedPerUser`: means that the system evaluates individual user credentials.

## Authentication namespace

### Description

Read only. The namespace of the `Authentication Provider`.

### Default value

UNICA

## Authentication user name

### Description

Specifies the login name for the reporting system user. The applications log in to Cognos as this user when Cognos is configured to use the Unica Authentication provider. This user also has access to Unica.

This setting applies only when the **Authentication mode** property is set to **authenticated** .

### Default value

cognos\_admin

## Authentication datasource name

### Description

Specifies the name of the data source for the reporting system user that holds the Cognos login credentials.

### Default value

Cognos

## Enable form authentication

### Description

Specifies whether form-based authentication is enabled. You set this property to `True` when either of the following condition is true:

- When the Unica is not installed in the same domain as the IBM® Cognos applications.
- When IBM Cognos is accessed by using an IP address (within the same network domain) instead of the Fully Qualified Hostname (which is being used to access the Unica applications), even if both the Unica applications and the IBM Cognos installation are on the same machine.

However, when the value is `True`, the login process to Cognos Connection passes the login name and password in clear text and therefore is not secure unless IBM Cognos and the Unica are configured to use SSL communication.

Even with SSL configured, the user name and password appear as clear text in the HTML source code when you "view source" in a displayed report. For this reason, you must install IBM Cognos and Unica in the same domain.

### Default value

False

## Valid Values

True | False

## Reports | Schemas | [product] | [schema name] | SQL Configuration

The SQL script creates views for the reporting schemas. The **Reports | Schemas | [product] | [schema name] | SQL Configuration** property provides information about the name of the views.

### Table/View Name

#### Description

Specifies the name of the view that the SQL script you generate for this reporting schema creates. As a best practice, you should not change the name for any of the standard or default Table/View names. If you do, you must also change the name of the view in the Cognos® model in IBM® Cognos® Framework Manager.

When you create a new reporting schema for a new audience level, you must specify the names of all the new reporting tables/views.

#### Default value

Varies by schema

#### Valid Values

A string with the following restrictions.

- It can be no longer than 18 characters
- It must use all uppercase letters

You must use the following naming convention:

- Start the name with the letter "UAR"
- Add a one-letter code to represent the Unica application. See the list of codes, below.
- Add an underscore character
- Add the table name, including a one or two letter code to indicate the audience level
- Finish with an underscore character.

The SQL generator appends a time dimension code, if appropriate. See the following list of codes:

For example: `UARC_COPERF_DY` is the name of the reporting view for Campaign Offer Performance by Day.

Following is the list of Unica application codes.

- Unica Campaign: C
- Unica Deliver: E
- Unica Interact: I

- Unica Collaborate: X
- Unica Plan: P
- Leads: L

Following is the list of the Time Dimension Codes added by the generator.

- Hour: HR
- Day: DY
- Week: WK
- Month: MO
- Quarter: QU
- Year: YR

## Reports | Schemas | Campaign

The **Reports | Schemas | Campaign** property provides information about the data source that identifies the Unica Campaign database.

### Input Datasource (JNDI)

#### Description

Specifies the name of the JNDI data source that identifies the Unica Campaign database, specifically, the system tables. This data source must exist if you want to use the SQL generation tool to generate scripts that create reporting tables. The SQL generation tool can generate scripts that create reporting views without this data source, but it cannot validate them.

The database type of this data source must match the database type that you select when you generate the SQL scripts for the Unica Campaign views or reporting tables.

#### Default value

`campaignPartition1DS`

## Reports | Schemas | Campaign | Offer Performance

The Offer Performance Schema yields contact and response history metrics for all offers and for offers by campaign. By default, the schema is configured to generate a "summary" view across all time.

### Audience Key

#### Description

Specifies the name of the column that is the Audience Key for the audience level that is supported by this reporting schema.

#### Default value

`CustomerID`

### **Valid Values**

A string value no longer than 255 characters

If the key includes more than one column, use commas between the column names. For example,

`ColumnX,ColumnY.`

## **Contact History Table**

### **Description**

Specifies the name of the Contact History table for the audience level that is supported by this reporting schema.

### **Default value**

`UA_ContactHistory`

## **Detailed Contact History Table**

### **Description**

Specifies the name of the Detailed Contact History table for the audience level that is supported by this reporting schema.

### **Default value**

`UA_DtlContactHist`

## **Response History Table**

### **Description**

Specifies the name of the Response History table for the audience level that is supported by this reporting schema.

### **Default value**

`UA_ResponseHistory`

## **Over Time Variations**

### **Description**

Specifies the calendar time periods that are used by the "over time" reports supported by this schema.

### **Default value**

`Day, Month`

### **Valid Values**

`Day, Week, Month, Quarter, Year`



## Reports | Schemas | Campaign | [schema name] | Columns | [Contact Metric]

Use the **Reports | Schemas | Campaign | [schema name] | Columns | [Contact Metric]** property to add contact metrics to the Campaign Performance or Offer Performance reporting schemas.

### Column Name

#### Description

Specifies the name to use in the reporting view for the column that is specified in the **Input Column Name** field.

#### Default value

[CHANGE ME]

#### Valid Values

The name can be no longer than 18 characters, it must be in all uppercase letters, and it cannot have spaces.

### Function

#### Description

Specifies how the contact metric is determined or calculated.

#### Default value

count

#### Valid Values

count, count distinct, sum, min, max, average

### Input Column Name

#### Description

The name of the column that provides the contact metric you are adding to this reporting schema.

#### Default value

[CHANGE ME]

#### Valid Values

The name of the column in the Contact History and Detailed Contact History tables.

### Control Treatment Flag

#### Description

If you use the sample IBM® Cognos® reports or create your own custom reports that include control groups, then each contact metric must have two columns in the reporting schema. One column

represents the metric for the control group and the other column represents the metric for the target group. The value in **Control Treatment Flag** specifies whether the column in the view represents the control group or the target group.

If your reports do not include control groups, you do not need the second column for the control group.

#### Default value

0

#### Valid Values

- 0: the column represents the target group
- 1: the column represents the control group

## Reports | Schemas | Campaign | [schema name] | Columns | [Response Metric]

Use the **Reports | Schemas | Campaign | [schema name] | Columns | [Response Metric]** property to add the response metrics that you want to include in your reports to the Campaign Performance or Offer Performance reporting schemas.

### Column Name

#### Description

Specifies the name to use in the reporting view for the column that is specified in the **Input Column Name** field.

#### Default value

[CHANGE ME]

#### Valid Values

The name can be no longer than 18 characters, it must be in all uppercase letters, and it cannot have spaces.

### Function

#### Description

Specifies how the response metric is determined or calculated.

#### Default value

count

#### Valid Values

count, count distinct, sum, min, max, average

## Input Column Name

### Description

The name of the column that provides the response metric you are adding to this reporting schema.

### Default value

[CHANGE ME]

### Valid Values

The name of the column in the Response History table.

## Control Treatment Flag

### Description

If you use the standard IBM® Cognos® reports or create your own custom reports that include control groups, then each response metric must have two columns in the reporting schema. One column represents the response from the control group and the other column represents the response from the target group. The value in **Control Treatment Flag** specifies whether the column in the view represents the control group or the target group.

If your reports do not include control groups, you do not need the second column for the control group.

### Default value

0

### Valid Values

- 0: the column represents the target group
- 1: the column represents the control group

## Reports | Schemas | Campaign | Campaign Performance

The Campaign Performance schema yields contact and response history metrics at the campaign, campaign-offer, and campaign-cell level.

## Audience Key

### Description

Specifies the name of the column that is the Audience Key for the audience level that is supported by this reporting schema.

### Default value

CustomerID

### Valid Values

A string value no longer than 255 characters.

If the key includes more than one column, use commas between the column names. For example,

`ColumnX,ColumnY`.

## Contact History Table

### Description

Specifies the name of the Contact History table for the audience level that is supported by this reporting schema.

### Default value

`UA_ContactHistory`

## Detailed Contact History Table

### Description

Specifies the name of the Detailed Contact History table for the audience level that is supported by this reporting schema.

### Default value

`UA_DtlContactHist`

## Response History Table

### Description

Specifies the name of the Response History table for the audience level that is supported by this reporting schema.

### Default value

`UA_ResponseHistory`

## Over Time Variations

### Description

Specifies the calendar time periods that are used by the "over time" reports supported by this schema.

### Default value

`Day, Month`

### Valid Values

`Day, Week, Month, Quarter, Year`

## Reports | Schemas | Campaign | Campaign Offer Response Breakout

The Campaign Offer Response Breakout schema supports reporting on campaign-detailed responses, which are broken out by response type and by offer data. This schema template gives different response counts for each custom Response Type for campaigns and offers grouped by campaign.

This schema

### Response History Table

#### Description

Specifies the name of the Response History table for the audience level that is supported by this reporting schema.

#### Default value

`UA_ResponseHistory`

## Reports | Schemas | Campaign | Campaign Offer Response Breakout | Columns | [Response Type]

Use the **Reports | Schemas | Campaign | Campaign Offer Response Breakout | Columns | [Response Type]** property to add any custom response types that you want to include in your reports to the reporting schema.

### Column Name

#### Description

Specifies the name to use in the reporting view for the column that is specified in the **Response Type Code** field.

#### Default value

[CHANGE ME]

#### Valid Values

The name can be no longer than 18 characters, it must be in all uppercase letters, and it cannot have spaces.

### Response Type Code

#### Description

The response type code for the specified response type. This value is held in the `ResponseTypeCode` column in the `UA_UsrResponseType` table.

#### Default value

[CHANGE ME]

#### Valid Values

The example response type codes are as follows:

- `EXP` (explore)
- `CON` (consider)
- `CMT` (commit)
- `FFL` (fulfill)
- `USE` (use)
- `USB` (unsubscribe)
- `UKN` (unknown)

Your Unica Campaign installation may have additional custom response type codes.

## Control Treatment Flag

### Description

If you use the standard IBM® Cognos® reports provided in the Unica Reports Pack or custom reports that include control groups, then each response type must have two columns in the reporting schema. One column represents the response type from the control group and the other column represents the response type from the target group. The value in **Control Treatment Flag** specifies whether the column in the view represents the control group or the target group.

If your reports do not include control groups, you do not need the second column for the control group.

### Default value

0

### Valid Values

- 0: the column represents the target group
- 1: the column represents the control group

## Reports | Schemas | Campaign | Campaign Offer Contact Status Breakout

The Campaign Offer Contact Status Breakout schema supports reporting on campaign-detailed contacts, which are broken out by contact status type and by offer data. This schema template gives different contact counts for each custom Contact Status Type for campaigns and offers grouped by campaign.

By default, none of the example Unica Campaign reports use this schema.

## Audience Key

### Description

Specifies the name of the column that is the Audience Key for the audience level that is supported by this reporting schema.

**Default value**`CustomerID`**Valid Values**

A string value no longer than 255 characters.

If the key includes more than one column, use commas between the column names. For example,

`ColumnX,ColumnY.`**Contact History Table****Description**

Specifies the name of the Contact History table for the audience level that is supported by this reporting schema.

**Default value**`UA_ContactHistory`**Detailed Contact History Table****Description**

Specifies the name of the Detailed Contact History table for the audience level that is supported by this reporting schema.

**Default value**`UA_DtlContactHist`**Reports | Schemas | Campaign | Campaign Offer Contact Status Breakout | Columns | [Contact Status]**

Use the **Reports | Schemas | Campaign | Campaign Offer Contact Status Breakout | Columns | [Contact Status]** to add the contact status that you want to include in your reports to the reporting schemas.

**Column Name****Description**

Specifies the name to use in the reporting view for the column that is specified in the **Contact Status** field.

**Default value**`[CHANGE ME]`**Valid Values**

The name can be no longer than 18 characters, it must be in all uppercase letters, and it cannot have spaces.

## Contact Status Code

### Description

The name of the contact status code. This value is held in the `ContactStatusCode` column in the `UA_ContactStatus` table.

### Default value

[CHANGE ME]

### Valid Values

The example contact status types are as follows.

- `CSD` (campaign send)
- `DLV` (delivered)
- `UNDLV` (undelivered)
- `CTR` (control)

Your Unica Campaign installation may have additional custom contact status types.

## Reports | Schemas | Campaign | Campaign Custom Attributes | Columns | [Campaign Custom Column]

Use the **Reports | Schemas | Campaign | Campaign Custom Attributes | Columns | [Campaign Custom Column]** property to add any custom campaign attributes that you want to include in your reports to the reporting schema.

### Column Name

#### Description

Specifies the name to use in the reporting view for the attribute that is identified in the **Attribute ID** field.

#### Default value

[CHANGE ME]

#### Valid Values

The name can be no longer than 18 characters, it must be in all uppercase letters, and it cannot have spaces.

### Attribute ID

#### Description



The value from the attribute's `AttributeID` column in the **UA\_CampAttribute** table.

**Default value**

0

**Value Type**

**Description**

The data type of the campaign attribute.

**Default value**

`StringValue`

**Valid Values**

`StringValue, NumberValue, DatetimeValue`

If this campaign attribute holds a currency value, select `NumberValue`.

If this campaign attribute's **Form Element Type** was set to `Select Box - String` in Unica Campaign, select `StringValue`.

## Reports | Schemas | Campaign | Campaign Custom Attributes | Columns | [Offer Custom Column]

Use the **Reports | Schemas | Campaign | Campaign Custom Attributes | Columns | [Offer Custom Column]** property to add any custom offer attributes that you want to include in your reports to the reporting schema.

Use this form to add

**Column Name**

**Description**

Specifies the name to use in the reporting view for the attribute that is identified in the **Attribute ID** field.

**Default value**

[CHANGE ME]

**Valid Values**

The name can be no longer than 18 characters, it must be in all uppercase letters, and it cannot have spaces.

**Attribute ID**

**Description**

The value from the attribute's `AttributeID` column in the **UA\_OfferAttribute** table.

#### Default value

0

### Value Type

#### Description

The data type of the offer attribute.

#### Default value

StringValue

#### Valid Values

StringValue, NumberValue, DatetimeValue

If this offer attribute holds a currency value, select `NumberValue`.

If this offer attribute's **Form Element Type** was set to `Select Box - String` in Campaign, select `StringValue`.

## Reports | Schemas | Campaign | Campaign Custom Attributes | Columns | [Cell Custom Column]

Use the **Reports | Schemas | Campaign | Campaign Custom Attributes | Columns | [Cell Custom Column]** property to add any custom cell attributes that you want to include in your reports to the reporting schema.

### Column Name

#### Description

Specifies the name to use in the reporting view for the attribute that is identified in the **Attribute ID** field.

#### Default value

[CHANGE ME]

#### Valid Values

The name can be no longer than 18 characters, it must be in all uppercase letters, and it cannot have spaces.

### Attribute ID

#### Description

The value from the attribute's `AttributeID` column in the **UA\_CellAttribute** table.

#### Default value

0

## Value Type

### Description

The data type of the cell attribute.

### Default value

`StringValue`

### Valid Values

`StringValue, NumberValue, DatetimeValue`

## Reports | Schemas | Interact

The Unica Interact reporting schemas reference three separate databases: the design time, runtime, and learning databases. Use the **Reports | Schemas | Interact** property to specify the JNDI names of the data sources for those databases.

The data sources that are specified on this page must exist if you want to use the Reporting SQL generation tool to generate scripts that create reporting tables. The SQL generation tool can generate scripts that create reporting views without these data sources, but it cannot validate the scripts.

The database type of the data sources must match the database type that you select when you generate the SQL scripts for the views or reporting tables.

### Interact Design Datasource (JNDI)

#### Description

Specifies the name of the JNDI data source that identifies the Unica Interact design time database, which is also the Unica Campaign system tables.

#### Default value

`campaignPartition1DS`

### Interact Runtime Datasource (JNDI)

#### Description

Specifies the name of the JNDI data source that identifies the Unica Interact runtime database.

#### Default value

`InteractRTDS`

### Interact Learning Datasource (JNDI)

#### Description

Specifies the name of the JNDI data source that identifies the Unica Interact learning database.

### Default value

`InteractLearningDS`

## Reports | Schemas | Interact | Interact Performance

The Interact Performance schema yields contact and response history metrics at the channel, channel-offer, channel-segment, channel-interaction point, interactive cell, interactive cell-offer, interactive cell-interaction point, interactive offer, interactive offer-cell, and interactive offer-interaction point levels.

### Audience Key

#### Description

Specifies the name of the column that is the Audience Key for the audience level that is supported by this reporting schema.

#### Default value

`CustomerID`

#### Valid Values

A string value no longer than 255 characters.

If the key includes more than one column, use commas between the column names. For example,

`ColumnX,ColumnY`.

### Detailed Contact History Table

#### Description

Specifies the name of the Detailed Contact History table for the audience level that is supported by this reporting schema.

#### Default value

`UA_DtlContactHist`

### Response History Table

#### Description

Specifies the name of the Response History table for the audience level that is supported by this reporting schema.

#### Default value

`UA_ResponseHistory`

### Over Time Variations

#### Description

Specifies the calendar time periods that are used by the "over time" reports supported by this schema.

**Default value**

Hour, Day

**Valid Values**

Hour, Day, Week, Month, Quarter, Year

## Reports | Schemas | Deliver

The **Reports | Schemas | Deliver** property specifies the name of the data source that identifies the Unica Deliver tracking tables, which are in the Unica Campaign system tables.

### Unica Deliver Tracking Datasource (JNDI)

**Description**

Specifies the name of the JNDI data source that identifies the Unica Deliver tracking tables, which are in the Unica Campaign system tables. This data source must exist if you want to use the Reports SQL generation tool to validate scripts that create reporting tables. The SQL generation tool can generate scripts that create reporting views without this data source, but it cannot validate them.

The database type of this data source must match the database type that you select when you generate the SQL scripts for the views or reporting tables.

**Default value**

campaignPartition1DS

## Chapter 23. Special characters in Unica Campaign object names

Some special characters are not supported in any Unica Campaign object names. In addition, some objects have specific naming restrictions.



**Note:** If you pass object names to your database (for example, if you use a user variable that contains a flowchart name), you must ensure that the object name contains only characters supported by your particular database. Otherwise, you will receive a database error.

### Special characters not supported

The following special characters are not supported in the names of campaigns, flowcharts, folders, offers, offer lists, schedules, segments, or sessions. These characters are also not supported in the Audience Level Name and the corresponding field names, which are defined in Campaign Settings.

**Table 51. Special characters not supported**

Character	Description
%	Percent
*	Asterisk
?	Question mark
	Pipe (vertical bar)
:	Colon
,	Comma
<	Less than symbol
>	Greater than symbol
&	Ampersand
\	Backward slash
/	Forward slash
"	Double quotation mark
+	Plus
\$	Dollar
Tab	Tab

## Objects with no naming restrictions

The following objects in Unica Campaign have no restrictions for characters used in their names.

- custom attribute *display* names (the *internal* names do have naming restrictions)
- offer templates

## Objects with specific naming restrictions

The following objects in Unica Campaign have specific restrictions on their names.

- Custom attribute *internal* names
- Audience Level Names and the corresponding field names, which are defined in Campaign Settings
- Cells
- Derived fields
- User table and field names

For these objects, names must:

- Contain only alphabetic or numeric characters, or the underscore (\_) character
- Start with an alphabetic character

For non-Latin-based languages, Unica Campaign supports all characters that are supported by the configured string encoding configured.



**Note:** Derived field names have additional restrictions.

## Naming restrictions for derived fields

Derived field names have the following restrictions:

- They cannot be the same as either of the following types of names:
  - A database keyword (such as INSERT, UPDATE, DELETE, or WHERE)
  - A field in a mapped database table
- They cannot use the words *Yes* OR *No*.

If you do not follow these naming restrictions, database errors and disconnects may result when these derived fields are called.



**Note:** Derived field names also have specific character restrictions. For details, see [Special characters in Unica Campaign object names \(on page 542\)](#)

# Chapter 24. Internationalization and character encodings

This section provides information on character encodings and language-sensitive database considerations, and lists the encodings supported by Unica Campaign.

## Character encodings in Unica Campaign

Unica Campaign supports the character encodings that are described in this topic.

On most operating systems, Unica Campaign uses the GNU iconv library. **does not ship an iconv for AIX® installations.** For an AIX® system, you must obtain the appropriate character sets.

This section lists the encodings that Unica Campaign supports. The values in these lists are valid values for setting the Unica Campaign internationalization parameters listed in [Set Unica Campaign language and locale property values \(on page 336\)](#). Note the following:

- Each bullet within an encoding group is a space-separated list of different names for the same encoding. Each name in a bullet with multiple names is an alias for the other encodings in the group. You can set the Unica Campaign configuration parameters to any of the values in a group, depending on how your system uses the encodings.
- When you set values for the Unica Campaign `StringEncoding` configuration parameters, the pseudo-encoding `WIDEUTF-8` is the recommended value in most cases. However, you can use one of the encodings included in the following lists. Additionally, if the database is DB2® or SQL Server, you should use a code page rather than one of the encodings in this list. For details, see the context help or the *Unica Platform Administrator's Guide*.
- Unica Campaign uses two character encodings that are treated slightly differently than other encodings: "ASCII" and "UTF-8." Both are case-sensitive. These encodings are accepted on all platforms including AIX®. They have slightly different behaviors in Unica Campaign for column widths during table mapping, and for transcoding operations.

Some possible locale abbreviations are shown in parentheses: Arabic (ar), Armenian (hy), Chinese (zh), English (en), French (fr), Georgian (ka), Greek (el), Hebrew (he), Icelandic (is), Japanese (ja), Korean (ko), Lao (lo), Romanian (ro), Thai (th), Turkish (tr), Vietnamese (vi).

### Western Europe

- CP819 IBM819 ISO-8859-1 ISO-IR-100 ISO8859-1 ISO\_8859-1 ISO\_8859-1:1987 L1 LATIN1 CSISOLATIN1
- CP1252 MS-ANSI WINDOWS-1252
- 850 CP850 IBM850 CSPC850MULTILINGUAL
- MAC MACINTOSH MACROMAN CSMACINTOSH
- NEXTSTEP
- HP-ROMAN8 R8 ROMAN8 CSHPROMAN8



## Unicode encodings

- ISO-10646-UCS-2 UCS-2 CSUNICODE
- UCS-2BE UNICODE-1-1 UNICODEBIG CSUNICODE11
- UCS-2LE UNICODELITTLE
- ISO-10646-UCS-4 UCS-4 CSUCS4
- UTF-8
- UCS-4BE
- UCS-4LE
- UTF-16
- UTF-16BE
- UTF-16LE
- UTF-32
- UTF-32BE
- UTF-32LE
- UNICODE-1-1-UTF-7 UTF-7 CSUNICODE11UTF7
- UCS-2-INTERNAL
- UCS-2-SWAPPED
- UCS-4-INTERNAL
- UCS-4-SWAPPED
- JAVA
- C99

## Arabic

- ARABIC ASMO-708 ECMA-114 ISO-8859-6 ISO-IR-127 ISO8859-6 ISO\_8859-6 ISO\_8859-6:1987 CSISOLATINARABIC
- CP1256 MS-ARAB WINDOWS-1256
- MACARABIC
- CP864 IBM864 CSIBM864

## Armenian

- ARMSCII-8

## Baltic Rim

- CP1257 WINBALTRIM WINDOWS-1257
- CP775 IBM775 CSPC775BALTIC
- ISO-8859-13 ISO-IR-179 ISO8859-13 ISO\_8859-13 L7 LATIN7

## Celtic

- ISO-8859-14 ISO-CELTIC ISO-IR-199 ISO8859-14 ISO\_8859-14 ISO\_8859-14:1998 L8 LATIN8

## Central Europe

- ISO-8859-2 ISO-IR-101 ISO8859-2 ISO\_8859-2 ISO\_8859-2:1987 L2 LATIN2 CSISOLATIN2CP1250 MS-EE WINDOWS-1250
- MACCENTRALEUROPE
- 852 CP852 IBM852 CSPCP852
- MACCROATIAN

## Chinese (both Simplified and Traditional)

- ISO-2022-CN CSISO2022CN
- ISO2022CNISO-2022-CN-EXT

## Chinese (Simplified)

- CN\_GB\_1988-80 ISO-IR-57 ISO646-CN CSISO57GB1988
- CHINESE\_GB\_2312-80 ISO-IR-58 CSISO58GB231280
- CN-GB-ISOIR165 ISO-IR-165
- CN-GB EUC-CN EUCCN GB2312 CSGB2312
- CP936 GBK
- GB18030
- HZ HZ-GB-2312

## Chinese (Traditional)

- EUC-TW EUCTW CSEUCTWB
- IG-5 BIG-FIVE BIG5 BIGFIVE CN-BIG5 CSBIG5
- CP950
- BIG5-HKSCS BIG5HKSCS

## Cyrillic

- CYRILLIC ISO-8859-5 ISO-IR-144 ISO8859-5 ISO\_8859-5 ISO\_8859-5:1988 CSISOLATINCYRILLIC
- CP1251 MS-CYRL WINDOWS-1251
- MACCYRILLIC
- KOI8-R CSKOI8R
- KOI8-U
- KOI8-RU
- KOI8-T
- 866 CP866 IBM866 CSIBM866
- 855 CP855 IBM855 CSIBM855
- CP1125 ("PC, Cyrillic, Ukrainian")
- MACUKRAINE

## English

- ANSI\_X3.4-1968 ANSI\_X3.4-1986 ASCII CP367 IBM367 ISO-IR-6 ISO646-US ISO\_646.IRV:1991 US US-ASCII CSASCII
- 437 CP437 IBM437 CSPC8CODEPAGE437

## Georgian

- GEORGIAN-ACADEMY
- GEORGIAN-PS

## Greek

- CP1253 MS-GREEK WINDOWS-1253
- ECMA-118 ELOT\_928 GREEK GREEK8 ISO-8859-7 ISO-IR-126 ISO8859-7 ISO\_8859-7 ISO\_8859-7:1987 CSISOLATINGREEK
- MACGREEK
- CP737869 CP-GR CP
- 869 IBM869 CSIBM869

## Hebrew

- HEBREW ISO-8859-8 ISO-IR-138 ISO8859-8 ISO\_8859-8 ISO\_8859-8:1988 CSISOLATINHEBREW
- CP1255 MS-HEBR WINDOWS-1255
- 862 CP862 IBM862 CSPC862LATINHEBREW
- MACHEBREW

## Icelandic

- MACICELAND
- 861 CP-IS CP861 IBM861 CSIBM861

## Japanese

- JISX0201-1976 JIS\_X0201 X0201 CSHALFWIDTHKATAKANA
- ISO-IR-87 JIS0208 JIS\_C6226-1983 JIS\_X0208 JIS\_X0208-1983 JIS\_X0208-1990 X0208 CSISO87JISX0208
- ISO-IR-159 JIS\_X0212 JIS\_X0212-1990 JIS\_X0212.1990-0 X0212 CSISO159JISX02121990
- EUC-JP EUCJP EXTENDED\_UNIX\_CODE\_PACKED\_FORMAT\_FOR\_JAPANESE CSEUCPKDFMTJAPANESE
- MS\_KANJI SHIFT-JIS SHIFT\_JIS SJIS CSSHIFTJI
- ISO-IR-14 ISO646-JP JIS\_C6220-1969-RO JP CSISO14JISC6220RO
- CP932
- ISO-2022-JP CSISO2022JP
- ISO-2022-JP-1
- ISO-2022-JP-2 CSISO2022JP2

## Korean

- EUC-KR EUCKR CSEUCKR
- CP949 UHC
- ISO-IR-149 KOREAN KSC\_5601 KS\_C\_5601-1987 KS\_C\_5601-1989 CSKSC56011987
- CP1361 JOHAB
- ISO-2022-KR CSISO2022KR

## Lao

Note that Lao uses the same alphabet as Thai.

- MULELAO-1
- CP1133 IBM-CP1133

## Northern Europe

- ISO-8859-4 ISO-IR-110 ISO8859-4 ISO\_8859-4 ISO\_8859-4:1988 L4 LATIN4 CSISOLATIN4
- ISO-8859-10 ISO-IR-157 ISO8859-10 ISO\_8859-10 ISO\_8859-10:1992 L6 LATIN6 CSISOLATIN6

## Romanian

- MACROMANIA

## Southern Europe

- ISO-8859-3 ISO-IR-109 ISO8859-3 ISO\_8859-3 ISO\_8859-3:1988 L3 LATIN3 CSISOLATIN3
- CP853

## Thai

- MACTHAI
- ISO-IR-166 TIS-620 TIS620 TIS620-0 TIS620.2529-1 TIS620.2533-0 TIS620.2533-1
- CP874 WINDOWS-874

## Turkish

- CP1254 MS-TURK WINDOWS-1254
- MACTURKISH
- 857 CP857 IBM857 CSIBM857
- ISO-8859-9 ISO-IR-148 ISO8859-9 ISO\_8859-9 ISO\_8859-9:1989 L5 LATIN5 CSISOLATIN5

## Vietnamese

- CP1258 WINDOWS-1258
- TCVN TCVN-5712 TCVN5712-1 TCVN5712-1:1993
- VISCII VISCIII.1-1 CSVISCII

## Miscellaneous

- ISO-8859-15 ISO-IR-203 ISO8859-15 ISO\_8859-15 ISO\_8859-15:1998
- ISO-8859-16 ISO-IR-226 ISO8859-16 ISO\_8859-16 ISO\_8859-16:2000
- CP858(IBM: "Multilingual with euro")
- 860 (IBM: "Portugal - Personal Computer")CP860 IBM860 CSIBM860
- 863 (IBM: "Canadian French - Personal Computer") CP863 IBM863 CSIBM863
- 865 (IBM: "Nordic - Personal Computer")CP865 IBM865 CSIBM865

## Date and time formats

Use the information in the following sections to determine how to configure the date and time format configuration properties `DateFormat`, `DateOutputFormatString`, `DateTimeFormat`, and `DateTimeOutputFormatString`.

### Formats for `DateFormat` and `DateTimeFormat`

If you are not configuring Unica Campaign for multiple locales, you can set the values for `DateFormat` and `DateTimeFormatConfiguration` parameters for any of the formats specified in the `DATE` macro, as described in this section.

However, **if you need to configure** Unica Campaign for **multiple locales** (if you have users with various languages and locales), **DO NOT** use date formats that contain 3-letter months (MMM), %b (abbreviated month name), or %B (full month name). Instead, use a delimited or fixed format that has a numeric value for the month. For more details about the multi-locale feature, see [About the multi-locale feature \(on page 333\)](#).

**Table 52. Date formats**

Format	Description	Example(s)
MM	2-digit month	01, 02, 03, ..., 12
MMDD	2-digit month and 2-digit day	March 31 is 0331
MMDDYY	2-digit month, 2-digit day, and 2-digit year	March 31, 1970 is 033170

**Table 52. Date formats (continued)**

Format	Description	Example(s)
MMDDYYYY	2-digit month, 2-digit day, and 4-digit year	March 31, 1970 is 03311970
DELIM_M_D  For DateTimeFormat, use  DT_DELIM_M_D	Any delimited month followed by day	March 31, 3/31, or 03-31
DELIM_M_D_Y  For DateTimeFormat, use  DT_DELIM_M_D_Y	Any delimited month, day, and year	March 31, 1970 or 3/31/70
DELIM_Y_M  For DateTimeFormat, use  DT_DELIM_Y_M	Any delimited year followed by month	1970 March, 70-3, 1970/3
DELIM_Y_M_D  For DateTimeFormat, use  DT_DELIM_Y_M_D	Any delimited year, month, and day	1970 Mar 31 or 70/3/31
YYMMM	2-digit year and 3-letter month	70MAR
YYMMDD	2-digit year, 3-letter month, and 2-digit day	70MAR31
YY	2-digit year	70
YYMM	2-digit year and 2-digit month	7003

**Table 52. Date formats (continued)**

Format	Description	Example(s)
YYMMDD	2-digit year, 2-digit month, and 2-digit day	700331
YYYYMM	4-digit year and 3-letter month	1970MAR
YYYYMMDD	4-digit year, 3-letter month, and 2-digit day	1970MAR31
YYYY	4-digit year	1970
YYYYMM	4-digit year and 2-digit month	197003
YYYYMMDD	4-digit year, 2-digit month, and 2-digit day	19700331
DELIM_M_Y  For DateTimeFormat, use  DT_DELIM_M_Y	Any delimited month followed by year	3-70, 3/70, Mar 70, March 1970
DELIM_D_M  For DateTimeFormat, use  DT_DELIM_D_M	Any delimited day followed by month	31-3, 31/3, 31 March
DELIM_D_M_Y  For DateTimeFormat, use  DT_DELIM_D_M_Y	Any delimited day, month, and year	31-MAR-70, 31/3/1970, 31 03 70
DD	2-digit day	31
DDMMM	2-digit day and 3-letter month	31MAR

**Table 52. Date formats (continued)**

<b>Format</b>	<b>Description</b>	<b>Example(s)</b>
DDMMYY	2-digit day, 3-letter month, and 2-digit year	31MAR70
DDMMYYYY	2-digit day, 3-letter month, and 4-digit year	31MAR1970
DDMM	2-digit day and 2-digit month	3103
DDMMYY	2-digit day, 2-digit month, and 2-digit year	310370
DDMMYYYY	2-digit day, 2-digit month, and 4-digit year	31031970
MMYY	2-digit month and 2-digit year	0370
MMYYYY	2-digit month and 4-digit year	031970
MMM	3-letter month	MAR
MMDD	3-letter month and 2-digit day	MAR31
MMDDYY	3-letter month, 2-digit day, and 2-digit year	MAR3170
MMDDYYYY	3-letter month, 2-digit day, and 4-digit year	MAR311970
MMYY	3-letter month and 2-digit year	MAR70
MMYYYY	3-letter month and 4-digit year	MAR1970



**Table 52. Date formats (continued)**

Format	Description	Example(s)
MONTH	Month of the year	January, February, March, and so on or Jan, Feb, Mar, and so on
WEEKDAY	Day of the week	Sunday, Monday, Tuesday, and so on (Sunday = 0)
WKD	Abbreviated day of the week	Sun, Mon, Tues, and so on  (Sun = 0)

## Formats for DateOutputFormatString and DateTimeOutputFormatString

If you are not configuring Unica Campaign for multiple locales, you can set the values for `DateOutputFormat` and `DateTimeOutputFormat` configuration parameters for any of the formats specified for `format_str` in the `DATE_FORMAT` macro, as shown in the following table.

However, **if you need to configure** Unica Campaign for **multiple locales** (that is, if you have users with various languages and locales), **DO NOT** use date formats that contain 3-letter months (MMM), %b (abbreviated month name), or %B (full month name). Instead, you should use one of the delimited or fixed formats that have a numeric value for the month. For more details about the multi-locale feature, see [About the multi-locale feature \(on page 333\)](#).

%a - Abbreviated weekday name

%A - Full weekday name

%b - Abbreviated month name

%B - Full month name

%c - Date and time representation appropriate for locale

%d - Day of month (01 - 31)

%H - Hour in 24-hour format (00 - 23)

%I - Hour in 12-hour format (01 - 12)

%j - Day of year (001 - 366)

%m - Month number (01 - 12)

%M - Minute (00 - 59)

%p - Current locale's AM/PM indicator for 12-hour clock

%S - Second (00 - 59)

%U - Week of year, with Sunday as first day of week (00 - 51)

%w - Weekday (0 - 6; Sunday is 0)

%W - Week of year, with Monday as first day of week (00 - 51)

%x - Date representation for current locale

%X - Time representation for current locale

%y - 2-digit year (00 - 99)

%Y - 4-digit year

%Z, %Z - Time zone name or abbreviation; no output if time zone is unknown

%% - Percent sign



**Note:** Characters that are part of the format and are not preceded by a percent sign (%) are copied unchanged to the output string. The formatted string must fit in 16 bytes or less. Use the # character to eliminate any leading 0. For example, %d produces 2-digit numbers in the range (01 - 31), while %#d produces 1- or 2-digit numbers as appropriate (1 - 31). Similarly, %m gives (01 - 12) while %#m gives (1 - 12).

# Chapter 25. Unica Campaign Error Codes

Unica Campaign notifies users of error events when they occur with error messages that consist of a code number and error text.

Unica Campaign is a Web-based client-server application with two servers and a number of environment variables that you must configure for it to work properly.

If you see an error message that says you have invalid user access rights, you may not have the correct privileges assigned in Unica Platform to perform that action. For more information, see the Unica Platform Administrator's Guide.

If you encounter errors while using Unica Campaign, try reading the descriptions and applying the solutions in this section before contacting Technical Support. If an error does not appear here, or if a solution fails, contact your administrator or contact Technical Support.

## List of Unica Campaign error codes

The following table lists the error messages that are generated by Unica Campaign.

**Table 53. Unica Campaign error codes**

Code	Error Description
301	Unable to allocate requested memory.
303	Name conflicts with a built-in function name operator or keyword.
304	Name is too long or contains invalid character(s).
305	Named variable has no value assigned.
306	Syntax error in expression.
308	Error loading saved expression from file (large mem).
309	Error loading saved expression from file (unknown function).
310	Error loading saved expression from file (rand obj).
311	Error loading saved object from file (invalid ID).

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
312	Error loading saved expression from file (stack).
314	Error saving object to file (invalid ID).
315	Error saving expression to file (large mem).
316	Consecutive operators in expression.
317	Operator syntax error.
318	Parenthesis missing.
319	Parentheses mismatched.
320	Unknown expression.
321	No name given.
322	No expression following an equal sign.
323	Ambiguous field name.
324	Can't sort more than 2 <sup>16</sup> points.
325	Error accessing virtual memory (stat=0).
328	Matrix multiply dimension mismatch.
329	Matrix multiply dimension too big.
330	Singular matrix error.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
331	Invalid number of arguments.
332	Argument(s) must be scalar number(s).
333	Argument must be greater than 0.
334	Values of argument(s) are invalid.
335	Values of argument(s) must be between -1 and 1.
336	Dimensionality of argument(s) is invalid for the function.
338	Length of arguments must be equal.
339	Dimensionality of arguments must be equal.
341	Standard deviation or other statistical calculation is invalid.
342	Only vector is allowed for the 1st argument.
343	Supplied argument must be integer.
345	Expression is mathematically undefined.
346	Unable to get training pattern.
348	Supplied keyword is not appropriate for this function.
349	Floating pointer value overflow error.
350	Trying to take square root of negative number.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
353	Function's total returned string size is too large.
354	String type not allowed in one or more arguments.
356	Row/column index is not valid.
357	Mixing numbers and text columns not allowed.
358	String quote mismatch.
359	Expression is too complex.
360	String length too long.
361	Number parsing code not valid.
362	This function cannot process numbers.
363	String quote mismatched or missing.
364	This function generates too much data.
365	Too many outputs from this function.
367	Multiple-column outputs not allowed in recursive expressions.
368	Recursive function trying to access future values (not causal).
369	Inputs from the first row contains error.
370	Output column(s) too long.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
371	Input/Output dimension of algorithm is corrupted.
372	Recursive variable invalid
373	Internal Only: Null ParseTree!
377	Unknown value in substitution
381	Error encountered interpreting variable type: 'Money'
382	Error encountered interpreting Variable Type: 'Telephone'
383	Error encountered interpreting Variable Type: 'Date'
384	Error encountered interpreting Variable Type: 'Time'
393	Boolean expression may be compared only to 1 or 0
394	Values out of range for one or more arguments.
395	A numeric column must be specified with any keyword other than CountOf.
396	BETWEEN syntax is: <val> BETWEEN <val1> AND <val2>
397	SUBSTR[ING] syntax is: SUBSTR[ING](<string><offset><size>)
398	Option [OutputValue] is only allowed for MinOf, MaxOf, and MedianOf keywords.
399	Null value found.
450	Unable to change file permissions (chmod).

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
451	Unable to retrieve file attributes (stat).
452	Unable to remove file.
453	Unable to create memory object; check log file for memory or file errors.
454	Unable to lock memory object page; check log file for memory or file errors.
455	Unable to load memory object; check log file for memory or file errors.
456	Unable to create I/O object; check log file for memory or file errors.
457	Unable to create I/O object; check log file for memory errors.
458	Invalid supporting file extension; this may indicate a corrupted file.
459	Invalid UTF-8 character encountered.
460	Unable to convert from wide character to native encoding.
461	Unable to convert from native encoding to wide character.
462	Unable to create directory.
463	Unable to remove directory.
500	Internal parse tree construction error.
600	Internal error: Configuration root not specified.
601	Configuration server URL is not specified.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
602	Specified configuration category not found.
603	Specified configuration property must have absolute file path.
604	Invalid response from configuration server.
605	INTERNAL ERROR: Requested configuration path is different from current root.
606	Configuration categories and properties cannot have empty name.
607	Configuration category name cannot contain forward slash.
608	Specified configuration property must have relative file path.
609	INTERNAL ERROR: Partition name not specified.
610	Cannot determine default partition.
611	No partition with the given name.
612	No partitions defined at all.
614	Invalid parameters specified in config.xml.
620	INTERNAL ERROR: Security manager is already initialized.
621	INTERNAL ERROR: Failed to initialize security manager; Invalid parameters.
622	INTERNAL ERROR: Invalid result set name specified.
623	User is not mapped to any partition.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
624	User is mapped to multiple partitions.
625	User is not mapped to the specified partition.
626	User is not authorized to access the application.
700	Out of memory.
701	<p>Cannot open file.</p> <p>Possible causes:</p> <p>Unica Campaign could not transcode a non-ASCII file name.</p> <p>Unica Campaign could not locate the specified file.</p> <p>Unica Campaign cannot open the file properly.</p> <p>A file could not be copied because it could not be opened.</p> <p>Suggested solutions:</p> <p>Verify the file exists in the expected location.</p> <p>Examine the log file for the name of the file causing the error.</p> <p>Ask your system administrator for assistance.</p>
702	File seek error.
703	File read error.
704	File write error.
710	Flowchart file data corrupted.
711	File create error.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
723	Error in one or more input variables to this function.
761	Out of disk space.
768	Error saving file.
773	Access denied.
774	INTERNAL HMEM ERROR: cannot flush memory when swapping is disabled.
778	Numerical error: Unknown floating point error.
779	Numerical error: Explicit generation.
780	Numerical error: Invalid number.
781	Numerical error: Denormal.
782	Numerical error: Division by zero.
783	Numerical error: Floating point overflow.
784	Numerical error: Floating point underflow.
785	Numerical error: Floating point round-off.
786	Numerical error: Floating point unemulated.
787	Numerical error: Square root of negative number.
788	Numerical error: Stack overflow.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
789	Numerical error: Stack underflow.
790	Internal Error.
967	Data dictionary contains an invalid definition.
997	Internal error: GIO stack overflow.
998	Error loading object: size check failed.
999	Extended Error
1400	Row offset not found for given row
1500	Insufficient memory for this operation.
1501	Maximum histogram range exceeded
1550	Internal Error 1550:
1649	Vector is not allowed for any argument.
1650	First parameter cannot be vector if COL keyword is used.
1709	Client/Server version mismatch.
1710	Unable to initialize sockets.
1711	Unable to create a socket.
1712	<p>Unable to connect to the specified server.</p> <p>Possible causes:</p>

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
	<p>Your browser cannot connect to the Unica Campaign server.</p> <p>Your browser cannot find the host name.</p> <p>Suggested solutions:</p> <p>Ask your network administrator to check that the server and client machines can 'ping' each other.</p> <p>Ask your Unica Campaign administrator to check that the Unica Campaign server machine does not use the port assigned to the Unica Campaign listener process for another application.</p> <p>Try the steps that caused the error again. If the error occurs again, reboot your client machine, and ask your system administrator to reboot the Unica Campaign server machine.</p>
1713	Unable to send socket data.
1714	<p>Unable to receive socket data.</p> <p>Possible causes:</p> <p>The number of bytes received from socket does not match the expected number.</p> <p>Unica Campaign timed out waiting for data from the socket.</p> <p>There was a socket error when sending messages.</p> <p>Suggested solutions:</p> <p>Ask your network administrator to check that the server and client machines can 'ping' each other.</p> <p>Ask your Unica Campaign administrator to check that the Unica Campaign server machine does not use the port assigned to the Unica Campaign listener process for another application.</p> <p>Try the steps that caused the error again. If the error occurs again, reboot your client machine, and ask your system administrator to reboot the Unica Campaign server machine.</p> <p>If this error occurs in an integrated <b>IBM Digital Analytics</b> environment, it indicates that the Unica Campaign backend listener server cannot access the <code>export.coremetrics.com</code> API URL due to a network connectivity issue. For more information, read the integration troubleshooting topics.</p>

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
1715	Unable to bind socket to specified port.
1716	Unable to perform socket listen.
1717	Communication request has timed out.
1719	INTERNAL ERROR: Communication request has timed out.
1729	Client/Server library: Error retrieving drive information.
1731	INTERNAL ERROR: Invalid argument index specified.
1733	Listener unable to create semaphore.
1734	Listener: invalid File Block server port.
1735	Listener unable to invoke specified command.
1736	Listener: invalid UDME server port.
1737	Listener: invalid Shannon server port.
1738	Listener: unable to communicate with server process.
1739	Listener: internal data integrity error.
1741	Unable to create thread.
1742	Unable to wait for thread.
1743	Client/Server library: Invalid process. Possible causes: A process, such as trigger, bulk loader, or the UDISvr, no longer exists. Suggested solutions: Examine the log file to determine if one of these processes ended abnormally. Ask your Unica Campaign administrator to restart the

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
	process that ended abnormally. If the error occurs again, ask your system administrator for assistance.
1744	Client/Server library: Invalid semaphore.
1745	Client/Server library: Invalid mutex.
1746	Client/Server library: Insufficient memory.
1747	INTERNAL ERROR: Client/Server library: Timeout elapsed object not signaled.
1748	Client/Server library: Wait for object failed.
1749	Client/Server library: Invalid directory specified.
1750	INTERNAL ERROR: Requested server function unsupported.
1751	Server shutting down; request refused.
1773	UDMEsvr: flowchart requested for deletion is in use.
1783	Someone else is already in EDIT or RUN mode.
1784	Edit before run is completed is not allowed
1785	Requested flowchart is active for another user.
1786	<p>Server process has exited.</p> <p>Possible causes: The Unica Campaign listener cannot start the Unica Campaign server process.</p> <p>Suggested solutions: Ask your system administrator for assistance.</p>

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
1787	Maximum number of flowchart instances is already in use.
1788	Requested flowchart is active for Unica Collaborate.
1789	Requested flowchart is in use by an Unica Campaign user.
1790	<p>Unable to authenticate user.</p> <p>Possible causes:</p> <p>The password you provided does not match the password stored in the Unica Platform.</p> <p>The user name or password field for accessing the database or other object is empty in the Unica Platform.</p> <p>The user name or password field for accessing the database or other object is empty in the Unica Platform.</p> <p>Suggested solutions:</p> <p>Check that the user name and password you provided are correct.</p> <p>Ask your Unica Campaign administrator to check that your user names and passwords stored in the Unica Platform are complete and correct.</p>
1791	Invalid group name specified.
1792	Invalid file mode specified.
1793	INTERNAL ERROR: Exit status requested for active process.
1794	Evaluation period has expired.
1795	Invalid License Code.
1796	Flowchart name has been changed by the writer



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
1797	Flowchart name has been changed by the writer
1823	INTERNAL ERROR: Mismatch in request parameter count.
1824	INTERNAL ERROR: Mismatch in request parameter type.
1825	INTERNAL ERROR: Mismatch in request scalar or vector count.
1830	Unsupported protocol type encountered.
1831	Invalid API.
1832	No server process found for specified run. Run may be completed already.
2000	HTTP session object is invalid.
2001	HTTP connection object is invalid.
2002	HTTP request object is invalid.
2003	Error adding HTTP request headers.
2004	Error setting HTTP proxy credentials.
2005	Error setting HTTP server credentials.
2006	Error sending HTTP request.
2007	Error receiving HTTP response.
2008	Error querying HTTP response headers.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
2009	Error reading HTTP response data.
2010	Error status returned in HTTP response.
2011	Error querying HTTP authentication schemes.
2012	No matching HTTP authentication schemes.
2013	Proxy server authentication failure. You must provide a valid proxy server username and password in a data source named "proxy" in the Unica Platform before re-attempting to log into Unica Campaign.
2014	Web server authentication failure. You must provide a valid web server username and password in a data source named "webserver\" in the Unica Platform before re-attempting to log into Unica Campaign.
2015	HTTP request error after PAC file authentication failure.
2016	HTTP request error after PAC file scheme failure.
2051	KafkaBrokerURL not entered in kafka configuration settings.
2052	TopicName not entered in kafka configuration settings.
2053	UserForKafkaDataSource not entered in kafka configuration settings.
2054	KafkaDataSource not entered in kafka configuration settings.
2055	KafkaKeyFile not entered in kafka configuration settings.
2056	KafkaCertificateFile not entered in kafka configuration settings.
2057	CertificateAuthorityFile not entered in kafka configuration settings.
2058	Error while creating Kafka producer object
2059	Error while producing/sending msg to Kafka Server by producer.
2060	Could not communicate to Kafka server
2061	Error while creating user given topic.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
2062	Authentication error while communicating with Kafka server.
2063	Configurations for GSSAPI Sasl Mechanism are missing.
2100	Circular List in Master Listener is not initialized.
2101	Client ID missing in GetListenerForClient request.
2102	Request for Listener received on Non Master Listener.
2103	Message intended for Master Listener received on Non-Master Listener.
2104	Requested listener is not available
2105	List of Server-ID missing in GetListenerForClient failover request.
2106	Master Listener Internal Error - Client ID not found in cached data for failover request.
2107	Unable to issue Disconnect command as Master Listener is not available
2108	Master Listener Internal Error while reading cache
2109	Master Listener Internal Error - runID not found in cached data.
10001	Internal Error.
10022	Internal Error: Cannot find Process.
10023	Internal Error: Cannot find Connection.
10024	Internal Error: Cannot find Process.
10025	Internal Error: Cannot find Connection.
10026	Internal Error: Unknown function tag.
10027	Flowchart contains cycles.
10030	Internal Error: Cannot get memory buffer from GIO.
10031	flowchart is running.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10032	Internal Error: Unknown copy state.
10033	Error modifying system table.
10034	One or more Processes are not configured.
10035	Process has multiple Schedule inputs.
10036	Internal Error: Cannot find Process.
10037	One or more pasted Processes have derived fields defined. They may need to be redefined.
10038	Branch has one or more input processes outside of the branch.
10039	flowchart DOM creation error.
10040	flowchart DOM parse error.
10041	Flowchart is recovered from auto-save file.
10042	The Flowchart that creates the Global Suppression Segment needed by this run is currently executing.
10043	The Global Suppression Segment is missing.
10044	The Global Suppression Segment is at the wrong audience level.
10046	Only one process box of this type is allowed.
10047	Only one branch is allowed.
10048	Flowchart must begin with an Interaction process box.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10049	Unable to find treatment in the Treatment Cache.
10116	Internal Error: Process not registered.
10119	Internal Error: Unknown function tag.
10120	Process is running.
10121	Process run results will be lost.
10122	Internal Error.
10125	Process is not configured.
10126	Process input is not ready.
10127	Process name not unique.
10128	Internal Error: Invalid Process index.
10129	Internal Error: Invalid Report ID.
10130	Internal Error: Invalid Table ID.
10131	Internal Error: Invalid Field index.
10132	Internal Error: Invalid Cell ID.
10133	Internal Error: Invalid Field index.
10134	Internal Error: Invalid Process for registration.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10136	Process run stopped by user.
10137	Changes are not allowed while the Process is queued.
10138	Changes are not allowed while the Process is running.
10139	Changes are not allowed while a subsequent Process is running or queued.
10140	The source of the Process has been changed. Derived fields and subsequent Processes may need to be reconfigured.
10141	One or more selected Tables don't exist.
10142	Changes are not allowed while the flowchart is running.
10143	Process DOM creation error.
10144	Process DOM parse error.
10145	Unknown Process Parameter.
10146	Process name contains invalid character.
10147	Output Cell Name is empty.
10148	Accumulate ID's option must be turned off for schedule process to run on queue.
10149	Command not allowed in Reader mode.
10150	Cannot open Segment data file.
10151	Segment data file error: Invalid header.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10152	Internal Error: Invalid Segment (Data file name is blank).
10153	Undefined user variable referenced in path.
10154	Critical error encountered.
10155	A preceding process has not been run in production mode.
10156	Cell name conflict detected in flowchart.
10157	Cell code conflict detected in flowchart.
10158	Topdown target cell is linked more than once.
10159	Linked topdown cell missing or already linked to another
10161	Invalid field name.
10162	Target cell is not yet approved to run in production.
10163	In order to run in production, all input cells for this process must be linked to cells in the target cell spreadsheet (TCS).
10164	This process cannot handle topdown cells that are control cells or have control cells.
10165	Cannot open Segment TempTable.
10166	Internal Error: Invalid Segment (Segment TempTable DB is blank).
10167	Internal Error: Invalid Segment (Segment TempTable name is blank).
11167	Inputs have different audience levels.
11168	Specified flowchart template not found in the system.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11169	Interact base table mapping not found.
10200	Internal Error: Invalid 'From' Process
10201	Internal Error: Invalid 'To' Process
10206	Internal Error: Invalid 'From' Process
10207	Internal Error: Invalid 'To' Process
10208	Internal Error: Invalid connection index.
10209	Internal Error: DOM creation error.
10210	Internal Error: DOM parse error.
10211	Conflicting cell code ignored.
10300	ServerComm out of memory.
10301	INTERNAL ERROR: No function registered for class.
10302	INTERNAL ERROR: Requested function not supported.
10303	Another flowchart connection has been established; reconnect not allowed.
10304	All communication ports in the range specified in UNICA_ACSVR.CFG are in use.
10305	Requested flowchart is already in use.
10306	Command not allowed in reader mode



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10307	Flowchart is in use. Not enough privileges to takeover.
10350	INTERNAL ERROR: flowchart not running.
10351	INTERNAL ERROR: Client connected to flowchart.
10352	Unrecognized command.
10353	Invalid syntax.
10354	INTERNAL ERROR: SUSPEND RUN in progress.
10355	No sessions were affected. The operation cannot be performed at this time. Please review the flowchart log for the probable cause and try again later.
10356	New connections have been disabled. An administrator must use unica_svradm's UNCAP command to re-enable.
10357	flowchart run completed with errors.
10358	Cached data not found
10359	Flowcharts need to be specified with relative, not absolute, pathnames to the partitionHome property defined in the central configuration repository provided by Unica.
10362	Server host name must be mentioned. As client is connected to the master listener
10363	The specified command can be run on master listener only in clustered environment
10364	Message intended for Master Listener received on Non-Master Listener.
10401	INTERNAL ERROR: Client already connected.
10402	Client is not connected to a server.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10403	Connection to server has been lost. Retry?
10404	<p>Unable to communicate with server process; it may have exited.</p> <p>Possible causes</p> <p>The Unica Campaign server process:</p> <ul style="list-style-type: none"> <li>• cannot start when you log in, or when you create or open a flowchart.</li> <li>• no longer exists when you re-connect to the server.</li> <li>• terminated abnormally.</li> </ul> <p>Suggested solutions</p> <p>Ask your Unica Campaign administrator to check that the:</p> <ul style="list-style-type: none"> <li>• Unica Campaign listener process is running.</li> <li>• same version of the Unica Campaign web application, listener, and server are running on your system.</li> <li>• port number is properly configured in the Unica Platform.</li> </ul> <p>Ask your system administrator to examine the system log for additional information related to this error.</p>
10405	Server process is unresponsive; RETRY to wait or CANCEL to disconnect.
10406	INTERNAL ERROR: Communication with server already in progress.
10407	You have been disconnected; an administrator has suspended this flowchart.
10408	You have been disconnected; an administrator has killed this flowchart.
10409	You have been disconnected; an administrator has stopped this flowchart.
10410	You have been disconnected; an administrator has removed this flowchart.
10411	You have been disconnected; an administrator has taken control of this flowchart.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10412	HTTP SessionID is invalid, or HTTP Session has timed out.
10440	Windows impersonation failure
10441	Continue sending windows authentication messages
10442	Stop sending windows authentication messages
10443	Failed to generate TYPE-1 message
10444	Failed to generate TYPE-2 message
10445	Failed to generate TYPE-3 message
10450	Server process is unresponsive; unable to connect at this time.
10451	Server process is unresponsive; unable to send trigger to all specified flowcharts at this time.
10452	Server process is unresponsive; unable to reconnect at this time.
10453	<p>Server process is unresponsive; unable to complete requested action at this time.</p> <p>Possible causes</p> <p>The Unica Campaign server is busy processing another request.</p> <p>Suggested solutions</p> <p>Ask your system administrator to check that the Unica Campaign server machine has enough CPU or memory resource.</p>
10454	Server process is updating flowchart data; unable to complete requested action at this time
10501	INTERNAL ERROR: SRunMgr RunProcess thread already running.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
10502	Process's run was cancelled by Run Manager destruction.
10530	Invalid Campaign Code format.
10531	Invalid Offer Code format.
10532	Failed to generate Campaign Code.
10533	Failed to generate Offer Code.
10534	Invalid Treatment Code format.
10535	Failed to generate Treatment Code.
10536	Invalid Cell Code format.
10537	Failed to generate Cell Code.
10538	Invalid Version Code format.
10539	Failed to generate Version Code.
10540	Campaign code format contains invalid characters.
10541	Cell code format contains invalid characters.
10542	Treatment code format contains invalid characters.
10550	HTTP communication error.
10551	Invalid response from ASM Server.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10552	ASM Server: unknown error.
10553	ASM Server: Login is invalid.
10554	ASM Server: Error while inserting into db.
10555	ASM Server: Error while trying to map ASM objects.
10556	ASM Server: Error because object already exists.
10557	ASM Server: Password Expired.
10558	ASM Server: Password is too short.
10559	ASM Server: Password not properly formed.
10560	Internal Error: parsing data returned from ASM Server.
10561	ASM Server: Valid login required.
10562	ASM Server: Group name required.
10563	ASM Server: Action not supported.
10564	ASM Server: Exceeded maximum allowable pw attempts.
10565	ASM Server: Password does not contain minimum number of numerics.
10566	ASM Server: Password cannot be same as login.
10567	ASM Server: Password cannot be reused from before.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10568	ASM Server: User logon is disabled.
10569	ASM Server: Password does not contain minimum number of letters.
10570	ASM Server: Password cannot be blank.
10571	ASM Server: Password is incorrect.
10572	This action requires proper privileges.
10573	ASM Server: Internal System Error.
10575	Flowchart in pending or rejected state.
10576	Internal Error: ASM client module is not initialized.
10577	Login required to query for db credentials.
10578	Security data integrity error.
10580	HTTP communication error
10581	Invalid response from Unica Deliver Server
10582	Unica Deliver Server: unknown error
10583	Unica Deliver Server: Internal System Error
10584	Unica Deliver server url is not set.
10585	Internal Error: parsing data returned from Unica Deliver Server

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10586	Error returned from Unica Deliver Server.
10590	setuid failed.
10591	setgid failed
10600	Internal Error: Cell already initialized
10601	Internal Error: Source Cell not initialized
10603	Internal Error: Invalid cell ID.
10604	Internal Error: Invalid field index.
10605	Audience ID field not defined.
10606	Internal Error: Cannot find table manager.
10607	Invalid Table ID.
10608	Operation not allowed while cell is being accessed.
10612	Internal Error: Derived Field not found.
10613	<p>Field not found.</p> <p>Possible causes:</p> <p>The field no longer exists because of table mapping change.</p> <p>The audience level changed.</p> <p>The field was deleted.</p> <p>Suggested solutions: Re-configure the process box to reference a different field.</p>

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10616	Internal Error: Derived variable not initialized.
10617	Internal Error: Expression returns multiple columns.
10619	Internal Error: Invalid row index.
10620	Ambiguous field name.
10621	Internal Error: selected fields not yet computed.
10624	Internal Error: Access object became invalid.
10625	Internal Error: No Data Source selected for Raw SQL Query.
10629	Error writing temporary file on Unica Campaign server.
10630	Operations not allowed on different audience levels.
10632	Reference to stored query not found.
10633	Internal Error: Derived variable should not contain data.
10634	Incompatible sort order detected. Set \enable_select_order_by=FALSE\ in dbconfig.lis.
10635	Cannot resolve reference to stored query: Stored Query table not yet mapped.
10636	User Variable not defined.
10637	Cell results are missing; preceding processes must be rerun.
10638	Invalid value in 'counts' field.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10639	Internal Error: Bad STCell _Select State.
10641	Derived variable name conflicts with an existing Persistent Derived Field.
10642	Temp table not available for <TempTable> token.
10643	Temp table has too many rows
10644	Temp table does not have enough rows
10645	<OutputTempTable> token used, but data source configuration does not allow temp tables.
10646	Cannot create temp table on system database. Check datasource configuration to make sure temp tables are allowed and bulk insert or database loader is enabled.
10661	HTTP communication error with Instance Manager
10700	Incompatible field type or width.
10800	Duplicate parameter name for Custom Macro.
10801	Missing parameter name for Custom Macro.
10802	Incorrect number of parameters for Custom Macro.
10803	Illegal parameter name for Custom Macro.
10804	Name conflicts with an existing Custom Macro.
10805	Missing parameter for Custom Macro.
10806	Parameter name is a reserved word.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
10807	Illegal Custom Macro name.
10808	Name conflicts with an existing Unica Macro.
10809	Parameter used within Custom Macro expression is not part of Macro definition.
10810	Audience level is not defined in selected ACO session.
10811	Proposed contact table is not defined in selected ACO session.
10812	Proposed offer attribute table is not defined in selected ACO session.
10813	Optimized contact table is not defined in selected ACO session.
10820	INTERNAL DYNAMIC CAST ERROR
10821	Invalid configuration for ODS Campaign.
11001	Internal Error: SendMessage error.
11004	Internal Error.
11005	Internal Error: Unknown report type.
11006	The flowchart is being accessed by another user.
11100	Memory allocation error.
11101	Internal Error: Unknown function tag.
11102	Internal Error: Unknown class name in IDtoPtr.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11104	Internal Error: Bad magic number in SCampaignContextConfig.
11105	File name not specified.
11107	Server campaign context internal error.
11108	Internal Error: Cannot lock report.
11109	Table is not defined.
11110	Environment variables not set.
11111	Internal Error: Error getting field info.
11112	Invalid Password.
11113	Flowchart name not unique or empty.
11114	Campaign code not unique.
11115	Active flowchart Cannot be deleted.
11116	The specified file is not a Unica Campaign flowchart file.
11117	Deletion of an old flowchart file is not supported. Please delete it manually.
11119	Unable to write unica_tbmgr.tmp file in tmp directory.
11120	Unable to rename unica_tbmgr.bin in conf directory.
11121	Unable to copy unica_tbmgr.tmp to unica_tbmgr.bin file.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11122	Unable to read unica_tbmgr.bin file in conf directory.
11128	Operation disallowed in configuration.
11131	Invalid template file format.
11132	XML initialization error.
11133	DOM creation error.
11134	DOM parse error.
11135	Internal Error: Unknown user variable
11136	Server campaign context cell locking error.
11137	Server campaign context file open error.
11138	The named user exists already.
11139	No user list table is mapped to the admin. session.
11140	User not found.
11141	Wrong password.
11142	File read error.
11143	Empty user variable.
11144	Flowchart name and Campaign code not unique.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11145	authentication_server_url missing in unica_acsvr.cfg file.
11146	Invalid user variable.
11147	User Variable not found.
11148	Changes to the virtual memory setting are not allowed.
11150	Unable to create folder file. Please check your OS privilege.
11151	Unable to delete folder file. Please check your OS privilege.
11152	Unable to rename folder/campaign/session file. Please check your operating system privilege.
11153	Unable to create campaign/session file. Please check your OS privilege.
11154	Unable to delete campaign/session file. Please check your OS privilege.
11155	Unable to move folder/campaign/session file. Please check your OS privilege.
11156	Failed to authenticate datasource.
11157	Effective date is later than expiration date
11158	Unable to open campaign/session file. Please check your OS privilege.
11159	Unable to read Log file. Please check your OS privilege.
11160	Cannot display log. Log file name not specified
11161	Operation is not allowed while the flowchart is running.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11162	Log file does not exist. Change logging levels if you would like to see more logging information.
11163	Campaign/session file does not exist in the file system.
11164	Server Stored List Internal Error.
11165	Stored List unknown function tag.
11166	Invalid security policy.
11201	Container internal error (1).
11202	Container internal error (2).
11203	Container data loading error.
11230	Unable to create transcoders between specified encoding and UTF-8.
11231	Unable to transcode text value.
11232	Cannot determine name of local host.
11251	New passwords mismatched. Please re-type.
11253	Stack overflow during sort operation.
11254	Too many arguments passed to command-line parser.
11255	Unbalanced quotes in command or config-file parameter.
11256	Unable to open flowchart LOG file for append.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11257	Unable to write to flowchart LOG file.
11258	Unable to rename flowchart LOG file.
11259	Invalid Multibyte or Unicode character encountered.
11260	Illegal or duplicate campaign code.
11261	Invalid old password
11262	New read/write passwords mismatched.
11263	New read-only passwords mismatched.
11264	Invalid read/write password.
11265	Invalid read-only password.
11266	Password must contain at least 6 characters.
11267	Report registered.
11268	Report name missing.
11269	New passwords mismatched.
11270	Cannot create temporary file on client computer.
11271	Error reading temporary file on client computer.
11272	Error writing temporary file on client computer.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11273	Set new configuration as default?
11274	Unmap the selected table(s)?
11275	No fields selected.
11276	No flowchart name. Execution Checkpoints will not be performed.
11280	Server version is newer than client version. Upgrade client installation?
11281	Server version is older than client version. Downgrade client installation?
11282	Installation executable retrieved but unable to execute.
11283	About to clear flowchart log. Are you sure?
11284	Help topic not found.
11285	Error parsing help topics file.
11286	flowchart is recovered from auto-save file.
11287	Error loading bitmap.
11288	Settings changed. Save catalog now?
11289	Flowchart is already open. Disconnect current user and connect?
11290	Flowchart must be saved before this operation can proceed.
11300	Invalid field name. Look at the end of message for invalid field name.  Possible causes:



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
	<p>The field no longer exists because of table mapping change.</p> <p>The audience level changed.</p> <p>The field was deleted.</p> <p>Suggested solutions: Re-configure the process box to reference a different field.</p> <p>Invalid Field Name=</p>
11301	Invalid field index.
11302	No more records.
11303	Operation not allowed while table is being accessed.
11304	Locked tables cannot be removed.
11305	Invalid table ID.
11306	ParseTree context in use.
11307	Random access of basetable by parsetree not allowed.
11308	Invalid table index.
11309	Invalid key index.
11310	Index key not initialized.
11311	Entry not found in dimension table.
11312	ID field not specified.
11313	Invalid Table access.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11314	Data already imported.
11315	Internal error: VFSYSTEM missing
11316	Input files not yet specified.
11317	No data.
11318	Modification not yet started.
11319	Entry into index field is not unique.
11320	<p>Unable to create lock file in conf directory.</p> <p>Possible causes: The Unica Campaign server cannot lock the dummy_lock.dat file</p> <p>Suggested solutions: Ask your system administrator to check that the file is not locked by another process. If another process is not locking the file, ask the Unica Campaign administrator to reboot the Unica Campaign server to remove the lock.</p>
11321	Internal table error
11322	Unknown function tag
11323	Data Dictionary file name not specified.
11324	Function or Operation not supported.
11325	'dbconfig.lis' file not found.
11326	Dimension table does not have a key field.
11327	ID of new version conflicts with existing versions.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11328	Cannot open table catalog file.
11329	Too many duplicate ID's to perform table join.
11330	Unable to delete template file
11331	Unable to delete catalog file.
11332	Error parsing Data Dictionary file: invalid format.
11333	Error converting text data to numeric.
11334	Field width too small to hold converted numeric value.
11335	Field width too small to hold source text data.
11336	Accessed table not mapped.
11337	Duplicate ID encountered in normalized table.
11338	Internal error: Invalid temp table.
11339	Audience definition not compatible: wrong number of fields.
11340	Audience definition not compatible: type mismatch.
11341	Name of new version conflicts with existing versions.
11342	Field not found. Data dictionary has changed.
11343	XML table catalog file is invalid.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11344	Loader command exited with an error status.
11345	Table schema has changed; please remap the table.
11346	No results for queue table.
11347	Internal error, wrong return format.
11348	Internal error loading catalog.
11349	No catalog loaded.
11350	Internal error connecting to table.
11351	Not connected to a table.
11352	Invalid keyword in dbconfig.lis file.
11353	Invalid UDI connection.
11354	Internal Error: base table not set.
11355	Invalid table name.
11356	DOM creation error.
11357	DOM parse error.
11358	Unable to import duplicate system table entry.
11359	Unable to lock system table.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11360	PACKED DECIMAL field type is supported for export only.
11361	This operation is not supported.
11362	Too many fields returned by SQL expression.
11363	Data field returned by SQL expression does not agree with user specification.
11364	Unknown database specification in raw SQL Custom Macro.
11365	<p>Raw SQL Custom Macro returning only ID list is not allowed in this context: &lt;FunctionName&gt;.</p> <p>Probable cause: There is already a custom macro with the same name as one of the default function names. For example, if you have a custom macro named "DATE" already present in your list, then if you try to use the default function "Date" in any other custom macro you would receive this error. To resolve the issue, try searching for an existing custom macro that has same name as the function name in the error message. Delete or rename that custom macro, then try using the function again.</p>
11366	Segment not found.
11367	Temp table not available for <TempTable> token.
11368	Contact History Table is not yet defined for this audience level.
11369	Response History Table is not yet defined for this audience level.
11370	Dimension Element expression is missing.
11371	Ambiguous bin definition.
11372	Custom Macro returned wrong number of fields.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11373	Custom Macro result fields are not compatible with the current audience level.
11374	Dimension Element name is not unique across all levels.
11375	Unknown Dimension Name.
11376	Unknown Dimension Element.
11377	Missing database specification for raw SQL Custom Macro.
11378	Campaign code not unique.
11379	Root dimension element missing in XML file.
11380	Error converting date from one format to another.
11381	Not enough privileges to use RAW SQL in Dimensions.
11382	Syntax error: Missing AND/OR operator.
11383	Syntax error: Extra AND/OR operator at end of selection criteria.
11384	Field not compatible: Numeric field expected.
11385	Field not compatible: Date field expected.
11386	Error returned from UDI server.
11387	Internal ID would exceed limit.
11388	Cannot open Segment data file.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11389	Segment data file error: Invalid header.
11390	Internal Error: Invalid Segment (Data file name is blank
11391	Error accessing segment data.
11392	Cannot perform table join unless tables are on the same database.
11393	Unable to add entry to non-persistent queue
11394	Audience level is reserved, cannot add.
11395	Audience level is reserved, cannot remove.
11396	Internal Error: Invalid optimized contact table name.
11397	Field data exceeded the table mapping width for this field. Remap the table and manually increase the field width before running the flowchart.
11398	Post temp table create run script completed with error(s).
11399	Cannot allocate an ID for a new object because the allocator is busy.
11400	Temp table not available for <OutputTempTable> token.
11401	Invalid audience level definition.
11402	Missing audience field definition.
11403	Invalid or missing audience field name.
11404	Duplicate audience field name.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11405	Invalid or missing audience field type.
11408	INTERNAL ERROR: Invalid ID.
11409	INTERNAL ERROR: Wrong DAO type.
11410	DAO INTERNAL ERROR.
11411	INTERNAL ERROR: System DAO factory is not initialized yet.
11412	INTERNAL ERROR: Unknown DAO implementation requested.
11413	INTERNAL ERROR: Invalid type detected in DAO transfer.
11414	Insert operation only supported on single table.
11415	Update operation only supported on single table.
11416	Delete operation only supported on single table.
11417	SQL query returned multiple records when unique record is expected.
11418	Default contact status not found in ContactStatus table.
11419	Contact History table must be mapped before Detailed Contact History table.
11420	Offer not found in the system.
11435	Delimited file record length exceeds allowable maximum. Remap the table and manually increase field widths as necessary before running the flowchart.
11500	Internal Error: Not a valid table in the database.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11501	Internal Error: No table has been selected.
11502	Selected table has no field entries.
11503	Invalid column index.
11504	Invalid column name.
11505	Invalid data source.
11506	Selected table is invalid or corrupted.
11507	Insufficient memory.
11508	Database row deletion error.
11509	Error in processing SQL query.
11510	No data returned - check query.
11511	No matching row was found in the query result.
11512	No more rows in the database.
11513	Error inserting row into database table.
11514	Bad database ID column.
11515	Error updating database table.
11516	Error creating new database table.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
11517	Number of columns not correct for this type of query.
11518	Database connection error.
11519	Error in getting results from database.
11520	Unknown database type for dataSources.
11521	Internal Error: Incorrect state for query results.
11522	Invalid database connection (user is not logged in to DB).
11523	First unique ID not set.
11524	Invalid data type for this column.
11525	Query has no FROM clause.
11526	Query uses alias.
11527	Internal Error: Error in database temporary table.
11528	Database Error.
11529	INTERNAL ERROR: No threads available to run query.
11530	Invalid property for dataSources
11531	Catalog/Template contains different DB login(s).
12000	Contact History Table not specified.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
12001	Customer ID not specified.
12002	Offer ID not specified.
12003	Channel field not specified.
12004	Date field not not specified.
12005	No proposed contacts table templates.
12006	No tables available for template. Template tables must be mapped at Customer level and must contain the required Offer, Channel, and Date fields.
12007	No tables available for opt-in/opt-out tables. Opt-in/opt-out tables must be mapped at Customer level.
12008	Opt-in/Opt-out tables not specified. \"Customer In...\" rules will not be available.
12009	Offer Table not specified.
12010	Offer Name field not specified. Offer ID will be used for display.
12011	Channel Table not specified.
12012	Channel Name field not specified. Channel ID will be used for display.
12015	Field name(s) of Offer audience level in template table does not agree with Contact History Table.
12016	Field name(s) of Offer audience level in offer table does not agree with Contact History Table.
12017	No tables available for offer table. Offer table must be mapped at Offer level.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
12018	No tables available for channel table. Channel table must be mapped at the Channel level.
12019	Killing the server process will lose all work since the last save. Are you sure?
12020	Window creation failed.
12021	Remove the following tables associated with this audience level?
12022	Remove the selected dimension hierarchies?
12023	Flowchart is in use. Do you still want to continue?  If you click Yes, changes made by other user will be lost.
12024	Remove the selected audience level?
12025	Audience name already exists.
12026	This flowchart has been modified or removed by another user. You will be switched to Summary tab now. All changes after the last save will be discarded.
12027	This flowchart needs to be updated. Click OK to update now. You will need to redo the last action after update is completed.
12028	Object is either initializing or has failed to initialize. You may try this operation again.
12029	Remove the selected items?
12030	You have chosen to cancel connection to Unica Campaign system tables. You will be switched to Summary tab now.
12031	Cannot proceed without connection to Unica Campaign system tables.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
12032	This table is only supported when Interact is installed.
12033	Failed to load flowchart. Retry?
12034	HTTP Session has timed out. Click OK to log back in.
12035	Flowchart control is not compatible. Browsers need to close to download the lower version. Please close all other browsers manually and click OK to close this browser. On browser restart, the control will get downloaded automatically.
12036	There are still other browsers running. Please close them before clicking OK.
12037	Field name contains illegal character.
12038	Audience level name not specified.
12039	Audience field(s) not specified.
12040	No errors detected in flowchart configuration.
12041	This running flowchart has been paused by another user.
12206	Cannot navigate up a directory: already at root.
12207	Cannot create directory; check log file for detailed error information.
12301	Merge process internal error.
12303	Merge process connecting from process error.
12304	Merge process cell locking error.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
12305	Merge process stopped by user.
12306	Merge process cell operation error.
12307	Merge process getting source cell error.
12308	Merge process not configured.
12309	No input cell is chosen.
12310	No input cell is used.
12311	Selected input cells have different audience levels.
12312	Missing source cell(s). Input connection may be invalid.
12401	Execute Internal Error (1)
12600	INTERNAL ERROR: SReport
12601	Report in use; unable to delete.
12602	INTERNAL ERROR: Invalid report ID.
12603	INTERNAL ERROR: Invalid report type saved.
12604	INTERNAL ERROR: Invalid report cell ID.
12605	INTERNAL ERROR: Report not initialized before run.
12606	INTERNAL ERROR: Missing value.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
12607	INTERNAL ERROR: Unable to lock report.
12608	INTERNAL ERROR: Invalid field specified.
12609	Report cannot be created without any cells.
12610	INTERNAL ERROR: No more cell records available.
12611	Report name conflicts with another registered report.
12612	Cannot open HTML file for write.
12613	Field type does not match internal setting. Table may need to be remapped.
12614	Report name is empty.
12615	Command not allowed in Reader mode
13000	An error occurred while parsing the response from the webapp.
13001	The client id is missing in the response from the webapp.
13002	The resolution id is missing in the response from the webapp.
13003	The <code>iscomplete</code> flag in the response from the webapp, has a bad value
13004	An unknown error code has been returned from the webapp.
13005	HTTP communication error
13006	The response required an <code>iscomplete</code> flag, but it was missing.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13101	Internal error.
13104	Cell locking error.
13110	Process not configured.
13111	Unknown function tag.
13113	Report locking error.
13114	Profile report generation error.
13115	Table locking error.
13116	No input cell.
13117	No input selected.
13118	Missing selection criteria.
13119	No data source is selected.
13120	Selected tables have different audience levels.
13121	Audience level not specified.
13122	DOM creation error.
13123	DOM parse error.
13124	Unknown parameter.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13125	Invalid parameter value.
13131	Database authentication required.
13132	String conversion error.
13133	No fields are selected for extract.
13134	Duplicate output name in fields to extract.
13135	No skip duplicate field selected.
13136	Command not allowed in Reader mode
13137	No source table selected.
13138	Error selecting based on dimension hierarchy: No tables mapped at selected Segment's audience level.
13139	Missing table mapping for selected optimize session.
13140	Missing CustomerInsight Selection.
13141	Selected CustomerInsight Selection is not valid.
13145	Missing NetInsight Selection.
13146	Selected NetInsight Selection is not valid.
13156	<p>Error Received in IBM Digital Analytics Response. Refer to log for more details.</p> <p>This error can occur in the IBM Digital Analytics Segment Selection dialog box when configuring a Select process in a flowchart. It indicates that the credentials assigned to the UC_CM_ACCESS data source are incorrect.</p>

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
13200	Contact process memory allocation error.
13201	Contact process internal error.
13203	Contact process connecting from process error.
13204	Contact process locking cell error.
13205	Contact process stopped by user.
13206	Contact process locking contact table error.
13207	Contact process locking version table error.
13208	Contact process getting cell information error.
13209	Contact process getting table information error.
13210	Contact process locking table error.
13211	Contact process unknown function tag error.
13212	Contact process GIO open error.
13213	Contact process locking report error.
13214	More information is needed for a creative piece.
13215	Exactly one variable cost item should be chosen.
13216	Conflict variable cost items.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13217	More information is needed for a version.
13218	At least one creative piece have to be chosen.
13219	At least one response channel have to be chosen.
13220	One contact channel has to be chosen.
13221	The chosen ID is not unique.
13223	Contact ID is not unique.
13224	Treatment page: No source cell.
13225	Treatment page: Contact ID is not selected.
13226	Treatment page: No version is selected.
13227	Contact list page: No export table is selected.
13228	Contact list page: No summary file is chosen.
13229	Contact list page: No export fields are chosen.
13230	Tracking page: Frequency of update is not chosen.
13231	Tracking page: Monitor period cannot be zero.
13232	Responders page: No responder table is chosen.
13233	Unreachable page: No unreachable table is chosen.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13234	Log page: No table is chosen to log contacts.
13235	Log page: No fields are chosen to log contacts.
13236	Log page: No table is chosen to log responders.
13237	Log page: No fields are chosen to log responders.
13238	Log page: No table is chosen to log unreachable.
13239	Log page: No fields are chosen to log unreachable.
13240	Contact process getting cell field information error.
13241	Contact list page: No trigger is specified.
13242	Contact list page: no sort field is chosen.
13244	Invalid field.
13246	Double to string conversion error.
13248	Contact list page: No export file is chosen.
13249	Contact list page: No delimiter is specified.
13250	Selected tables have different audience levels.
13251	Contact list page: No export dictionary file is chosen.
13252	Log page: No file is chosen to log contacts.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13253	Log page: No delimiter is specified for contacts.
13254	Log page: No dictionary file is specified for contacts.
13255	Log page: No file is chosen to log responders.
13256	Log page: No delimiter is specified for responders.
13257	Log page: No dictionary file is specified for responders.
13258	Log page: No file is chosen to log unreachable.
13259	Log page: No delimiter is specified for unreachable.
13260	Log page: No dictionary file is specified for unreachable.
13261	Contact List Page: Selected Data Export Filename contains Invalid Path
13262	Contact List Page: Selected Data Dictionary for Export File contains an Invalid Path.
13263	Contact List Page: No skip duplicate field is chosen.
13264	Contact List Page: Update Records require a base table with audience same as input.
13265	Log Page Contact: Update Records require a base table with audience same as input.
13266	Log Page Responders: Update Records require a base table with audience same as input.
13267	Log Page Unreachable: Update Records require a base table with audience same as input.
13268	Tracking page: No trigger is specified.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13269	Responders page: No responder query is specified.
13270	Responders page: No data source is selected.
13271	Unreachable page: No unreachable query is specified.
13272	Unreachable page: No data source is selected.
13273	Selected source cells have different audience levels.
13274	Unknown parameter for Contact Process.
13275	Invalid parameter value for Contact Process.
13276	Version Name is not unique.
13277	Empty or Duplicate Cell Codes.
13278	About to modify a version used by another flowchart.
13279	Log Page Contact: No skip duplicate field is chosen.
13280	Log Page Responders: No skip duplicate field is chosen.
13281	Log Page Unreachable: No skip duplicate field is chosen.
13282	Contact process DOM creation error.
13283	No data source is selected.
13284	Contact List Page: Selected Data Dictionary File does not exist.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
13285	Log page: No fields are chosen to log contacts.
13286	Command not allowed in Reader mode
13301	Internal error.
13304	Cell locking error.
13310	Profile report generation error.
13311	Unknown function tag.
13312	Report locking error.
13313	No input is selected.
13314	No field is selected.
13315	No query is specified.
13316	No data source is specified.
13317	Name not unique.
13318	No table is selected.
13320	Unknown parameter.
13321	Invalid parameter value.
13322	Name not specified.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13323	Invalid name.
13324	Command not allowed in Reader mode
13400	Schedule process memory allocation error.
13401	Schedule process internal error.
13403	Connecting from process error.
13404	Cell locking error.
13405	Process stopped by user.
13408	Date format error.
13409	Time format error.
13410	Total schedule period is zero.
13411	No schedule to run is chosen.
13412	Run on time needs times.
13413	Run on triggers needs triggers.
13414	Need output triggers.
13415	Elapsed time is zero.
13416	Additional wait must work with one of the first three run options.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13417	Schedule run time(s) outside the schedule period.
13418	Invalid time format.
13419	At least one of the custom run options has to be selected.
13420	Delay exceeds the total schedule period.
13421	Invalid time. Start time has expired.
13422	Input queue table not selected.
13423	Selected queue table is invalid.
13424	Cannot use 'Run Selected Process' on this process
13501	Sample process internal error.
13503	Sample process connecting from process error.
13504	Sample process cell locking error.
13505	Sample process stopped by user.
13506	Sample process locking sample table error.
13507	Sample process locking version table error.
13508	Sample process getting source cell error.
13510	Sample process unknown function tag.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13511	Sample process not configured.
13512	Sample process output cell size exceeding input cell size.
13513	No source cell is chosen.
13514	No ordering field is chosen.
13515	Name not unique.
13516	Unknown parameter for Sample Process.
13517	Invalid parameter value for Sample Process.
13518	Sample name not specified.
13519	Invalid sample name.
13520	Command not allowed in Reader mode
13521	Sample size not specified.
13601	Internal error.
13602	GIO open error.
13603	The named trigger doesn't exist.
13604	Trigger name not specified
13605	Trigger completed with error(s).

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
13701	Score process internal error.
13703	Score process connecting from process error.
13704	Score process cell locking error.
13705	Score process stopped by user.
13706	Score process cell operation error.
13707	Number of models can't be zero.
13708	Score process GIO open error.
13709	Environment variables not set.
13716	Score field prefix is missing.
13717	An internal model is not selected.
13718	An external model is not chosen.
13719	Model variables are not completely matched.
13720	No input is selected.
13721	The number of models is zero.
13723	Score field prefix not unique.
13724	The external model (rtm) file is incompatible with current SCORE configuration.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13725	Invalid field.
13726	dbscore process completed with error(s).
13727	Unknown parameter for Score Process.
13728	Cannot find external model file.
13729	Cannot get model information. Model file may be invalid.
13730	Command not allowed in Reader mode
13801	SelectOpt process internal error.
13803	SelectOpt process connecting from process error.
13804	SelectOpt process cell locking error.
13805	SelectOpt process stopped by user.
13806	SelectOpt process cell operation error.
13807	SelectOpt process table locking error.
13809	SelectOpt process report locking error.
13812	dbscore process completed with error(s).
13825	Duplicate Personalization Field Name specified.
13833	Personalization Field Display Name is blank.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13834	Personalization Field Display Name contains invalid character.
13901	Internal error.
13903	Connecting from process error.
13904	Cell locking error.
13905	Process stopped by user.
13906	Cell operation error.
13907	Table locking error.
13909	Unknown function tag error.
13910	Report locking error.
13911	No input is selected.
13912	No export table is selected.
13913	No fields are selected for export.
13914	No sort field is chosen.
13915	Invalid field name.
13917	Invalid field name.
13918	No export file is selected.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
13921	String conversion error.
13923	Selected cells have different audience levels.
13924	No delimiter is specified.
13925	No export data dictionary file name is specified.
13926	Selected Data Export Filename contains Invalid Path
13927	Selected Data Dictionary for Export File contains an Invalid Path.
13928	No skip duplicate field selected.
13929	Update Records require a base table with audience same as input.
13930	Snapshot process DOM creation error.
13931	Unknown parameter for Snapshot Process.
13932	Invalid parameter value for Snapshot Process.
13933	Empty or Duplicate Cell Codes.
13934	Selected Data Dictionary File does not exist.
13935	Command not allowed in Reader mode
13936	Kafka configuration missing.
13937	Either of Number of partitions / Number of Replica / Retention Period is not configured.
13938	Number of Replicas Must be less than or equal to Number of Brokers.
14001	Model process internal error.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
14003	Model process connecting from process error.
14004	Model process cell locking error.
14005	Model process stopped by user.
14006	Model process cell operation error.
14008	Model process report locking error.
14009	No responder cell is selected.
14010	No non-responder cell is selected.
14013	Model file name is not selected.
14014	At least one variable has to be used for modeling.
14015	No responder and non-responder cells are selected.
14016	udmerun process completed with error(s).
14017	Selected Model Filename contains Invalid Path
14018	Command not allowed in Reader mode
14101	EvalOpt process internal error.
14103	EvalOpt process connecting from process error.
14104	EvalOpt process locking cell error.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
14105	EvalOpt process stopped by user.
14106	EvalOpt process cell operation error.
14107	EvalOpt process table locking error.
14108	EvalOpt process unknown function tag.
14110	EvalOpt process locking report error.
14111	No responder cell is chosen.
14112	No non-responder cell is chosen.
14113	No responder field is chosen.
14114	No non-responder field is chosen.
14115	Unknown parameter for EvalOpt Process.
14116	Set number not specified.
14117	Set number out of range.
14118	Set name is empty.
14119	Not supported options.
14120	Command not allowed in Reader mode
14202	PopulateSegment internal error.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
14203	PopulateSegment cell locking error.
14204	PopulateSegment process unknown function tag.
14205	No input is selected.
14206	Segment name not unique within the specified folder.
14207	Segment name is not specified.
14208	Invalid segment name.
14209	Invalid security policy.
14210	Security policy not specified.
14301	TestOpt process internal error.
14303	TestOpt process connecting from process error.
14304	TestOpt process cell locking error.
14305	TestOpt process stopped by user.
14306	TestOpt process cell operation error.
14307	TestOpt process table locking error.
14308	No source cell is selected.
14309	Number of tests to be optimized is zero.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
14310	One of the financials is not configured.
14317	Report locking error.
14319	Error to get selected field index.
14320	Probability field value exceeds 1.0.
14321	Invalid field.
14322	No probability field is selected.
14323	No treatment is chosen.
14324	Command not allowed in Reader mode
14501	Custom macros internal error.
14502	The Custom Macro expression type is not specified.
14503	The custom macro name is empty.
14504	The Custom Macro expression is missing.
14505	Custom Macro unknown function tag.
14701	Stored field internal error.
14703	No variable name is specified.
14704	No expression is specified.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
14705	There is already a stored derived field of the same name.
14706	Stored field unknown function tag.
14901	List box selection error
14902	Too many items selected
14903	No items selected
14905	Selection not found
14906	Unrecognized tree view operation
14907	No cost information selected
14908	Dialog initialization error
14909	Specified cell name (process name + output cell name) is too long
14912	Creative ID may only contain alphanumeric and underscore
14913	Output Cell Names not unique.
14914	Overwrite current info?
15101	Dialog initialization error
15201	List box selection error
15202	Dialog initialization error

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
15203	Specified cell name (process name + output cell name) is too long
15204	Invalid cell size limit.
15301	Dialog initialization error
15501	String not found
15502	Minimum rate > maximum rate
15503	Dialog initialization error
15504	Invalid output cell name
15701	Dialog initialization error
15702	Specified cell name (process name + output cell name) is too long
15801	Selected string not found
15802	Tree expansion error
15803	Dialog initialization error
15804	Segment name not specified
15805	Segment names Cannot be specified
15901	Selected string not found
15903	Dialog initialization error

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
15904	Specified cell name (process name + output cell name) is too long
15905	List box selection error
15906	Invalid cell/record size limit.
15907	Existing expression based on tables and fields will be lost.
15908	Existing criteria based on dimension hierarchy will be lost.
16001	Dialog initialization error
16002	No list box selection found
16051	Stored trigger internal error.
16053	The trigger name is empty.
16054	The trigger command is empty.
16055	There is already a trigger defined under the same name.
16056	Stored trigger unknown function tag.
16101	Selection error
16102	Multiple selection error
16103	No items selected
16104	Selection style error

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
16105	Selection not found
16106	Dialog initialization error
16201	Dialog initialization error
16202	ListBox selection error
16203	Specified cell name (process name + output cell name) is too long
16302	Source table is not yet mapped.
16303	DimensionInfo Internal Error: Unknown Function.
16304	DimensionInfo Internal Error.
16305	Illegal number of levels.
16306	Required field missing in source table. It needs to be remapped.
16400	No data base source defined
16401	No table selected
16402	Internal Error: no table manager
16403	Bad Unica Campaign Table index
16404	Internal Error
16405	Internal Error: New Table Unknown function

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
16406	No file name specified
16407	No data dictionary specified
16408	The selected table has no defined fields
16409	Internal Error: No table created
16410	No name specified for new table
16411	Need a user name and password for the database
16412	The database type is not currently supported
16413	Table is not a base table -- no relations allowed
16414	Bad field index
16415	Record table ID not specified
16416	Internal Error: No dimension table with this name
16417	Table is not a dimension or general table
16418	Internal Error: No base table with this name
16419	Entry point not valid for this operation
16420	Mapping to an existing table not valid for this operation
16421	Error creating new Flat File

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
16422	Error - no file/table option selected
16423	Error - no database selected
16424	Error - table selected is invalid
16425	Error - bad key field index
16426	Error - blank key field name
16427	Error - Table Name is duplicate or invalid
16428	Field name must begin with a letter and may only contain alphanumerics and underscore
16429	Dimension table ID not specified
16430	Duplicate field names specified
16431	Table name must begin with a letter and may only contain alphanumerics and underscore
16432	Error - Dimension Name is duplicate or invalid
16433	Error - Folder not found
16501	Derived field internal error.
16503	Derived field unknown function tag error.
16504	Derived field not exist.
16505	Derived field report locking error.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
16506	Derived field table locking error.
16507	Derived field cell locking error.
16508	Derived field already exists.
16509	Derived field getting all field information error.
16601	Internal Error.
16603	The authorized process's schedule period has expired.
16701	Selected string not found
16702	Parent window not found
16703	Filename not specified
16704	No fields selected
16705	Dialog initialization error
16706	Specified Source File does not exist
16707	Remapping system table - are you sure?
16708	Overwrite old definition?
16709	Syntax check OK
16710	Discard changes to current expression?

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
16711	Specified Dictionary File does not exist
16712	Derived variable name not specified
16713	Query name not specified
16714	Trigger name not specified
16715	No field selected
16716	Illegal field name
16717	Invalid name: name must begin with a letter and must contain only alphanumerics or '_'
16718	Remove entry?
16719	Remove folder? All folder information (sub-folders etc.) will be lost.
16720	Name not specified
16721	Invalid Data Dictionary file. It may be a directory.
16722	Data Dictionary file exists. Do you want to overwrite it?
16723	File Not Found
16724	Overwrite Existing File?
16725	Audience level not specified
16726	Audience ID Field(s) not specified

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
16727	Duplicate Audience ID Fields
16728	Invalid run state - operation will be terminated
16729	No table(s) selected
16730	No cell(s) selected
16731	Selected tables have different audience levels
16732	Selected cells have different audience levels
16733	Audience level is already defined as the table's primary audience
16734	Audience level is already defined for this table
16735	Base Table's related fields not compatible with Dimension Table's key fields
16736	The file path length has exceeded the allowed limit
16737	No fields are checked
16738	Table or field name not specified
16739	Derived variable name conflicts with Unica Campaign Generated Field
16740	Required value missing.
16741	Unable to translate existing expression for Point & Click mode. Restart with empty expression?
16742	Unable to translate expression for Point & Click mode. Switch to Text Builder mode?

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
16743	Current expression is not valid. Switch to Text Builder mode anyway?
16744	Tree expansion error
16745	Folder already exists.
16746	About to execute trigger command. Are you sure?
16747	Derived variable name conflicts with an existing Persistent Derived Field
16748	No delimiter is specified.
16750	No derived variable name has been specified.
16751	Selected segments have different audience levels
16752	Illegal field name. User Variable values may only be set in a Select process
16753	The full name is too long.
16754	An administrator must define at least one audience level before a new table can be created.
16755	Remap of Optimized List table is not permitted.
16756	Audience ID Field not compatible: type mismatch.
16757	Output cell name is too long.
16758	Process name is too long.
16759	Output cell name is empty.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
16760	Security policy not specified.
16761	Security policy is restored to the original policy.
16762	Start or end date missing.
16763	Invalid date specification.
16764	No dates selected.
16765	End date cannot be earlier than start date.
16769	Data Packaging internal error.
16770	Package name not specified.
16771	View Log permission is required to access log entries.
16772	Dictionary file name cannot be the same as data file name.
16773	Data Package folder already exists. Existing content within that folder will be deleted.
16901	Stored template internal error.
16903	The template name is empty.
16906	Stored template unknown function tag.
16908	Templates directory does not exist.
16909	Templates directory is invalid.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
16910	There is already a stored template of the same name.
17001	Stored catalog internal error.
17003	The catalog name is empty.
17006	Stored catalog unknown function tag.
17008	Catalogs directory does not exist.
17009	Catalogs directory is invalid.
17012	The catalog file extension is invalid. Only 'cat' and 'xml' are allowed.
17013	Target catalog file extension is not same as original.
17014	Unica Campaign Data Folder Identifier is empty.
17015	Unica Campaign Data Folder Path is empty.
17016	Duplicate Identifiers in Unica Campaign Data Folders.
17017	There is already a stored catalog of the same name.
17018	Catalog name conflicts with an existing catalog on another security policy. Please choose a different name.
17101	Group process internal error.
17102	No input is selected.
17103	No audience is chosen.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17104	No query string.
17105	No filter query string.
17106	No based-on function is chosen.
17107	No based-on field is chosen.
17108	No level is selected.
17109	No count operator is chosen.
17110	Group process cell locking error.
17112	Group process unknown function tag.
17113	Group process report locking error.
17114	The chosen audience is not in the selected table.
17115	An invalid audience level is chosen.
17116	Unknown parameter for Audience Process.
17117	Command not allowed in Reader mode
17201	List box selection error
17202	Dialog initialization error
17203	Tree expansion error

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17204	Combo box insertion error
17205	Invalid cell size limit.
17302	Optimize process internal error.
17303	Optimize process cell locking error.
17304	Optimize process table locking error.
17306	Optimize process unknown function tag error.
17307	Optimize process report locking error.
17308	No input is selected.
17309	No fields are selected for export.
17310	Invalid field name.
17311	String conversion error.
17312	Selected input cells have different audience levels.
17313	Empty or Duplicate Cell Codes.
17314	Proposed contact table is not defined in selected Unica Optimize session.
17315	DB source is not defined in selected Unica Optimize session.
17316	Required field is missing from proposed contact table.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17317	Selected Unica Optimize session is currently running.
17318	Database authentication required.
17319	No Unica Optimize session is selected.
17321	Contact Date is invalid.
17322	Contact Date is expired.
17323	Command not allowed in Reader mode
17324	Selected offer not found.
17325	Channel not found for selected offer.
17326	Cell is missing offer assignment.
17327	Internal Error: Offer is missing.
17328	Internal Error: Channel is missing.
17329	Score field is not specified.
17330	Missing or retired Offer or Offer List detected.
17331	Tried to run flowchart while associated Unica Optimize session was running.
17332	Failed while trying to write to the proposed attribute table.
17333	One or more export field is unmapped.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17334	Tried to delete optimize process box while associated Unica Optimize session was running.
17351	Selection error
17352	Selection not found
17402	CreateSeg process internal error.
17403	CreateSeg process cell locking error.
17404	CreateSeg process unknown function tag.
17405	No input is selected.
17406	Segment name not unique within the specified folder.
17407	Segment name not specified.
17408	Invalid segment name.
17409	Invalid security policy.
17410	Security policy not specified.
17411	Selected input cells have different audience levels.
17412	Bin file creation is OFF and No Temp Table DS is specified.
17413	Invalid Data Source Name for Segment Temp Tables
17452	Segment name not specified

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17502	Internal error.
17503	Cell locking error
17504	Table locking error.
17505	Unknown function tag error.
17507	Report locking error.
17509	No input is selected.
17510	Fulfillment page: No export table is selected.
17511	Personalization page: No export fields are chosen.
17512	Log page: No table is chosen to log contacts.
17513	Log page: No fields are chosen to log contacts.
17514	Getting cell field information error.
17515	No trigger is specified.
17516	Personalization page: no sort field is chosen.
17518	Invalid field name.
17519	Double to string conversion error.
17521	Fulfillment page: No export file is chosen.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17522	Contact list page: No delimiter is specified.
17523	Fulfillment page: No export dictionary file is chosen.
17524	Log page: No file is chosen to log contacts.
17525	Log page: No delimiter is specified for contacts.
17526	Log page: No dictionary file is specified for contacts.
17527	Fulfillment Page: Selected Data Export Filename contains Invalid Path
17528	Fulfillment Page: Selected Data Dictionary for Export File contains an Invalid Path.
17529	Personalization Page: No skip duplicate field is chosen.
17530	Fulfillment Page: Update Records require a base table with audience same as input.
17531	Log Page Contact: Update Records require a base table with audience same as input.
17532	Selected input cells have different audience levels.
17533	Empty or Duplicate Cell Codes.
17534	Log page: No skip duplicate field is chosen.
17535	Fulfillment page: Selected Data Dictionary File does not exist.
17538	Offer codes are not unique.
17539	Command not allowed in Reader mode

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17540	Invalid Offer ID in Unica Deliver Document
17541	Empty audience level.
17542	No offers are selected.
17544	Cell is missing offer assignment.
17549	Error returned from Unica Deliver server during run.
17550	Internal error: unknown Unica Deliver status.
17552	List box selection error
17553	Selection not found
17554	Offer name or code is empty.
17555	The specified records are cleared from Contact History, Detailed Contact History, and Treatment tables.
17557	You are about to permanently delete ALL contact history entries ever created by this process. Are you sure you want to continue?
17558	Invalid expiration duration specified.
17559	Document settings updated from Unica Deliver server.
17560	Duplicate tracking codes not allowed.
17561	Cannot determine tracking audience level.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17562	Invalid number of contacts
17563	Invalid number of responses
17564	Invalid or missing start/end date
17565	Start date is later than end date
17566	You are about to permanently delete the selected contact history entries created by this process. Are you sure you want to continue?
17567	There are no contact history entries created by this process.
17568	Records for this process are cleared from Contact History, Detailed Contact History tables, and Treatment tables.
17570	Missing field assignment for document PF.
17571	Missing field assignment for offer parameter.
17572	Missing field assignment for tracking field.
17573	Unica Deliver directory is invalid.
17574	Missing field assignment for content type.
17575	Unica Deliver is still completing the last operation. Please try again later.
17576	No Unica Deliver document selected.
17577	Unknown parameter.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17578	Invalid parameter.
17579	DOM creation error.
17580	Multiple cells selected. Assignment rule will apply to all selected cells.
17581	Internal Error: Offer is missing.
17582	Internal Error: Channel is missing.
17583	Contact History is tracked at a different audience level. All audience ID fields must be specified.
17584	No output queue selected.
17585	Output queue not found.
17586	Required field not found in output queue.
17587	Log page: Contact History Table is not yet defined for this audience level.
17588	Output page advanced settings: Contact History Table is not yet defined for this audience level.
17589	Output page advanced settings: Response History Table is not yet defined for this audience level.
17590	A new offer parameter name has been added to one of the offer URLs since the process box was configured. You must map a field to this offer parameter before you can begin the run.
17591	The process box needs to be reconfigured due to a Personalization Field change in the Unica Deliver document.
17592	Missing or draft or retired Offer or Offer List detected.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
17593	Assigned Offer List does not contain any offers.
17595	Cannot clear contact history. Response history exists for the selected treatments.
17596	No contact history records found.
17597	Contact history exists for current run. History must be cleared before starting branch or process run.
17599	Specified contact status code is not defined in the system.
17600	Duplicate field names. Output table cannot be created.
17602	Response process internal error.
17603	Response process cell locking error.
17604	Response process table locking error.
17605	Response process unknown function tag error.
17607	Response process report locking error.
17608	Response process getting cell field information error.
17611	Double to string conversion error.
17613	Empty audience level.
17614	No input is selected.
17615	Selected input cells have different audience levels.



**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17616	No offers selected.
17617	One or more offers are missing cell assignments.
17618	Offer Code Field is missing.
17620	Campaign Code Field is missing.
17621	Cell Code Field is missing.
17622	Channel Code Field is missing.
17623	Product ID Field is missing.
17624	No table is chosen to log for other destination.
17625	Update Records require a base table with audience same as tracking.
17626	No file is chosen to log for other destination.
17627	No delimiter is specified for logging to delimited file.
17628	No dictionary file is specified for logging.
17629	No fields are chosen to log for other destination.
17630	Invalid field name.
17631	Offer with selected response type is already added in this process.
17632	Response type(s) not specified.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17633	Response channel is not specified.
17634	Response date field is not a date type field.
17635	Response date value is not in the specified format.
17636	No offer is selected
17637	Internal Error: Offer not found.
17638	Internal Error: Contact channel not found.
17639	Internal Error: Campaign not found.
17640	Offer field must be specified to track all incoming responses.
17641	When tracking at a different audience level from the input cell, all audience ID fields must be specified under 'Additional Fields' in the Log tab.
17642	Default response type not found in the User Response Type Table
17643	Default contact status not found in the Contact Status Table
17644	No treatment mapping specified.
17651	List box selection error
17653	Response name is empty
17654	Records for this Process are cleared from Response History and Tracking tables.
17655	About to clear Response History and Tracking table records for this Process. Are you sure?

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
17656	Response channel is not specified.
17657	Records for this Process are cleared from Contact History and Tracking tables.
17658	About to clear Contact History and Tracking table records for this Process. Are you sure?
17659	Contact History is tracked at a different audience level. All audience ID fields must be specified.
17702	Cube process internal error.
17703	Cube process cell locking error.
17704	Cube process unknown function tag.
17705	No input cell or segment.
17706	Segment name not unique.
17713	No output cube specified.
17714	Dimension no longer exists.
17715	Selected Segment is based on an unknown audience level.
17717	Report locking error.
17718	Invalid field name.
17752	Cube name is missing.
17753	There are no dimensions available.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
17754	There are no dimensions specified for this cube.
17755	Invalid configuration: duplicate dimensions selected.
17800	Error formatting date for display.
17801	Error parsing user input date.
17802	Error formatting currency value for display.
17803	Error parsing user input currency value.
17804	Error formatting number for display.
17805	Error parsing user input number.
17806	Error formatting time for display.
17807	Client Stored List Internal Error.
17808	Error formatting datetime for display.
19000	Internal Error: Unknown function tag.
19001	Memory error
19002	DOM exception
19003	Pipe open error
19005	End date specified is earlier than start date

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
19006	Invalid Report name
19007	Invalid Attribute name
19010	Invalid character(s) were found in a Numeric Field.
19011	Segment is in use. Cannot modify.
19013	Invalid cube specification
19014	Invalid Effective Date
19015	Invalid Expiration Date
19016	Expiration Date specified is earlier than Effective Date
19018	Folder names must be unique within the same folder. The specified folder name already exists in this folder.
19019	Cannot remove folder: folder contents (files/subfolders) must first be removed.
19020	Folder contains in use segment(s). Cannot move.
19021	Cannot delete.
19022	Cannot move.
19023	Folder contains active segment(s). Cannot delete.
19024	Folder contains inactive segment(s). Cannot delete.
19025	No destination folder was selected. Please select a destination folder and retry.

**Table 53. Unica Campaign error codes (continued)**

Code	Error Description
19026	Invalid folder ID specified.
19027	Session name must be unique within the same folder. The specified session name already exists in this folder.
19028	Cannot move Campaign/Session because it contains an active Flowchart.
19029	Cannot move. Move will result in duplicate segment names in destination folder.
19030	An object having the destination name already exists.
19500	Process internal error.
19501	String conversion error.
19502	Selected Unica Optimize session not found.
20000	Internal Error: Unknown function tag.
20002	DOM exception
20003	Pipe open error
20004	Offer code not unique
20005	End date specified is earlier than start date
20006	Invalid Report name
20007	Invalid Attribute name
20008	Offer is used. Cannot delete.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
20009	Folder contains used offer(s). Cannot delete.
20010	Invalid character(s) were found in a Numeric Field.
20011	Segment is in use. Cannot modify.
20012	Offer version name not unique
20013	Invalid cube specification
20014	Invalid Effective Date
20015	Invalid Expiration Date
20016	Expiration Date specified is earlier than Effective Date
20017	Offer version code not unique
20018	Folder names must be unique within the same folder. The specified folder name already exists in this folder.
20019	Cannot remove folder: folder contents (files/subfolders) must first be removed.
20020	Folder contains in use segment(s). Cannot move.
20021	Cannot delete.
20022	Cannot move.
20023	Folder contains active segment(s). Cannot delete.
20024	Folder contains inactive segment(s). Cannot delete.

**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
31607	Kafka configuration missing.
31608	Either of Number of partitions / Number of Replica / Retention Period is not configured.
31609	Number of Replicas Must be less than or equal to Number of Brokers.
32101	No field mapping provided for Link process box.
32102	Link process internal error.
32103	Link process cell locking error.
32104	No input is selected.
32105	Unknown parameter for Link Process.
32106	No actionID specified.
32107	Link No Config.
32108	Connection name not configured.
32109	NO Audience Level defined
32110	Link run API failed
32111	Link Run API invalid status
32112	Selected input cells have different audience levels.
32113	Link run API failed
32114	Link status API failed
32115	Link Run API invalid status
32116	Link delete API failed
32117	No request id
32118	Action not deployed in HIP
32119	Link Info API failed
32120	Link schedule API failed
33100	A listener failover event occurred but the listener was recovered. Your most recent action will be lost. You must repeat the action. If you were editing a flowchart, the last-saved version will be re-loaded in View mode.
33101	No input is selected.



**Table 53. Unica Campaign error codes (continued)**

<b>Code</b>	<b>Error Description</b>
33102	Journey process cell locking error.
33103	Selected input cells have different audience levels.
33104	Unknown parameter for Journey Process.
33105	Session name not selected.
33106	Entry source name not selected.
33107	Entry source code not available.
33108	Entry source id not available.
33109	Data definition name not selected.
33110	Data definition code not available.
33111	Data definition id not available.
33112	Data mapping of export fields require.
33113	Invalid field name.
33114	Journey process internal error.
33122	Data mapping of mandatory export fields require

## Chapter 26. Troubleshooting

**Issue:** Troubleshooting tip for Campaign Flowchart not opening from Schedule management page.

**Resolution:**

Navigate to Affinium|Campaign|navigation

Change the below parameter

flowchartDetailPageURI='flowchartDetails.do?campaignID=&id=' to 'flowchartDetails.go?campaignID=&id='