

IBM Campaign  
Version 9 Release 0  
October 2015

## *Administrator's Guide*



**Note**

Before using this information and the product it supports, read the information in "Notices" on page 391.

This edition applies to version 9, release 0, modification 0 of IBM Campaign and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1998, 2015.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

## Chapter 1. Administration in IBM

### Campaign . . . . . 1

Campaign-related administrative tasks in IBM EMM 2

## Chapter 2. Managing security in IBM

### Campaign . . . . . 3

About security policies . . . . . 3

    The global security policy . . . . . 3

    How Campaign evaluates permissions. . . . . 4

    Using the Owner and Folder Owner roles . . . . . 4

    Guidelines for designing security policies. . . . . 5

Security scenarios. . . . . 5

    Scenario 1: Company with a single division . . . . . 5

    Scenario 2: Company with multiple separate divisions. . . . . 7

    Scenario 3: Restricted access within a division . . . . . 9

Implementing security policies . . . . . 10

    To create a security policy . . . . . 10

    To delete a security policy . . . . . 10

    Assigning security policies to folders or objects 11

About administrative permissions in Campaign . . . . . 11

    Configuring report folder permissions . . . . . 12

    Reference: Administrative permissions in Campaign . . . . . 12

Windows impersonation administration . . . . . 16

    What is Windows impersonation? . . . . . 16

    Why use Windows impersonation? . . . . . 16

    What is the relationship between Campaign users and Windows users? . . . . . 17

    The Windows impersonation group . . . . . 17

    Windows impersonation and logging into IBM EMM . . . . . 17

    Working with Windows impersonation . . . . . 17

### Chapter 3. Database table administration . . . . . 19

Table administration concepts . . . . . 19

    What are system tables? . . . . . 19

    What are user tables? . . . . . 19

    About mapping tables. . . . . 20

    What is a data dictionary? . . . . . 21

    What is a table catalog? . . . . . 21

Initial table administration tasks . . . . . 22

    Prerequisites: expected state following installation 22

    To test system table access . . . . . 22

    To test user table access . . . . . 22

    Mapping Customer audience level system tables 23

Working with system tables . . . . . 24

    To map or re-map a system table . . . . . 24

    To unmap a system table . . . . . 25

    To view system table contents . . . . . 25

Working with user tables . . . . . 26

    About working with user tables . . . . . 26

    Guidelines for mapping user tables . . . . . 27

    To access data sources from within a flowchart 27

    Working with user tables while editing a flowchart . . . . . 27

    Working with user tables from the Campaign Settings page . . . . . 28

Working with data dictionaries . . . . . 37

    To open a data dictionary. . . . . 38

    To apply changes to a data dictionary . . . . . 38

    When to use a data dictionary . . . . . 38

    Data dictionary syntax. . . . . 38

    To manually create a new data dictionary . . . . . 38

Working with table catalogs . . . . . 39

    To access table catalogs . . . . . 39

    To open a table catalog . . . . . 39

    To create a table catalog . . . . . 39

    To load a stored table catalog . . . . . 40

    Deleting table catalogs. . . . . 41

    To update pre-computed profiles for tables in a table catalog . . . . . 41

    To define a data folder for a table catalog . . . . . 42

### Chapter 4. Customizing campaigns .. 43

Custom campaign attributes . . . . . 43

Custom cell attributes . . . . . 43

Custom offer attributes . . . . . 44

    What is a static attribute?. . . . . 44

    What is a hidden static attribute? . . . . . 44

    What is a parameterized attribute? . . . . . 44

Working with custom attributes . . . . . 45

    To create a custom attribute . . . . . 45

    To modify a custom attribute . . . . . 46

Custom initiatives . . . . . 48

    To add an initiative. . . . . 48

Custom products . . . . . 48

    To add a product . . . . . 48

### Chapter 5. Offer template administration . . . . . 49

What is an offer? . . . . . 49

What is an offer template? . . . . . 49

Offer templates and security. . . . . 50

Why use offer templates?. . . . . 50

Planning offer templates and offers . . . . . 50

Working with offer attributes . . . . . 51

    Using custom attributes in offer templates . . . . . 51

    To create a custom attribute . . . . . 51

    To modify a custom attribute . . . . . 53

    Standard offer attributes in Campaign . . . . . 54

Working with offer templates . . . . . 54

    Using drop-down lists in offer templates . . . . . 55

    To create an offer template . . . . . 55

    To modify an offer template . . . . . 56

    To reorder offer templates . . . . . 57

    To retire an offer template . . . . . 58

    Adding list values to the Channel attribute. . . . . 58

Template icons . . . . .	58
Default offer attributes . . . . .	59
Introduction to using Marketing Operations assets in Campaign offers . . . . .	60
Guidelines for using Marketing Operations assets in Campaign offers . . . . .	61
Setting up Marketing Operations assets for use in Campaign offers. . . . .	61

## Chapter 6. Audience level

<b>administration . . . . .</b>	<b>63</b>
About audience levels . . . . .	63
Why different audience levels are needed in Campaign . . . . .	64
The default Customer audience level . . . . .	64
Creating additional audience levels . . . . .	64
About audience levels and system tables . . . . .	64
Default Customer audience level system tables	64
About audience levels and strategic segments ..	65
Audience level unique identifiers . . . . .	65
Required fields in audience-level-specific tables	66
About audience levels and user tables . . . . .	67
User tables with a single audience level . . . . .	67
User tables with multiple audience levels . . . . .	68
Working with audience levels . . . . .	68
Workflow for setting up a new audience level ..	68
Removing an audience level. . . . .	71
About global suppressions and global suppression segments . . . . .	71
Switching audiences with global suppressions ..	72
About creating global suppression segments ..	72
Updating global suppression segments . . . . .	73
Deleting global suppression segments . . . . .	73
Logging for global suppressions . . . . .	73

## Chapter 7. Contact history

<b>administration . . . . .</b>	<b>75</b>
Contact history concepts . . . . .	75
What is contact history? . . . . .	75
What is detailed contact history? . . . . .	76
What is contact status? . . . . .	76
About updating contact status . . . . .	76
How does contact history relate to audience levels? . . . . .	77
How does contact history relate to database and system tables? . . . . .	77
What is offer history? . . . . .	77
What is treatment history? . . . . .	77
Working with contact history . . . . .	78
Creating contact history tables for new audience levels . . . . .	78
Mapping contact history tables to system tables	78
To add a contact status code. . . . .	78
Logging history . . . . .	79
Updating contact history . . . . .	80
Clearing contact history . . . . .	80
Default contact status codes . . . . .	80

## Chapter 8. Response history

<b>administration . . . . .</b>	<b>81</b>
Response history concepts . . . . .	81
What is response history? . . . . .	81
What is the response type? . . . . .	81
How does response history relate to audience levels? . . . . .	81
How does response history relate to database tables? . . . . .	82
Foreign key constraints in response history tables	82
What is an action table? . . . . .	82
What does an action table contain? . . . . .	82
Why use an action table? . . . . .	82
Working with response history . . . . .	83
Creating response history tables for new audience levels . . . . .	83
Mapping response history tables to IBM Campaign system tables . . . . .	83
To set the number of days after an offer expires to record responses. . . . .	83
Adding a response type . . . . .	83
Logging response history. . . . .	84
Response history references . . . . .	84
Default response types . . . . .	84
Sample UA_ActionCustomer Table . . . . .	85

## Chapter 9. Operational monitoring . . . . .

<b>administration . . . . .</b>	<b>87</b>
To configure operational monitoring . . . . .	87
To access the All Monitored Runs page . . . . .	87
Viewing the All Monitored Runs page . . . . .	87
To sort the list of flowcharts on the All Monitored Runs page. . . . .	88
To view associated campaigns or flowcharts ..	88
To refresh the All Monitored Runs page display	88
Working with flowcharts through the All Monitored Runs page. . . . .	88
To stop a running flowchart . . . . .	88
To suspend a running flowchart . . . . .	89
To resume a suspended flowchart . . . . .	89
Operational monitoring references. . . . .	89
Flowchart states and actions. . . . .	89
Properties related to operational monitoring ..	91
All Monitored Runs page icons. . . . .	91

## Chapter 10. Dimension hierarchy

<b>administration . . . . .</b>	<b>93</b>
What is a dimension hierarchy? . . . . .	93
Why use dimension hierarchies? . . . . .	93
About dimension hierarchies and cubes . . . . .	94
About dimension hierarchies and database tables..	94
Working with dimension hierarchies . . . . .	95
About working with dimension hierarchies. ..	95
Designing dimension hierarchies . . . . .	95
Accessing dimension hierarchies in Campaign..	95
To create a dimension hierarchy . . . . .	96
To load a stored dimension hierarchy. . . . .	96
To edit a dimension hierarchy . . . . .	97
To update a dimension hierarchy . . . . .	97
To remove a dimension hierarchy . . . . .	97

## **Chapter 11. Trigger administration . . . 99**

What is an inbound trigger? . . . . .	99
Why use an inbound trigger? . . . . .	99
Inbound triggers and the Schedule process . . . . .	99
What is broadcasting? . . . . .	99
What is an outbound trigger? . . . . .	99
Synchronous outbound triggers . . . . .	100
Asynchronous outbound triggers . . . . .	100
Why use an outbound trigger? . . . . .	100
Return values for outbound triggers . . . . .	101
How are triggers defined? . . . . .	101
Working with trigger folders and triggers . . . . .	101
To create folders to organize triggers . . . . .	101
To move a trigger folder . . . . .	101
To edit a trigger folder . . . . .	102
To delete a trigger folder . . . . .	102
To create a trigger . . . . .	102
To edit or move a trigger . . . . .	103
To delete a trigger . . . . .	104
Setting up outbound triggers . . . . .	104
Setting up a process to execute an outbound trigger . . . . .	104
To set up a flowchart to execute an outbound trigger on success . . . . .	104
To set up a flowchart to execute an outbound trigger on failure . . . . .	104
Setting up inbound triggers . . . . .	105
To set up inbound triggers . . . . .	105
Configuring the Schedule process to run with an inbound trigger . . . . .	105
To broadcast a trigger to all flowcharts of a campaign . . . . .	106
To broadcast a trigger to a flowchart . . . . .	106
To broadcast a trigger to all campaigns . . . . .	106
Setting up the trigger utility on a remote Windows machine . . . . .	106
unica_actrg utility: required files . . . . .	107
Reference information for trigger administration . . . . .	107
Tokens supported by triggers . . . . .	107
Campaign trigger utility syntax . . . . .	109
Campaign trigger utility options . . . . .	109

## **Chapter 12. Logging administration 111**

The Campaign Listener log . . . . .	111
The Campaign web application log . . . . .	111
The Campaign Server Manager log . . . . .	112
The Campaign session utility log . . . . .	112
The cleanup utility log . . . . .	112
Flowchart logs . . . . .	112
The sessions log . . . . .	112
The web connections log . . . . .	112
Windows event logs . . . . .	112
The log4j log file . . . . .	113
Working with logs . . . . .	113
Campaign Listener logging tasks . . . . .	113
Campaign web application logging tasks . . . . .	113
Flowchart logging tasks . . . . .	114
Windows event logging tasks . . . . .	116
log4j logging tasks . . . . .	116

## **Chapter 13. Unique code administration . . . . . 119**

About campaign codes . . . . .	119
About offer codes . . . . .	119
About cell codes . . . . .	119
About treatment codes . . . . .	120
Code formats . . . . .	120
Default code formats . . . . .	121
Code format requirements . . . . .	121
About changing the default code formats . . . . .	121
About code configuration properties . . . . .	123
About code generators . . . . .	123
Default code generators in Campaign . . . . .	123
About custom code generators . . . . .	123
Requirements for custom code generators . . . . .	124
About configuring Campaign to use custom code generators . . . . .	124
About creating custom code generators . . . . .	125
About outputting unique codes . . . . .	125
About outputting errors . . . . .	125
About placing the custom code generators . . . . .	126
To specify the location of the custom offer code generator . . . . .	126
Code generation references . . . . .	126
Properties related to code generation . . . . .	126
Parameters for the default campaign and cell code generators . . . . .	127
Parameters for the default offer code generator . . . . .	127
Parameters for custom code generators . . . . .	128

## **Chapter 14. Advanced settings administration . . . . . 129**

To access Advanced settings . . . . .	129
About the General settings . . . . .	129
Save flowchart run results . . . . .	129
Use in-DB optimization during flowchart run . . . . .	129
Disable global suppression for this flowchart . . . . .	130
Y2K threshold . . . . .	130
Auto-save . . . . .	130
Checkpoint . . . . .	131
Maximum data errors allowed . . . . .	131
Send trigger on flowchart run errors . . . . .	131
Send trigger on flowchart success . . . . .	131
About the Server Optimization settings . . . . .	131
Campaign Virtual Memory Usage . . . . .	132
Disallow use of temporary tables for this flowchart . . . . .	132
About the Test Run Settings . . . . .	132

## **Chapter 15. IBM Campaign utilities 133**

Campaign Listener (unica_aclsnr) . . . . .	133
Campaign listener requirements . . . . .	133
Campaign listener syntax . . . . .	133
Campaign listener options . . . . .	134
Starting and stopping the listener . . . . .	134
Campaign listener log . . . . .	135
Campaign Listener Shutdown utility (svrstop) . . . . .	135
Campaign svrstop utility reference . . . . .	136
To use the svrstop utility to shut down the Campaign listener . . . . .	136

To use the svrstop utility to shut down the Contact Optimization listener . . . . .	137
Campaign Server Manager (unica_svradm) . . .	137
To run the Campaign Server Manager . . .	138
Campaign Server Manager commands . . .	138
To kill a running flowchart . . . . .	143
Campaign session utility (unica_acsesutil) . . .	143
Environment variables required for unica_acsesutil . . . . .	143
Campaign session utility use cases . . . .	144
Campaign session utility return values . . .	147
Campaign session utility syntax . . . . .	147
Campaign session utility options . . . . .	148
Campaign cleanup utility (unica_acclean) . . .	149
Environment variables required for unica_acclean . . . . .	150
Campaign cleanup utility use cases . . . .	150
Campaign cleanup utility syntax . . . . .	153
Campaign cleanup utility options . . . . .	153
Campaign report generation utility (unica_acgenrpt) . . . . .	155
Use case: capture cell counts from flowchart runs . . . . .	155
IBM Campaign report generation utility syntax	156
IBM Campaign report generation utility options	156
Database testing utilities. . . . .	158
cxntest utility . . . . .	158
odbctest utility . . . . .	159
db2test utility . . . . .	160
oratest utility . . . . .	161
Database load utilities . . . . .	161
Fast loader repeated tokens. . . . .	162

## Chapter 16. IBM Digital Analytics and Campaign integration . . . . . 165

Integrating IBM Digital Analytics and Campaign	165
About the translation table . . . . .	168
Mapping the translation table . . . . .	170

## Chapter 17. Configuring Campaign for non-ASCII data. . . . . 173

About using non-ASCII data or non-US locales . .	173
About character encodings . . . . .	173
About interactions with non-ASCII databases	173
About the multi-locale feature. . . . .	174
Configuring Campaign for a non-ASCII language or a non-US locale. . . . .	176
Set the operating system language and regional settings . . . . .	176
Set encoding parameters for the web application server (WebSphere only). . . . .	177
Set Campaign language and locale property values . . . . .	177
Un-map and re-map system tables . . . . .	178
Test your database and server configurations	178
Configuring Campaign for multiple locales . . .	181
Before you begin: Campaign must be installed	181
Configuring for multiple locales on SQL Server	181
Configuring for multiple locales on Oracle . .	182
Configuring for multiple locales on DB2 . . .	183

## Appendix A. Configuration properties on the configuration page . . . . . 187

Marketing Platform configuration properties . . .	187
General   Navigation. . . . .	187
General   Data filtering . . . . .	188
General   Password settings . . . . .	188
General   Miscellaneous. . . . .	190
Platform . . . . .	191
Platform   Scheduler. . . . .	193
Platform   Scheduler   Recurrence definitions	194
Platform   Scheduler   Schedule registrations   Campaign   [Object type] . . . . .	195
Platform   Scheduler   Schedule registrations   Campaign   [Object type]   [Throttling group] .	196
Platform   Security . . . . .	196
Platform   Security   Login method details   Windows integrated login . . . . .	197
Platform   Security   Login method details   LDAP . . . . .	199
Platform   Security   Login method details   Web access control . . . . .	202
Platform   Security   LDAP synchronization	203
Platform   Security   LDAP synchronization   LDAP reference to IBM Marketing Platform group map . . . . .	210
Reporting configuration properties . . . . .	211
Reports   Integrations   Cognos [version]. . .	211
Reports   Schemas   [product]   [schema name]   SQL Configuration . . . . .	214
Reports   Schemas   Campaign . . . . .	215
Reports   Schemas   Campaign   Offer Performance. . . . .	215
Reports   Schemas   Campaign   [schema name]   Columns   [Contact Metric] . . . .	216
Reports   Schemas   Campaign   [schema name]   Columns   [Response Metric] . . . .	218
Reports   Schemas   Campaign   Campaign Performance. . . . .	219
Reports   Schemas   Campaign   Campaign Offer Response Breakout . . . . .	220
Reports   Schemas   Campaign   Campaign Offer Response Breakout   Columns   [Response Type] . . . . .	220
Reports   Schemas   Campaign   Campaign Offer Contact Status Breakout . . . . .	221
Reports   Schemas   Campaign   Campaign Offer Contact Status Breakout   Columns   [Contact Status] . . . . .	222
Reports   Schemas   Campaign   Campaign Custom Attributes   Columns   [Campaign Custom Column] . . . . .	223
Reports   Schemas   Campaign   Campaign Custom Attributes   Columns   [Offer Custom Column] . . . . .	223
Reports   Schemas   Campaign   Campaign Custom Attributes   Columns   [Cell Custom Column] . . . . .	224
Reports   Schemas   Interact . . . . .	225
Reports   Schemas   Interact   Interact Performance. . . . .	226
Reports   Schemas   eMessage . . . . .	226

Campaign configuration properties . . . . .	227
Campaign . . . . .	227
Campaign   collaborate . . . . .	228
Campaign   navigation . . . . .	229
Campaign   caching . . . . .	232
Campaign   partitions . . . . .	233
Campaign   monitoring . . . . .	318
Campaign   ProductReindex . . . . .	320
Campaign   unicaACLlistener . . . . .	321
Campaign   unicaACOOptAdmin . . . . .	326
Campaign   server . . . . .	327
Campaign   logging . . . . .	327

## Appendix B. Special characters in

### Campaign object names. . . . . 329

Special characters not supported . . . . .	329
Objects with no naming restrictions . . . . .	329
Objects with specific naming restrictions . . . . .	330
Naming restrictions for derived fields . . . . .	330

## Appendix C. Supported data types for user tables . . . . . 331

## Appendix D. Internationalization and encodings. . . . . 335

Character encodings in Campaign . . . . .	335
Western Europe . . . . .	335
Unicode encodings . . . . .	336
Arabic. . . . .	336
Armenian . . . . .	336
Baltic Rim . . . . .	336
Celtic . . . . .	336
Central Europe . . . . .	337

Chinese (both Simplified and Traditional) . . . . .	337
Chinese (Simplified) . . . . .	337
Chinese (Traditional) . . . . .	337
Cyrillic . . . . .	337
English . . . . .	337
Georgian . . . . .	338
Greek . . . . .	338
Hebrew . . . . .	338
Icelandic . . . . .	338
Japanese . . . . .	338
Korean . . . . .	338
Lao. . . . .	339
Northern Europe . . . . .	339
Romanian . . . . .	339
Southern Europe . . . . .	339
Thai . . . . .	339
Turkish . . . . .	339
Vietnamese . . . . .	339
Miscellaneous . . . . .	339
Date and time formats . . . . .	340
Formats for DateFormat and DateTimeFormat . . . . .	340
Formats for DateOutputFormatString and DateTimeOutputFormatString . . . . .	342

## Appendix E. Campaign error codes 345

Campaign error codes . . . . .	345
--------------------------------	-----

## Contacting IBM technical support .. 389

## Notices . . . . . 391

Trademarks . . . . .	393
Privacy Policy and Terms of Use Considerations . . . . .	393





---

## Chapter 1. Administration in IBM Campaign

You perform most administrative functions in Campaign by choosing **Settings > Campaign Settings**. Use the Campaign Settings page to perform the tasks described below.

*Table 1. Administrative tasks performed from the Campaign Settings page*

Task	Description
Manage Table Mappings	Manage user table mappings and system table mappings. <ul style="list-style-type: none"><li>• Most system tables are mapped automatically during installation if you use the recommended system table data source name, UA_SYSTEM_TABLES. Important: Only map or remap system tables when no one is using Campaign.</li><li>• User tables must be mapped to an existing database table or file to make data accessible to processes in flowcharts. You can remap a user table at any time. Reasons for remapping user tables are to remove unnecessary fields, make new fields available, rename a table or fields, add an audience level, or change profiling characteristics.</li></ul>
View Datasource Access	View the system table database and all configured customer databases. You can select a database to see its configuration details and you can log in or out of customer databases.
Manage Dimension Hierarchies	Use dimension hierarchies to group data based on value ranges. Examples are time, geography, product, department, and distribution channel. You can create any kind of hierarchy that relates to your business or campaign. Dimension hierarchies are the building blocks of cubes.
Manage Audience Levels	An audience level is an identifiable group that can be targeted by a 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.
View System Log	View the Campaign Listener (aclsnr) log for the current session. The log file opens in a new browser window.
Define Custom Attributes	Define attributes that can be used in campaigns, offers, and cells. For example: <ul style="list-style-type: none"><li>• Define a campaign attribute of Department to indicate which department in your organization is sponsoring the campaign.</li><li>• Define a custom offer attribute of Interest Rate for mortgage offers.</li><li>• Define a custom cell attribute of Audience Type to store the values Cross-sell, Up-sell, Loyalty.</li></ul>
Define Offer Templates	An offer template defines the structure of a particular type of offer. Offer templates are required. Users cannot create an offer without basing it on a template.

Additional administration tasks can be performed in the ways described below.

*Table 2. Additional administrative tasks*

Task	Description
Adjust security	Open the <b>Settings</b> menu and use the <b>User</b> options.
Manage contact history and response history	Modify system tables supplied with Campaign to capture the desired information.
Adjust administrative settings from within in a flowchart	Open a flowchart, click the <b>Admin</b> icon, and use the available options. For example, use <b>Advanced Settings</b> to disable global suppression for a specific flowchart and make other adjustments.
Adjust configuration settings	Choose <b>Settings &gt; Configuration &gt; Campaign</b> .

Table 2. Additional administrative tasks (continued)

Task	Description
Run utilities to perform administrative functions	Campaign supplies a number of command-line utilities to perform server, session, and database tasks.
Integrate Campaign with other IBM® products	Campaign is designed to work together with many other IBM products. For instructions, see the <i>Campaign Administrator's Guide</i> , the <i>Marketing Platform Administrator's Guide</i> , and the documentation supplied with the product you are integrating. The installation guides may also provide important information about integration.
Configure reporting	See the <i>IBM EMM Reports Installation and Configuration Guide</i> supplied with the Marketing Platform.

## Campaign-related administrative tasks in IBM EMM

Administrative tasks for Campaign that involve features or functions implemented across IBM EMM are performed in Marketing Platform. These include the following tasks:

- Managing users, groups, role assignments, security policies, and permissions
- Administering Windows impersonation
- Configuring proxy server authentication
- Managing configuration properties
- Configuring reporting
- Scheduling flowcharts with the IBM Scheduler

For information on performing these tasks, see the *Marketing Platform Administrator's Guide*.

---

## Chapter 2. Managing security in IBM Campaign

Campaign uses the security functions of Marketing Platform to control user access to objects and features in Campaign. Administrators use the Marketing Platform security interface to configure the user accounts, group memberships, roles, and permissions required for users to access Campaign.

User access to the objects and features in Campaign is implemented using security policies.

---

### About security policies

Security policies are the "rule books" that govern security in Campaign; they are consulted each time a user performs an action in the application. Security policies are created per partition (there is no sharing of security policies across partitions). A partition in Campaign can have multiple security policies.

A security policy consists of multiple roles that you define. Each role contains a set of permissions that determine the actions users can perform and the objects that they can access. You can assign users to a role directly, or assign groups to a role (users in those groups would be assigned the role).

When you create an object such as a campaign or offer in the top-level folder, you apply a security policy to the object. In addition, when you create a top-level folder, you apply a security policy to the folder, and any objects or subfolders you create within that folder inherit the security policy that you applied to the folder.

Applying security policies to objects or folders allows you to separate the objects in Campaign for use by different groups of users. For example, you could configure your security policies so that users belonging to one policy cannot access or even view objects that are associated with other policies.

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

### The global security policy

Campaign includes a default global security policy that you can use as is or modify to suit the needs of your organization. If you choose not to create your own security policies, the global security policy is applied by default to the objects that you create in Campaign.

You can use the global policy in addition to your own policies, or use your own policies exclusively. You cannot delete the global policy, even if it is not in use.

Any security policies that you create exist under the global security policy. Under the global policy, you could create a separate security policy for employees of each division in your organization.

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

The pre-defined roles are:

- **Folder Owner** - All permissions enabled
- **Owner** - All permissions enabled
- **Admin** - All permissions enabled. The default user `asm_admin` is assigned this role.
- **Execute** - All permissions enabled
- **Design** - Read and write permissions on most objects. Cannot schedule flowcharts or sessions.
- **Review** - Read-only permissions

The global security policy applies to all users through the Owner and Folder Owner roles, including users who have not been assigned to any other specific role in the global policy. Because the global policy always applies, it can be used, for example, to globally deny permissions to a role.

## How Campaign evaluates permissions

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

1. Identifies all groups and roles to which this user belongs within the global security policy. A user can belong to one, many, or no roles. A user belongs 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. A user belongs 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 has been assigned to a custom-defined policy, if any exist. If so, the system then 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 as follows:
  - a. If any roles have **Denied** permission for this action, then the user is not allowed to perform it.
  - 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.

## Using the Owner and Folder Owner roles

By default, each security policy contains an Owner and a Folder Owner role with all permissions granted. These roles are created by default when you create a security policy. You can modify the permissions of these roles or use the default permissions. You can also modify the permissions for these roles in the global security policy, but you cannot delete them.

The Owner and Folder Owner roles apply to all users; you do not need to assign users to them. The Owner role applies to single objects that a user created. The Folder Owner role applies to all objects in a folder that a user owns.

These roles are useful for restricting users' access to objects that they do not own. For example, you could create a Read-Only role that grants only read permissions on all objects within the security policy. Assign all users to the Read-Only role. As long as no other role explicitly denies permissions (for example, edit or delete),

each user is allowed to edit or delete their own objects (under the Owner role) and objects in their own folders (under the Folder Owner role), but only view objects and folders that are owned by others (under the Read-Only role).

## Guidelines for designing security policies

Follow these guidelines when designing security policies:

- **Keep the design simple.** 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 many of the objects in Campaign, users must be granted both view and modify permissions for the object. This requirement applies to the following objects:
  - campaigns
  - flowcharts
  - offers
  - offer lists
  - offer templates
  - sessions
  - strategic segments

---

## Security scenarios

This section provides security model examples and explains how they are implemented in Campaign using security policies.

- “Scenario 1: Company with a single division”
- “Scenario 2: Company with multiple separate divisions” on page 7
- “Scenario 3: Restricted access within a division” on page 9


### Scenario 1: Company with a single division

All of the 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. You need to create sets of permissions that will determine employees' ability to access, modify, or use these objects, based on their roles within the organization.

#### Solution

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, define roles corresponding to the employee jobs, and for each role, define the appropriate permissions for each object or function.

*Table 3. Object permissions for this scenario*

Functions/Role	Manager	Designer	Reviewer
Campaigns			
• Add Campaigns			
• Edit Campaigns			
• Delete Campaigns			
• Run Campaigns			
• View Campaign Summary			
Offers			
• Add Offers			
• Edit Offers			
• Delete Offers			
• Retire Offers			
• View Offer Summary			

For example, a Manager has full access and editing ability for campaigns and offers. A Reviewer can access campaigns and offers, but cannot add, edit, delete, or run them.

Optionally, you can also create user groups in IBM EMM that match these roles, and then assign user permissions simply by adding users to these groups.

The following table shows a sample subset of the object permissions for this scenario.

*Table 4. Object permissions for this scenario*










Functions/Role	Manager	Designer	Reviewer
Campaigns			
• Add Campaigns			

Table 4. Object permissions for this scenario (continued)

Functions/Role	Manager	Designer	Reviewer
• Edit Campaigns			
• Delete Campaigns			
• Run Campaigns			
• View Campaign Summary			
Offers			
• Add Offers			
• Edit Offers			
• Delete Offers			
• Retire Offers			
• View Offer Summary			

## Scenario 2: Company with multiple separate divisions

Your company has two business divisions, Eastern and Western, that do not share data between them. Within each division, people performing different functions need to access the same objects (campaigns, offers, templates), but with differing permissions to act on these objects, depending on their role.

### Solution

Define two separate security policies, each with the appropriate roles and permissions. The roles in each security policy can be the same or different, depending on the needs of each division. Except for individuals who need to work across both divisions (for example, the controller, cross-divisional managers, or the CEO), assign each user to a role within only one policy. Do not assign any role to the users in the global policy. For users that work across both divisions, assign them a role in the global policy and grant them the desired permissions.

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 following tables show only a sample subset of the possible object permissions in Campaign.

Table 5. Eastern Division Security Policy











Functions/ Role	Folder Owner	Object Owner	Manager	Designer	Reviewer
Campaigns					
• Add Campaigns					
• Edit Campaigns					
• Delete Campaigns					
• View Campaign Summary					
Offers					
• Add Offers					
• Edit Offers					
• Delete Offers					
• View Offer Summary					

Table 6. Western Division Security Policy

Functions/ Role	Folder Owner	Object Owner	Manager	Designer	Reviewer
Campaigns					
• Add Campaigns					
• Edit Campaigns					
• Delete Campaigns					
• View Campaign Summary					
Offers					
• Add Offers					
• Edit Offers					



Table 6. Western Division Security Policy (continued)

Functions/ Role	Folder Owner	Object Owner	Manager	Designer	Reviewer
• Delete Offers					
• Add Campaigns					

### Scenario 3: Restricted access within a division

Employees within a division of your company require read access to the same set of objects (campaigns, offers, templates, and so on), but they are allowed to edit and delete only their own objects and objects in folders that they own.

#### Solution

Define a Read-Only role that grants only read permissions on objects. Assign all users within the division to this role. Keep the default permissions as defined for the Owner and Folder Owner roles.

**Note:** If your company requires only a single security policy, you can use the global policy and assign all users to the Review role.

Each user is allowed to edit or delete their own objects (under the Owner role) and objects in their own folders (under the Folder Owner role), but only view objects and folders owned by others (under the Read-Only role).

The following table shows a sample subset of the object permissions for this scenario.

Table 7. Object permissions for Scenario 3































Functions/Role	Folder Owner	Object Owner	Reviewer
Campaigns			
• Add Campaigns			
• Edit Campaigns			
• Delete Campaigns			
• View Campaign Summary			
Offers			
• Add Offers			
• Edit Offers			

Table 7. Object permissions for Scenario 3 (continued)

Functions/Role	Folder Owner	Object Owner	Reviewer
• Delete Offers			
• View Offer Summary			

## Implementing security policies

This section describes how to create and delete security policies in Campaign and apply security policies to Campaign folders and objects.

**Note:** You must be assigned permission to administer the User Roles & Permissions page in Marketing Platform to work with 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.

### To create a security policy

1. Click **Settings > User Roles & Permissions**. The User Roles & Permissions page displays.
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 & Permissions page. By default, the policy contains a Folder Owner role and an Object Owner role.

### To delete a security policy

Use this procedure to delete any user-created security policies in Campaign that are not in use. You cannot delete the Global Policy.

**Note:** Do not delete any security policies that have been applied to objects in Campaign. If you need to delete a security policy that is in use, first set the security object of every object/folder using that security policy to a different policy (for example, the global policy). Otherwise, those objects 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.

1. Click **Settings > User Roles & Permissions**.  
The User Roles & Permissions page displays.
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**.

A confirmation dialog displays.

6. Click **OK** to delete the policy.

## Assigning security policies to folders or objects

When you create a top-level folder or object in Campaign, you must select a security policy for it. Only policies in which you have been assigned a role are available for you to associate with top-level objects or folders.

By default, all objects in Campaign are associated with the global policy, but you can assign an optional custom-defined policy.

Keep in mind the following rules when associating a folder or object with a security policy:

- **You cannot assign a security policy to objects within folders.** Objects automatically inherit the security policy of the folder in which they reside.
- **The top-level folder determines the security policy.** Objects within a folder, including sub folders, inherit the security policy of the parent folder. In other words, the security policy of the top-level folder determines the security policy of objects and subfolders within it. Therefore, you cannot manually assign a security policy to objects within folders. 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.
- **Security policy changes when 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 or location belonging to a different security policy than its source, the security policy of the lower-level objects or subfolders is automatically changed to the security policy of the new folder or location.

---

## About administrative permissions in Campaign

Administrative permissions in Campaign are assigned for each partition. These administrative functions are different from the object-related functional permissions in security policies, including the global security policy. Users with these permissions can perform the allowed actions on any objects within the partition.

Each partition includes these four pre-defined 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, Run 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.

The procedures for managing administrative roles and permissions in Campaign is the same as the procedures for managing roles and permissions in Marketing Platform.

## 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 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 be enabled after configuration.
- The Cognos server where reports are configured is up and running.
  1. Log in as a Campaign administrator who has the **ReportSystem** role.
  2. Select **Settings > Sync Report Folder Permissions**.

The system retrieves the names the folders 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 & 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.

## Reference: Administrative permissions in Campaign

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:** You can set the permissions for all functions within a category by setting the permissions of the category heading.

### Administration

Table 8. Administration (Administrative permissions)

Permission	Description
Access Monitoring Area	Allows access to the Campaign Monitoring area.

Table 8. Administration (Administrative permissions) (continued)

Permission	Description
Perform Monitoring Tasks	Allows performing of monitoring tasks in the Campaign Monitoring area.
Access Analysis Area	Allows access to reports in the Campaign Analytics area.
Access Optimizations Link	If Contact Optimization is installed, allows access to that application.
Run svradm Command Line Tool	Allows performing of administrative functions using the Campaign Server Manager (unica_svradm).
Run genrpt Command Line Tool	Allows running of the Campaign report generation utility (unica_acgenrpt).
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 Campaign Server Manager (unica_svradm) or the Campaign user interface.
Terminate Server Processes	Allows terminating the Campaign Server (unica_acsvr) using the Campaign Server Manager (unica_svradm).
Terminate Campaign Listener	Allows terminating the Campaign Listener (unica_aclsnr) using the Campaign Server Manager (unica_svradm) or using the svrstop utility.
Run sesutil Command Line Tool	Allows running of the Campaign session utility (unica_acsesutil).
Override Virtual Memory Settings	Allows overriding the Virtual Memory setting in flowchart Advanced Settings.
Access Custom Attributes	Allows access to and managing of custom attribute definitions from the Campaign Settings 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.
Cell Content Report Export	If Cell Content Report Export is granted, allows printing and exporting of the Cell Content report.
Perform Cleanup Operations	Allows performing cleanup operations using unica_acclean or a custom tool.
Change Object/Folder Ownership	Allows changing 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 9. 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 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 10. 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 11. 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 12. 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 13. 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.

## System tables

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

*Table 14. 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 IBM Campaign user tables. User tables contain data about customers and prospects for use in flowcharts.

*Table 15. 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.

Table 15. User tables (Administrative permissions) (continued)

Permission	Description
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	Allows the use of raw SQL in Select process queries, custom macros, and dimension hierarchies.

## 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 16. 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

This section contains the following information:

- “What is Windows impersonation?”
- “Why use Windows impersonation?”
- “What is the relationship between Campaign users and Windows users?” on page 17
- “The Windows impersonation group” on page 17
- “Windows impersonation and logging into IBM EMM” on page 17

## What is Windows impersonation?

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

For example, if Windows impersonation is enabled, when the Campaign user `jsmith` edits a flowchart, a `unica_acsvr` process starts under the Windows user ID associated with the Marketing 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 Campaign users and Windows users?

To use Windows impersonation, you must establish a one-to-one relationship between Campaign users and Windows users. That is, each 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 Campaign. You must create Campaign users in Marketing 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 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 Campaign partition directory for the group.

## Windows impersonation and logging into IBM EMM

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

## Working with Windows impersonation

Setting up Windows impersonation involves the following tasks, described in this section:

- “Set the Windows impersonation property”
- “Create Campaign users”
- “Create the Windows impersonation group” on page 18
- “Assign the Windows impersonation group to policies” on page 18
- “Assign rights to the Windows impersonation group” on page 18

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

### Set the Windows impersonation property

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 *Marketing Platform Administrator's Guide*.

### Create Campaign users

You can use Marketing Platform to create 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.

## Create the Windows impersonation group

**Note:** You must have administration privileges on the Windows server to complete this task.

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

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

## Assign the Windows impersonation group to policies

**Note:** You must have administration privileges on the Windows server to complete this task.

After you create a Windows group to store users that correspond to 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.

## Assign rights to the Windows impersonation group

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

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

---

## Chapter 3. Database table administration

As a Campaign administrator, you must:

- Map Campaign system tables, if this is not done during Campaign installation.
- Map user tables so that customer data can be used by Campaign flowcharts.

In addition, as an administrator, you work with:

- Data dictionaries, which define the structure of user tables based on fixed-width flat files
- Table catalogs, which enable the efficient management of mapped user tables

---

### Table administration concepts

This section describes the following table administration concepts:

- “What are system tables?”
- “What are user tables?”
- “About mapping tables” on page 20
- “What is a data dictionary?” on page 21
- “What is a table catalog?” on page 21

#### What are system tables?

System tables are database tables containing Campaign application data. Specifically, system tables store meta data on campaign objects, such as campaigns, sessions, flowcharts, offers, templates, custom macros, stored derived fields, triggers. Contact and response history information is also stored in system tables.

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

#### What are user tables?

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

**Note:** Before mapping a user table in Campaign, ensure that the table uses only data types that Campaign supports. For a list of supported data types for each database, see Appendix C, “Supported data types for user tables,” on page 331.

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.

You work with three types of user tables, described below:

- Base tables
- Dimension tables
- 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 Campaign. It is the simplest table type that can be created and is used only to export data from Campaign for use by other applications (general tables cannot be subsequently accessed in 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 Campaign.

## About mapping tables

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

A table mapping is the metadata 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.

## What is a data dictionary?

A data dictionary is a file that defines the format of data in a fixed-width ASCII flat file used in 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 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-IBM 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.

## What is a table catalog?

A table catalog is a collection of mapped user tables.

Table catalogs store all the user table mapping meta data information for re-use across flowcharts. Table catalogs are stored by default in a proprietary binary format using a .cat extension. For more details, see "Stored table catalogs overview" in the *Campaign User's Guide*.

You can also store (and subsequently load) a table catalog in XML format by providing a table catalog name ending with a .XML extension. 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.

You use table catalogs to:

- Easily save, load, and update your 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 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.

You should delete table catalogs only using the Campaign interface. If you remove tables or otherwise change table catalogs directly in the file system, Campaign cannot guarantee data integrity in your system.

---

## Initial table administration tasks

This section describes the following initial table administration tasks to be performed after installation of Campaign:

- “Prerequisites: expected state following installation”
- “To test system table access”
- “To test user table access”
- “Mapping Customer audience level system tables” on page 23

### Prerequisites: expected state following installation

The initial administration tasks described in this section assume that the Campaign installation is complete, including:

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

For more information about these tasks, see the installation documentation.

In addition, you should define necessary audience levels before beginning to work with user tables, as you need to specify audience levels related to your base tables.

For instructions on confirming that the Campaign system is ready for you to begin the table administration tasks, see:

- “To test system table access”
- “To test user table access”
- “Mapping Customer audience level system tables” on page 23

### To test system table access

You should confirm that the Campaign system tables are mapped and that database connections are working properly.

1. Select **Settings > Campaign Settings**. The Campaign Settings page opens, presenting links for various administration tasks.
2. In the **Data Source Operations** section, click **Manage Table Mappings**.

The Table Mappings window opens, with **Show System Tables** selected.

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

Each IBM Campaign system table entry should have a populated database table name in the right column, though it is possible that your implementation does not use particular features, which may leave some system tables unmapped.

If you cannot confirm that the system tables are mapped, contact the person who performed the Campaign installation and configuration.

### To test user table access

You should confirm that Campaign is configured properly to access the necessary user tables.

1. Select **Settings > Campaign Settings**. The Campaign Settings page opens, presenting links for various administration tasks.
2. In the **Data Source Operations** section, click **Manage Table Mappings**.  
The Table Mappings window opens, with **Show System Tables** selected.
3. Select **Show User Tables**. Initially, there are no mapped user tables and the list is empty.
4. Click **New Table**. The New Table Definition window opens.
5. Click **Next**.  
You are prompted to specify whether to map to a file or database.  
When **Map to Existing Table in Selected Database** is selected, you should see one or more databases in the **Select Data Source** list. If no entries appear in the Select Data Source box, you must define the data sources. For more information, see the installation documentation.
6. If Campaign is using one or more flat files for user data:
  - a. Select **Map to Existing File**, then click **Next**. The New Table Definition window now contains fields for the flat file and data dictionary location.
  - b. Click **Browse** to locate the necessary files, or enter the relative path and file name directly into the text box. Files must be located under the partition root of Campaign to be accessible.
 You can now map the user data into Campaign, as described in “Working with user tables” on page 26.

You can also view the customer databases that Campaign is set up to access when editing a flowchart. Click **Settings > Campaign Settings** and select **View Datasource Access**. The **Database Sources** window opens, listing the system table database, and all configured customer databases. From this window, you can log into and log out of customer databases.

## Mapping Customer audience level system tables

Campaign is delivered with the Customer audience level. The system database tables to support the Customer audience level are created when you run the provided system table creation script, as described in the installation documentation.

However, the Customer audience level tables are not mapped following installation. You must map these tables as follows:

*Table 17. Mappings for Customer audience level tables*

IBM Campaign System Table	Database Table to Map to
Customer Contact History	UA_ContactHistory
Customer Detailed Contact History	UA_DtlContactHist
Customer Response History	UA_ResponseHistory
Customer Segment Membership	UA_SegMembership

### About mapping segment membership tables

The segment membership table is one of the audience level system tables that Campaign creates when you define a new audience. If you use strategic segments in Campaign flowcharts or Optimize sessions within Contact Optimization, 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 Contact Optimization that use strategic segments may provide inaccurate results.

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

Using strategic segments in Campaign flowcharts and Contact Optimization 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 Campaign or Contact Optimization 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.

---

## **Working with system tables**

This section contains the following information:

- “To map or re-map a system table”
- “To unmap a system table” on page 25
- “To view system table contents” on page 25

### **To map or re-map a system table**

Most system tables are mapped automatically during initial installation and configuration if you use the recommended system table data source name, UA\_SYSTEM\_TABLES. For more information, see the installation documentation. If you need to map system tables, continue with these instructions.

**Important:** You should only map or remap system tables when no users are using Campaign.

1. Select **Settings > Campaign Settings**. The Campaign Settings page opens, presenting links for various administration tasks.
2. In the **Data Source Operations** section, click **Manage Table Mappings**. The Table Mappings window opens.
3. Select **Show System Tables**.
4. Select the table to map from the **IBM Campaign System Table** list and double-click it or click **Map Table** or **Re-Map Table**.

The Select source database and match required fields window opens.

5. If the table is not automatically selected in the **Source Table** drop-down list, select it (entries are alphabetically listed by owner.table names). Source table fields in the 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 should be matched automatically.



**Note:** When mapping a system table, do not select a different table from the Source Table list; you will not be able to complete the mapping. If you do this accidentally, click Cancel and select a the correct table in the Table Mappings window.

6. Click **Finish**.

## To unmap a system table

**Important:** Unmapping a system table without re-mapping it can cause significant application problems. Unmapping a system table can cause features and existing campaigns to stop working.

**Important:** You should only unmap system tables when no users are using Campaign.

1. Select **Settings > Campaign Settings**. The Campaign Settings page opens, presenting links for various administration tasks.
2. In the **Data Source Operations** section, click **Manage Table Mappings**. The Table Mappings window opens.
3. Select **Show System Tables**.
4. Select the table to unmap in the **IBM Campaign System Table** list, then click **Unmap Table**. You are prompted to confirm the unmapping.

You should immediately remap the system table or be sure that it does not need to be mapped in your environment.

## To unmap the segment membership table

When you unmap the segment membership table you must also clear existing cache files and restart the Campaign and Contact Optimization listeners.

**Note:** If you are using Contact Optimization, 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 Campaign, unmap the segment membership table for the audience.
2. Delete `unica_tbmgr.cache` from the conf directory of your 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 Contact Optimization installation.  
By default, this file is located at `Optimize\partitions\<partition[n]>\conf`.
4. Restart the Campaign listener (`unica_ac1snr`).
5. Restart the Contact Optimization listener (`unica_aol1snr`).

## To view system table contents

You can view the contents of mapped system tables.

For convenience, you can view the contents of most system tables from within Campaign's table manager. You cannot edit the system table data.

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

1. Select **Settings > Campaign Settings**. The Campaign Settings page opens, presenting links for various administration tasks.
2. In the **Data Source Operations** section, click **Manage Table Mappings**. The Table Mappings window opens.
3. Select **Show System Tables**.
4. Select the system table whose contents you want to view.
5. Click **Browse**.  
A window opens showing the table data.  
You can sort by any column in ascending order by clicking on it. Clicking on the column again will toggle the sort order.

To close the window, click the **X** in the upper-right corner.

---

## Working with user tables

This section contains the following information:

- “About working with user tables”
- “To access data sources from within a flowchart” on page 27
- “Working with user tables while editing a flowchart” on page 27
- “Working with user tables from the Campaign Settings page” on page 28
- “To map a base record table to an existing database table” on page 28
- “Mapping a base record table to an existing file” on page 30
- “To map a dimension table” on page 32
- “To map a general table to a database table” on page 33
- “To map a general table to a file” on page 34
- “Re-mapping user tables” on page 34
- “Profiling values and counts” on page 35
- “To unmap a user table” on page 36
- “To create a new user table through an output process” on page 36

**Note:** You should define necessary audience levels before beginning to work with user tables, as you need to specify audience levels related to your base tables.

### About working with user tables

Typically, most of the marketing data you will access from a flowchart will reside in a database, but it may be convenient to access data directly from a flat file.

Campaign supports the ability to work with data stored in either 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 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).

## Guidelines for mapping user tables

Follow these guidelines for 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 Campaign objects, see Appendix B, “Special characters in Campaign object names,” on page 329.
- Do not use function names or keywords from the IBM 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 *IBM Macros for IBM EMM 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.

## To access data sources from within a flowchart

To access the customer or prospect database tables from within a flowchart, you must ensure that you are logged in to the referenced databases.

1. While editing a flowchart, click the **Admin** icon and select **Database Sources**.  
The Database Sources window opens. The database containing system tables is listed, as well as all databases for which Campaign is configured to access.
2. To log into 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, you must map that table, as described in the next section.

## Working with user tables while editing a flowchart

You can work with user tables while editing a flowchart by:

- Using the **Admin** menu
- Mapping a new user table through the Select process
- Exporting data to a base or general table through Snapshot, Call List, and Mail List processes

### Working with user tables by using the Admin menu

Click the **Admin** icon and select **Tables**. The Table Mappings window opens, showing a list of mapped user tables.

The following are the tasks you can then perform:

- “To map a base record table to an existing database table” on page 28
- “Mapping a base record table to an existing file” on page 30
- “To map a dimension table” on page 32
- “To map a general table to a database table” on page 33
- “To map a general table to a file” on page 34
- “Re-mapping user tables” on page 34
- “To unmap a user table” on page 36

### Working with user tables from the Select process

In the **Source** tab of a Select process, in the **Input** drop-down list, select **New Table**. The New Table definition window opens.

The following are the tasks you can then perform:

- “To map a base record table to an existing database table”
- “Mapping a base record table to an existing file” on page 30

## Working with user tables from exported data

You can create new user tables from output processes.

## Working with user tables from the Campaign Settings page

Select **Settings > Campaign Settings**. The Campaign Settings page opens.

Then click **Manage Table Mappings**. The Table Mappings window opens.

You can then perform the following tasks:

- “To map a base record table to an existing database table”
- “Mapping a base record table to an existing file” on page 30
- “To map a dimension table” on page 32
- “To map a general table to a database table” on page 33
- “To map a general table to a file” on page 34
- “Re-mapping user tables” on page 34
- “To unmap a user table” on page 36

**Note:** Before mapping a user table in Campaign, ensure that the table uses only data types that Campaign supports. For a list of supported data types for each database, see Appendix C, “Supported data types for user tables,” on page 331.

## To map a base record table to an existing database table

You can map a new base record table to an existing database table:

- When editing a flowchart, either through the **Admin** menu, or a Select process, as described in “Working with user tables while editing a flowchart” on page 27.
- Through the Campaign Settings page, as described in “Working with user tables from the Campaign Settings page.”

You map a new base record table to make data accessible to processes in flowcharts.

1. Begin from the **New Table Definition - Select the table type** window.

**Note:** When you access the table mapping wizard from a Select process, the Dimension Table and General Table options are not listed.

2. If necessary, select **Base Record Table**, then click **Next**.
3. Select **Map to Existing Table in Selected Database**, select the data source name, then click **Next**.
4. Select the table you want to map from the **Source Table** list.

The tables are listed in alphabetical order by `<owner>.<table_name>`. If you do not see a table you expect, check whether the data source is configured to filter specific table entries.

The source fields in the table you select are mapped automatically to new table fields in the base record 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**, an **Down 1** buttons, until the table is mapped as required.

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.

5. Click **Next**.
6. Optionally, you can change the name which Campaign will use for the base record table and its fields to a more user-friendly value.
  - a. To change the table name, edit the name in the **IBM Campaign Table Name** field.
  - b. To change a field name, select the field name in the **New Table Fields** list, and edit the text in the **IBM Campaign Field Name** field.
7. Click **Next**.
8. Select an Audience Level from the drop-down list. The **Audience Fields** list is automatically populated with the fields required to define the selected audience level. You must match one or more fields in the new base table that correspond to each required key.
9. If the selected audience level is normalized (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.
10. Click **Next**.
11. Optionally, in the Specify Additional audience levels screen, 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**. The Audience Level and ID Fields window opens.
  - 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 the selected audience level is normalized (that is, 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 f for each audience level you want to add for the base table, then click **Next**.
12. If dimension tables exist in the current table catalog, the Specify Relationship to Dimension Tables window 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**.
13. 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.
14. Click **Finish**.

You have created the 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 file

You can map a new base record table to an existing file on the Campaign server within your partition (that is, the file must be located under your partition root):

- When editing a flowchart, either through the **Admin** menu, or a Select process, as described in “Working with user tables while editing a flowchart” on page 27.
- Through the Campaign Settings page, as described in “Working with user tables from the Campaign Settings page” on page 28.

You map a new base record table to make data accessible to processes in flowcharts.

#### To map a base record table to a file:

1. Begin from the **New Table Definition - Select the table type** window.
2. If necessary, select **Base Record Table**, then click **Next**.
3. Select **Map to Existing File**, then click **Next**.
4. Specify the type of file and mapping settings.

#### To map a base record table to an existing fixed-width flat file:

You can map a new base record table to an existing file on the Campaign server within your partition (that is, the file must be located under your partition root):

- When editing a flowchart, either through the **Admin** menu, or a Select process.
- Through the Campaign Settings page.

You map a new base record table to make data accessible to processes in flowcharts.

1. Begin from the **New Table Definition - Select the table type** window.
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**.  
The Settings section of the window appears.
5. In the Settings section of the window, click **Browse** to select the **Source File** from within your campaign partition root directory. 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.

#### To map a base record table to an existing delimited file:

You can map a new base record table to an existing file on the Campaign server within your partition (that is, the file must be located under your partition root) when editing a flowchart or through the Campaign Settings page.

You map a new base record table to make data accessible to processes in flowcharts.

1. Begin from the **New Table Definition - Select the table type** window.
2. If necessary, select **Base Record Table**, then click **Next**.
3. Select **Map to Existing File**, then click **Next**.
4. Select **Delimited File** for the **File Type**.

5. In the Settings section of the window, check **First Row of Data Contains Field Names** if this is the case. This automatically uses the first row of data to define the fields of the base table. You can override these values later.
6. Select the **Field Delimiter** (this is the character used to separate each field in a row of data): **TAB**, **COMMA**, or **SPACE**.
7. Select the **Qualifier** used to indicate how strings are delimited in the file: **None**, **Single Quote**, or **Double Quote**.

This is important if you have a space-delimited file with spaces as part of a field entry. In this case, the field must be enclosed in quotation marks so the embedded spaces are not interpreted as field delimiters. For example, if you had a row of data such as: "John Smith" "100 Main St.", with the delimiter as SPACE and Qualifier set to NONE, Campaign would parse this as five different fields ("John" as the value of the first field, "Smith" as the second, "100" as the third, and so forth). If the qualifier is set to **Double Quote**, then this record is correctly parsed as two fields (name and street address).

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

8. Click **Browse** to select the **Source File** from within your partition directory.
9. Specify the fields to use in the new table. By default, all fields in the file are listed.

If you are mapping a delimited file, the first 50 lines of the file will be sampled for field type and width. You can override the field type (Numeric or Text) and the width, which are automatically detected. For example, if the IDs in the first 50 lines are detected to be 2 characters wide, but you know that IDs later in the file consist of up to 5 characters, increase the value to 5.

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

You 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. Click on the **Field Name** column in the **New Table Fields** section to automatically sort column names alphabetically in ascending (or descending) order.

10. When finished, click **Next**. The Specify table name and field information screen opens.
11. Accept the defaults, or edit the **IBM Campaign Table Name** field to change the name of the table as it will appear in Campaign. You can also change the **IBM Campaign Field Name** mapped to each source field name by selecting the field name and editing the text in the **IBM Campaign Field Name** text box in the **Edit Selected Field Information** section.
12. When your changes are complete, click **Next**. The Specify the selected table's audience level and assign ID field(s) to it screen opens.
13. Select an **Audience Level** from the drop-down 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, in the Specify Additional audience levels screen, 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**. The Audience Level and ID Fields window opens.

- 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 the selected audience level is normalized (that is, 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 f for each audience level 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.

### To map a dimension table

You can map a new dimension table based on an additional table:

- When editing a flowchart, through the Admin menu, as described in “Working with user tables while editing a flowchart” on page 27.
- Through the Campaign Settings page, as described in “Working with user tables from the Campaign Settings page” on page 28.

You 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.

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

1. Begin from the **New Table Definition - Select the table type** window.

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

2. Select **Dimension Table**, then click **Next**.
3. 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 base dimension table 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 Campaign will use for the dimension table and its fields.
  - a. To change the table name, edit the name in the **IBM 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 **IBM 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**.

**Note:** 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.

### To map a general table to a database table

You can map a new general table to an existing database table:

- When editing a flowchart through the **Admin** menu, as described in “Working with user tables while editing a flowchart” on page 27.
- Through the Campaign Settings page, as described in “Working with user tables from the Campaign Settings page” on page 28.

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

1. Begin from the **New Table Definition - Select the table type** window.
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 Campaign will use for the general table and its fields.

To change the table name, edit the name in the **IBM Campaign Table Name** field.

To change a field name, select the mapping in the **New Table Fields** list, and edit the text in the **IBM Campaign Field Name** field.

6. Click **Finish**

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

## To map a general table to a file

You can map a new base record table to a file:

- When editing a flowchart, through the Admin menu, as described in “Working with user tables while editing a flowchart” on page 27.
- Through the Campaign Settings page, as described in “Working with user tables from the Campaign Settings page” on page 28.

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

1. Begin from the **New Table Definition - Select the table type** window.
2. Select **General Table**, then click **Next**.
3. Select **Map to Existing File**, then click **Next**.
4. To map a base record table to a fixed-width flat file:
  - a. Keep the default **File Type** selection.
  - b. Click **Browse** to select the **Source File**. Campaign will automatically populate 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 a base record 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 the **Field Delimiter** used: **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's Fields window opens.
6. Specify the fields to use in the new table. By default, all fields in the file are listed.

If you are mapping a delimited file, the first 50 lines of the file will be sampled for field type and width. You can override the field type (Numeric or Text) and the width, which are automatically detected. For example, if the IDs in the first 50 lines are detected to be 2 characters wide, but you know that IDs later in the file 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.

You 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.

7. When finished, click **Next**.

The **Specify table name and field information** window opens.
8. Accept the defaults, or edit the **IBM Campaign Table Name** field to change the name of the table as it will appear in Campaign; and change the IBM Campaign field names that are mapped to the source field names.
9. Click **Finish**. You have created the general table based on a file.

## Re-mapping user tables

You can remap a user table at any time. You may want to do this 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.
- Or change profiling characteristics.

If you remove a field that has been 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.

Remember that 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.

You can re-map a user table:

- When editing a flowchart, through the Admin menu, as described in “Working with user tables while editing a flowchart” on page 27.
- Through the Campaign Settings page, as described in “Working with user tables from the Campaign Settings page” on page 28.

#### **To re-map a user table:**

1. Begin from the **New Table Definition - Select the table type** window.
2. Select the table to re-map.
3. Click **Re-Map Table**.
4. You then repeat the steps involved in mapping the table:
  - “To map a base record table to an existing database table” on page 28
  - “Mapping a base record table to an existing file” on page 30
  - “To map a dimension table” on page 32
  - “To map a general table to a database table” on page 33
  - “To map a general table to a file” on page 34

### **Profiling values and counts**

When mapping a user table, 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. Profiling enables users to see a table's values when editing a flowchart, without having to view raw data and allows them to easily select from valid values when building a query. Pre-computed profiles provide quick and efficient 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.

**Note:** You can choose to both pre-compute profiles and allow users to dynamically profile in real-time, or you can choose to disable real-time profiling and force users to always use the pre-computed profiles. Enabling or disabling the Allow real-time profiling option applies to all table fields, not just the checked ones. If you disable 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 disable 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.

### To configure profiling:

1. Check the fields for which you want Campaign to pre-compute distinct values and frequency counts.

By default, 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 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 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.

### To unmap a user table

You can unmap a user table at any time (unmapping a table does not delete the original underlying data or affect any other flowchart).

**Important:** Unmapping a user table causes any process in the current flowchart referencing that user table to become unconfigured.

You can unmap a user table:

- When editing a flowchart, through the Admin menu, as described in “Working with user tables while editing a flowchart” on page 27.
  - Through the Campaign Settings page, as described in “Working with user tables from the Campaign Settings page” on page 28.
1. Begin from the **New Table Definition - Select the table type** window.
  2. Select the table to unmap.
  3. Click **Unmap Table**. You are prompted to confirm.
  4. Click **OK** to unmap the table.

**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.

### To create a new user table through an output process

From the output processes (Snapshot, Call List, and Mail List) in the **Export** drop-down list, you can create new user tables.

1. While editing a flowchart, open the output process from which you want to create the new user table.
2. In the **Export to** drop-down list, select **New Mapped Table**. The New Table Definition window opens.
3. Select **Base Record Table**, **Dimension Table**, or **General Table**. Typically in this process, you would export the data to a new base table in an existing flat file or database. If you need to read the exported data back into Campaign, you must export it as a base table.
4. Click **Next**.
5. Select **Create New File** or **Create New Table in Selected Database**.

6. If you select to create a new database table:
  - a. Select the database in which to create the table.
  - b. Click **Next**.
  - c. Select the **Source Table Fields** to export to the new table or file. You can select from Campaign Generated Fields, the audience-level identifier, and fields from the input cell. Use the **Add**, **Remove**, **Up**, and **Down** buttons to define fields in the **New Table Fields** list.
  - d. Click **Next**.
  - e. Specify a **Database Table Name** and **IBM Campaign Table Name** for the new table.
  - f. Optionally, select the new table fields and modify the **IBM Campaign Field Name**.
  - g. Click **Next**.
  - h. Select the **Audience Level** for the new table and specify the audience level field in the new table.
  - i. Click **Next**.
  - j. Optionally, select additional audience levels for the new table by clicking **Add**.
  - k. Click **Next**.
  - l. Define profiling for the new table. For more information, see “Profiling values and counts” on page 35.
  - m. Click **Finish**.
7. If you select to create a new file:
  - a. Click **Next**.
  - b. Select **Fixed-Width Flat File** or **Delimited File**, then specify the **Settings** fields appropriately.
  - c. Click **Next**.
  - d. Select the **Source Table Fields** to export to the new table or file. You can select from Campaign Generated Fields, the audience-level identifier, and fields from the input cell. Use the **Add**, **Remove**, **Up**, and **Down** buttons to define fields in the **New Table Fields** list.
  - e. Click **Next**.
  - f. Select the **Audience Level** for the new table and specify the audience level field in the new table.
  - g. Click **Next**.
  - h. Optionally, select additional audience levels for the new table by clicking **Add**.
  - i. Click **Next**.
  - j. Define profiling for the new table. For more information, see “Profiling values and counts” on page 35.
  - k. Click **Finish**.

---

## Working with data dictionaries

You can edit the data dictionary for an existing or newly created 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 Campaign server or be accessible from the server to be used for table mapping.

## To open a data dictionary

Find the data dictionary you want, then open it using Notepad or any other text editor.

The file you see will look similar to the following example:

```
CellID, ASCII string, 32, 0, Unknown,  
MBRSH, 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,
```

You can 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.

## To apply changes to a data dictionary

You must save, close, and reopen the flowchart.

## When to use a data dictionary

You use data dictionaries in Snapshot processes to ensure that any fixed-width output files you create adhere to a specific structure.

## Data dictionary syntax

Each line in a data dictionary defines a field in fixed-width flat file, using the following syntax:

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

**Note:** 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.

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.

**Note:** The format and comment fields are not used by Campaign. Therefore, for the best results, use "Unknown" for the format value and to leave the Comment field blank.

## To manually create a new data dictionary

This section describes how to create a new data dictionary manually. It may be easier to create a new data dictionary by starting with an existing data dictionary created by Campaign.

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 in the following:  
acct\_id, ASCII string, 15, 0, Unknown,  
hsehd\_id, ASCII Numeric, 16, 0, Unknown,  
occpn\_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.

---

## Working with table catalogs

This section contains the following information:

- “To access table catalogs”
- “To open a table catalog”
- “To create a table catalog”
- “To load a stored table catalog” on page 40
- “Deleting table catalogs” on page 41
- “To update pre-computed profiles for tables in a table catalog” on page 41
- “To define a data folder for a table catalog” on page 42

### To access table catalogs

1. Select **Settings > Campaign Settings**.  
The Campaign Settings page appears.
2. Click **Manage Table Mappings**.  
The Table Mappings window appears.
3. In the Table Mappings window, select **Show User Tables**.

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

### To open a table catalog

1. Follow the instructions in “To access table catalogs.”
2. Click **Load**. The previously stored catalogs are listed in the **Items List**.
3. Select the catalog to open and click **Load Catalog**.

### To create a table catalog

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 table mappings or restore table mappings.

1. Follow the instructions in “To access table catalogs.” Ensure that the user tables you want to save as a table catalog are mapped in Campaign.
2. In the Table Mappings window, select the user tables you want to save to the table catalog, and click **Save**.

3. In the Save Tables window, select the option to save all table mappings to the table catalog, or the option to save only the selected table mappings to the table catalog, then click **OK**.

The Save Table Mappings to Catalog File window opens.

4. Enter details for the new table catalog, including the folder under which you want to save it, a name, security policy, notes, and whether you want to save this catalog with database authentication information.
5. Enter a name for the table catalog. If you enter .XML as the extension name, the table catalog will be stored in XML format rather than as a proprietary binary file.

**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 (\_).

6. (Optional) Add a description for the table catalog in the **Note** field.
7. (Optional) Check **Save with Database Authentication Information**.
  - If you leave **Save with Database Authentication Information** unchecked, anyone using this 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 they do 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 current authentication information (the login[s] and password[s] you are currently using to access these data source[s]) are saved with the table catalog and anyone with access permissions to this table catalog will automatically be connected to the data sources using the authentication stored in the table catalog. This means that users of this table catalog will not need to provide a login or password to access these data sources and will have all the privileges of the stored login for reading and writing to the data source. You may want to avoid this setting for security reasons.
8. Select a folder in which to save the table catalog using the **Save Under** drop-down list.

You can organize table catalogs in folders. Select an existing folder from the **Items List**, or click the **New Folder** button to create a new folder.

If you do not select a specific folder, or choose **None** in the **Save Under** drop-down list, the current table catalog will be saved at the top level. The selected folder appears after the **Save Under** field.

9. Click **Save**.

The table catalog is saved as a .cat file if no extension was provided with the name and in the location you have selected. If you specified a file name with a .xml extension, the table catalog is saved in XML format.

## To load a stored table catalog

You can load a table catalog that you previously saved for use in the current flowchart.

**Note:** If you choose the Load table mappings from stored table catalog (existing mappings are cleared) option, the mapped tables in that flowchart are lost; that is, they are replaced with the table mappings from the loaded catalog. If you choose



Merge table mappings from stored table catalog (old mappings are overwritten), old table mappings not in the new table catalog to be loaded are preserved.

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

1. Follow the instructions in “To access table catalogs” on page 39.
2. Click **Load**.  
The Load Tables window opens.
3. Choose the appropriate option to select whether, when the tables are loaded, you want to clear or to overwrite the existing mappings. By default, the option to clear existing mappings is selected.
4. Click **OK**.  
The **Stored Table Catalogs** window opens.
5. Select the name of the table catalog that you want to load.  
When you click on the name of a table catalog, its information appears in the **Info** box and the Load Catalog button is enabled.
6. Click **Load Catalog**.  
The selected catalog is loaded. You see the details of the table(s) in the new catalog displayed in the Table Mappings window.

## Deleting table catalogs

You delete table catalogs from a flowchart page in **Edit** mode. This task is not available from the **Manage Table Mappings** link on the Campaign Settings page.

**Important:** You should remove table catalogs only using the Campaign interface. If you remove tables or otherwise change table catalogs directly in the file system, Campaign cannot guarantee data integrity in your system.

For details about removing stored table catalogs, see the *Campaign User's Guide*.

## To update pre-computed profiles for tables in a table catalog

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

1. Follow the instructions in “To access table catalogs” on page 39.
2. 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 compute record counts and values for all user tables, you do not need to select any tables.

3. Click **Compute**.

The **Recompute** window opens.

If you have not selected one or more user tables, by default, the Recompute Record Counts and List of Distinct Values for All Tables option is selected.

If you have selected a subset of tables, the Recompute Record Counts and List of Distinct Values for Selected Tables option is selected. The option to compute All Tables is available.

**Note:** If you have not selected any tables for computing and want to enable the option to compute values for selected tables, click **Cancel** on the Recompute window. The window closes and you are returned to the Table Mappings window, where you can select the tables for which you want the record counts and values to be computed.

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

When the computations are complete, you return to the **Table Mappings** window.

## To define a data folder for a table catalog

When you create a table catalog, you can also specify one or more data folders associated 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, click the **Admin** icon and select **Tables**.
2. In the Table Mappings window, select the mapped user tables that you want to save to a catalog. Click **Save**.
3. In the Save Table Mappings to Catalog File window, click in the **IBM Campaign Data Folders** section to add an item.
4. Enter a name and folder location for the data folder you are adding, relative to the current partition's home directory. For example, if you are working in partition1, the folder location you specify is relative to the partitions/partition1 folder.
5. Click **Save**.

After you have saved the table catalog with its associated data folder(s), when you reload the catalog in a flowchart containing output processes such as Snapshot, these folders appear as options in the file location selection dialog.

For example, if you add a IBM Campaign Data Folder named MyFolder with the Folder Location temp, in a Snapshot process configuration dialog, File in MyFolder appears in the **Export to** drop-down list. When you select File in MyFolder, the **File Name** field in the Specify Output File window is automatically populated with the relative path temp/.

---

## Chapter 4. Customizing campaigns

You can customize your campaigns by using custom campaign attributes, initiatives, and products.

---

### Custom campaign attributes

**Note:** If your Campaign installation is integrated with Marketing Operations, you must use Marketing Operations to create custom campaign attributes. For details, see the Marketing Operations documentation.

You can customize campaigns by adding custom campaign attributes to store metadata about each campaign.

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

Custom cell attributes are included in the target cell spreadsheet (TCS) for every campaign, even campaigns that have already been created. 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 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 Campaign Generated Fields (UCGFs). 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 *Campaign User's Guide*.

**Note:** If Campaign is integrated with Marketing Operations, you must use Marketing Operations to create custom cell attributes. For details, see the Marketing Operations documentation.

---

## Custom offer attributes

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.

### 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.

---

## Working with custom attributes

The following topics explain how to work with custom attributes that can be used in campaigns, offer templates and offers, or cells on a Target Cell Spreadsheet:

- “To create a custom attribute”
- “To modify a custom attribute” on page 46
- “Using custom attributes in offer templates” on page 51
- “Adding list values to the Channel attribute” on page 58

### To create a custom attribute

The steps for creating custom campaign, offer, or cell attributes are the same. The custom attribute type that you specify during creation determines whether the attribute can be used on campaigns, offer templates and offers, or cells on a Target Cell Spreadsheet.

**Note:** You must have permissions to add custom campaign, offer, and cell attributes. For information, see the *Marketing Platform Administrator's Guide*.

1. Select **Settings > Campaign Settings**.
2. In the **Templates and Customizations** section, click **Custom Attribute Definitions**.

The Custom Attribute Definitions window displays the standard offer attributes and any custom attributes that have already been created.
3. Click the **Add a Custom Attribute** icon.
4. Use the Add Attribute Details window to define the attribute:
  - a. Enter an **Attribute Display Name** to specify the label that users will see. There are no special naming restrictions.
  - b. Enter an **Internal Name**, which will be used to reference this attribute when writing an IBM EMM expression, such as a query or custom macro. The internal name must start with an alphabetic character, may not include spaces, and must be globally unique. Internal names are not case sensitive. It is a best practice to use the same name as the display name without any spaces (for example, "InterestRate").
  - c. Use the **Attribute Of** drop-down list to indicate where this attribute can be used:
    - A **Campaign** attribute will be included on every campaign, even campaigns that have already been created.
    - An **Offer** attribute will be available for use on new offer templates. If the attribute is included on an offer template, then any offers based on that template will include the attribute.
    - A **Cell** attribute will be included in the Target Cell Spreadsheet for every campaign, even campaigns that have already been created.
  - d. Optionally, enter a **Description**.
  - e. Optionally, check the **Mandatory** box to make the value of this attribute required. You can change this setting later, if desired.
    - For campaigns, users will be required to specify a value for this attribute (the field cannot be left blank).
    - For cells, users will be required to specify a value in the Target Cell Spreadsheet (the cell cannot be left blank).
    - For offers, administrators will be required to specify a value when the attribute is added to an offer template. The specified value will be used

for any offers based on that template, unless the user specifies a different value when creating or editing the offer.

**Note:** If you add an offer attribute to an offer template as "Hidden static" or "Parameterized", a value is always required, even if the attribute was defined as non-Mandatory. If you add an offer attribute to an offer template as "Static", the **Mandatory** setting determines whether or not a value is required.

- f. Use the **Form Element Type** list to specify the type of data to be stored in the attribute field for offers or cells.

**Important:** After you add the custom attribute, you cannot change its data type.

- g. Depending on your selection, you may need to enter one of the following:
  - **Maximum String Length:** If you selected **Select Box - String** or **Text Field - String**, you must specify the maximum number of characters that will be stored for any value of this attribute.
  - **Decimal Places:** If you selected **Text Field - Numeric** or **Text Field - Currency**, you must specify the number of decimal places that will be displayed to the right of the decimal point.

**Important:** For **Text Field - Currency**, the currency value reflects the number of decimal places customarily used in the local currency. If you specify a number of decimal places less than the number commonly used, the currency value will be truncated.

- h. If you selected **Select Box - String** as the Form Element Type:
  - Optionally, check **Allow addition of list items from within edit forms** 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.

**Important:** Once the campaign, offer template, or offer is saved, the user cannot remove the new list item. The value is saved back to the custom attribute definition and is then available to all users. Only administrators can remove items from lists, by modifying the custom attribute.

- The **Source list of available values** is the list of items available in the Select Box. Populate the list by entering values in the **New or Selected Item** field and clicking **Accept**. To delete a value, select it in the Source list of available values and click **Remove**.
- Optionally, specify a **Default Value** for the Select Box. The default value will be used on the campaign, offer, or TCS, unless the user specifies a different value when creating or editing the campaign, offer, or cell.
- Specify the **Sort Order** to determine how the values will appear in the list.

5. Click **Save Changes**.

## To modify a custom attribute

The steps for modifying custom attributes for a campaign, offer, or cell are the same.

**Note:** You must have permissions to modify offer and cell custom attributes. For more information, see the *Marketing Platform Administrator's Guide*.

1. Select **Settings > Campaign Settings**.
2. In the **Templates and Customizations** section, click **Custom Attribute Definitions**.

The Custom Attribute Definitions window displays the standard offer attributes and any custom attributes that have already been created.

3. Click the name of the attribute you want to modify.
4. Use the Attributes Details window to make changes:
  - a. **Attribute Display Name:** You can change the display name of custom attributes that you created. You cannot change the display name of the standard offer attributes.
  - b. **Internal Name:** You can change the internal name of custom attributes that you created, but any uses of the internal name by existing flowcharts will cause an error, and you will receive a warning when you save the attribute. You cannot change the internal name of the default offer attributes.

**Important:** Internal names for custom attributes must be globally unique, must start with an alphabetic character, cannot include spaces, and are not case-sensitive.

- c. Optionally, modify the **Description**.
- d. Check or clear the **Mandatory** box, depending on whether you want to require a value for this attribute:
  - If you change from Mandatory to non-mandatory, a value will no longer be required when the attribute is used.
  - If you change from non-mandatory to Mandatory, a value will be 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 will be required to specify a value before saving.
- e. For a **Select Box - String** or **Text Field - String** field, you can modify the **Maximum String Length** to specify the maximum number of characters to store for any value of this attribute.

**Important:** Decreasing the length of an existing attribute will truncate existing values, which can adversely affect response tracking if the field is used for matching purposes.

- f. For a **Text Field - Numeric** or **Text Field - Currency** field, you can modify the **Decimal Places** to specify how many decimal places to the right of the decimal point will be displayed.

**Important:** If you decrease the value, the display will be truncated in the user interface. However, the original value is retained in the database.

- g. If the attribute is a **Select Box - String**, you can:
  - Change **Allow addition of list items from within edit forms**, to allow or disallow users to add new unique values to the list when defining campaigns, offers, or offer templates. This option does not apply to custom cell attributes.
  - Edit a list item: Select it in the **Source list of available values**, change it in the New or Selected Item field, and click **Accept**.
  - Add a list item: Enter a value in the **New or Selected Item** field and click **Accept**.

- Remove a list item: Select it in the **Source list of available values** and click **Remove**.
  - Change the **Default Value**.
  - Change the **Sort Order** to determine the order of items in the list.
5. Click **Save Changes**.

---

## Custom initiatives

Campaign is delivered with a built-in attribute called **Initiative**. The **Initiative** attribute is a drop-down list on the campaign **Summary** tab. Initially, there are no values in the drop-down list; as an administrator, you define the initiatives to select from.

### To add an initiative

You can add initiatives that users can select from the **Initiative** drop-down list in the campaign **Summary** tab. You add initiatives directly in the database table `UA_Initiatives`.

1. Access the Campaign system table database using your database management system.
2. In the database table `UA_Initiatives`, add values in the `InitiativeName` column. Each value can be a maximum of 255 characters.
3. Save the changes to the `UA_Initiatives` table.

---

## Custom products

Users can associate an offer with one or more products. Product IDs are stored in the `UA_Product` table in the Campaign system table database. Initially, there are no records in this table. As an administrator, you can populate this table.

### To add a product

You can add products that users can associate with offers. You add products directly in the database table `UA_Products`.

1. Access the Campaign system table database using your database management system.
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.



---

## Chapter 5. Offer template administration

Before proceeding with offer management administration, you must understand the following:

- “What is an offer?”
- “What is an offer template?”
- “Offer templates and security” on page 50
- “Why use offer templates?” on page 50
- “Custom offer attributes” on page 44

---

### 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 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 Appendix B, “Special characters in Campaign object names,” on page 329.

---

### What is an offer template?

An offer template defines the structure of a particular type of offer. Users create offers by basing them on the offer templates you create.

**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. As part of defining an offer template, you specify the relevant offer attributes and how they will be used.

**Note:** There are no specific restrictions on offer template names.

---

## 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 Campaign security, see the *Marketing Platform Administrator's Guide*.

---

## Why use offer templates?

Offer templates provide you and your Campaign users with the following benefits:

- By creating offer templates, you simplify offer creation for your users because only 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 your campaigns.

---

## Planning offer templates and offers

Planning offers includes considering which template to use, which attributes are parameterized, whether you will use holdout control groups for cells assigned this offer, and other considerations.

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 *Campaign User's Guide*.

---

## Working with offer attributes

The following topics explain how to work with offer attributes:

- “Using custom attributes in offer templates”
- “To create a custom attribute” on page 45
- “To modify a custom attribute” on page 46
- “Standard offer attributes in Campaign” on page 54

### Using custom attributes in offer templates

Once you have created a custom offer attribute, it can be added to any new offer template. Any offers that are created based on that template will include the custom attribute. For related information, see “Using drop-down lists in offer templates” on page 55.

### To create a custom attribute

The steps for creating custom campaign, offer, or cell attributes are the same. The custom attribute type that you specify during creation determines whether the attribute can be used on campaigns, offer templates and offers, or cells on a Target Cell Spreadsheet.

**Note:** You must have permissions to add custom campaign, offer, and cell attributes. For information, see the *Marketing Platform Administrator's Guide*.

1. Select **Settings > Campaign Settings**.
2. In the **Templates and Customizations** section, click **Custom Attribute Definitions**.

The Custom Attribute Definitions window displays the standard offer attributes and any custom attributes that have already been created.

3. Click the **Add a Custom Attribute** icon.
4. Use the Add Attribute Details window to define the attribute:
  - a. Enter an **Attribute Display Name** to specify the label that users will see. There are no special naming restrictions.
  - b. Enter an **Internal Name**, which will be used to reference this attribute when writing an IBM EMM expression, such as a query or custom macro. The internal name must start with an alphabetic character, may not include spaces, and must be globally unique. Internal names are not case sensitive. It is a best practice to use the same name as the display name without any spaces (for example, "InterestRate").
  - c. Use the **Attribute Of** drop-down list to indicate where this attribute can be used:
    - A **Campaign** attribute will be included on every campaign, even campaigns that have already been created.
    - An **Offer** attribute will be available for use on new offer templates. If the attribute is included on an offer template, then any offers based on that template will include the attribute.
    - A **Cell** attribute will be included in the Target Cell Spreadsheet for every campaign, even campaigns that have already been created.

- d. Optionally, enter a **Description**.
- e. Optionally, check the **Mandatory** box to make the value of this attribute required. You can change this setting later, if desired.
  - For campaigns, users will be required to specify a value for this attribute (the field cannot be left blank).
  - For cells, users will be required to specify a value in the Target Cell Spreadsheet (the cell cannot be left blank).
  - For offers, administrators will be required to specify a value when the attribute is added to an offer template. The specified value will be used for any offers based on that template, unless the user specifies a different value when creating or editing the offer.

**Note:** If you add an offer attribute to an offer template as "Hidden static" or "Parameterized", a value is always required, even if the attribute was defined as non-Mandatory. If you add an offer attribute to an offer template as "Static", the **Mandatory** setting determines whether or not a value is required.

- f. Use the **Form Element Type** list to specify the type of data to be stored in the attribute field for offers or cells.

**Important:** After you add the custom attribute, you cannot change its data type.

- g. Depending on your selection, you may need to enter one of the following:
  - **Maximum String Length:** If you selected **Select Box - String** or **Text Field - String**, you must specify the maximum number of characters that will be stored for any value of this attribute.
  - **Decimal Places:** If you selected **Text Field - Numeric** or **Text Field - Currency**, you must specify the number of decimal places that will be displayed to the right of the decimal point.

**Important:** For **Text Field - Currency**, the currency value reflects the number of decimal places customarily used in the local currency. If you specify a number of decimal places less than the number commonly used, the currency value will be truncated.

- h. If you selected **Select Box - String** as the Form Element Type:
  - Optionally, check **Allow addition of list items from within edit forms** 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.

**Important:** Once the campaign, offer template, or offer is saved, the user cannot remove the new list item. The value is saved back to the custom attribute definition and is then available to all users. Only administrators can remove items from lists, by modifying the custom attribute.

- The **Source list of available values** is the list of items available in the Select Box. Populate the list by entering values in the **New or Selected Item** field and clicking **Accept**. To delete a value, select it in the Source list of available values and click **Remove**.

- Optionally, specify a **Default Value** for the Select Box. The default value will be used on the campaign, offer, or TCS, unless the user specifies a different value when creating or editing the campaign, offer, or cell.
- Specify the **Sort Order** to determine how the values will appear in the list.

5. Click **Save Changes**.

## To modify a custom attribute

The steps for modifying custom attributes for a campaign, offer, or cell are the same.

**Note:** You must have permissions to modify offer and cell custom attributes. For more information, see the *Marketing Platform Administrator's Guide*.

1. Select **Settings > Campaign Settings**.
2. In the **Templates and Customizations** section, click **Custom Attribute Definitions**.

The Custom Attribute Definitions window displays the standard offer attributes and any custom attributes that have already been created.

3. Click the name of the attribute you want to modify.
4. Use the Attributes Details window to make changes:
  - a. **Attribute Display Name:** You can change the display name of custom attributes that you created. You cannot change the display name of the standard offer attributes.
  - b. **Internal Name:** You can change the internal name of custom attributes that you created, but any uses of the internal name by existing flowcharts will cause an error, and you will receive a warning when you save the attribute. You cannot change the internal name of the default offer attributes.

**Important:** Internal names for custom attributes must be globally unique, must start with an alphabetic character, cannot include spaces, and are not case-sensitive.

- c. Optionally, modify the **Description**.
- d. Check or clear the **Mandatory** box, depending on whether you want to require a value for this attribute:
  - If you change from Mandatory to non-mandatory, a value will no longer be required when the attribute is used.
  - If you change from non-mandatory to Mandatory, a value will be 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 will be required to specify a value before saving.
- e. For a **Select Box - String** or **Text Field - String** field, you can modify the **Maximum String Length** to specify the maximum number of characters to store for any value of this attribute.

**Important:** Decreasing the length of an existing attribute will truncate existing values, which can adversely affect response tracking if the field is used for matching purposes.

- f. For a **Text Field - Numeric** or **Text Field - Currency** field, you can modify the **Decimal Places** to specify how many decimal places to the right of the decimal point will be displayed.

**Important:** If you decrease the value, the display will be truncated in the user interface. However, the original value is retained in the database.

g. If the attribute is a **Select Box - String**, you can:

- Change **Allow addition of list items from within edit forms**, to allow or disallow users to add new unique values to the list when defining campaigns, offers, or offer templates. This option does not apply to custom cell attributes.
- Edit a list item: Select it in the **Source list of available values**, change it in the New or Selected Item field, and click **Accept**.
- Add a list item: Enter a value in the **New or Selected Item** field and click **Accept**.
- Remove a list item: Select it in the **Source list of available values** and click **Remove**.
- Change the **Default Value**.
- Change the **Sort Order** to determine the order of items in the list.

5. Click **Save Changes**.

## Standard offer attributes in Campaign

The following table lists the standard offer attributes that are delivered with Campaign.

*Table 18. Standard offer attributes*

Attribute Display Name	Attribute Internal Name	Form Element Type
Average Response Revenue	AverageResponseRevenue	Text Field - Currency
Channel	Channel	Select Box - String
Channel Type	ChannelType	Select Box - String
Cost Per Offer	CostPerOffer	Text Field - Currency
Creative URL	CreativeURL	Text Field - String
Effective Date	EffectiveDate	Text Field - Date
Expiration Date	ExpirationDate	Text Field - Date
Duration	ExpirationDuration	Text Field - Numeric
Fulfillment Cost	FulfillmentCost	Text Field - Currency
Interaction Point ID	UACInteractionPointID	Text Field - Numeric
Interaction Point	UACInteractionPointName	Text Field - String
Offer Fixed Cost	OfferFixedCost	Text Field - Currency

---

## 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 Chapter 2, “Managing security in IBM Campaign,” on page 3.

## Using drop-down lists in offer templates

A drop-down list, also known as a Select Box, is a list of values from which the user can select a single item. To make a drop-down list available in offer templates (and therefore in offers), define a custom offer attribute of the type **Select Box - String**, and add it to an offer template. Any offers based on that template will include the list.

The values that are available in the list are specified when the custom attribute is created. Additionally, you can specify the **Sort Order** of the values, a **Default Value**, and whether a value for this attribute is required (**Mandatory**). You can optionally check **Allow addition of list items from within edit forms** to allow users to add new unique values to the list when editing the offer template or creating an offer based on the template. For example, if a Select Box contains the values *Small*, *Medium*, *Large*, any user could add the value *Extra Large*.

**Note:** Once the offer template or offer is saved, the user cannot remove the new list item. The value is saved back to the custom attribute definition and is then available to all users. Only administrators can remove items from lists, by modifying the custom attribute.

The **Mandatory** setting determines whether a value is required for this attribute. If **Mandatory** is selected, administrators will be required to specify a value when the attribute is added to an offer template. The specified value will be used for any offers based on that template, unless the user specifies a different value when creating or editing the offer.

**Note:** If you add an offer attribute to an offer template as “Hidden static” or “Parameterized”, a value is always required, even if the attribute was defined as non-Mandatory. If you add an offer attribute to an offer template as “Static”, the **Mandatory** setting determines whether or not a value is required.

## To create an offer template

1. Select **Settings > 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...**  
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 Interact, select **Allow offers created from this template to be used in real-time interactions**.
- 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 (<< and >>) 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 you have added to the offer template, supply a default value that is 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 have 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, 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.

## To modify an offer template

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. Under **Templates and Customizations**, click **Offer Template Definitions**.  
The Offer Template Definitions window opens.
3. Click the name of the offer template to modify.  
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 (<< and >>) 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 a radio box **Flowchart run date**. If you select this option instead of entering a default Offer Valid date, Campaign uses the date on which the process that uses the offer runs, not the entire flowchart.

8. Click **Finish**.

## To reorder offer templates

The order of offer templates you define is the order templates are presented to users when they create a new offer. By default, offer templates are listed in the order in which you created them. However, users see only the specific offer templates allowed by the security policy of the offer template and the roles of the user; therefore, the user may not see all offer templates. However, the order of the ones they do see is the order you specify.

1. Select **Settings > 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 top or bottom of the list of offer templates, click **Reorder...**

The Reorder Offer Templates window opens, listing offer templates in the current order.

4. To change the order of offer templates, select one template at a time and click the up or down icons to move that templates location in the list.
5. When the offer templates are in the desired order, click **Save Changes**.

## To retire an offer template

You retire an offer template when you no longer want users to be able to create new offers based on a template. Previously created offers based on the template you are retiring 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**.

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. To the right of the offer template you no longer want available, click **Retire**.

You are prompted to confirm the retirement.

4. Click **OK** to retire the offer template, or **Cancel** to stop the operation.

If you click **OK**, the offer template is retired, and its status is indicated on the screen.

## Adding list values to the Channel attribute

Campaign includes the **Channel** custom attribute for use in offer templates. The **Channel** attribute is intended to indicate the outbound communication channel for the offer, such as e-mail or phone.

As delivered, the **Channel** attribute, of type **Select Box - String**, does not include any available values. To make use of the **Channel** attribute, you must provide the values from which users can select by modifying the attribute.

You can also allow users to enter additional values for the field by checking Allow additional of list items from within edit forms.

## Template icons

You select a template icon as part of the basic options when you create or modify an offer template. The template icon serves as a visual clue about the offer template when users create new offers.

The following table lists and shows the available template icons:

*Table 19. Offer template icons*











Icon Name	Icon
offertemplate_default.gif	

Table 19. Offer template icons (continued)

Icon Name	Icon
offertemplate_manychans.gif	
offertemplate_manydates.gif	
offertemplate_manyresp.gif	
offertemplate_manysegs.gif	
offertemplate_repeatingtabl.gif	
offertemplate_simpleemail.gif	
offertemplate_simplemail.gif	
offertemplate_simplephone.gif	
offertemplate_versions.gif	

## 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 Marketing Operations assets in Campaign offers

If both Marketing Operations and Campaign are installed, and you have licensed the IBM Marketing Asset Management add-on for Marketing Operations, your campaigns can include digital assets from your Marketing Operations asset libraries. For example, an offer can include a product logo that is stored in a Marketing Operations asset library.

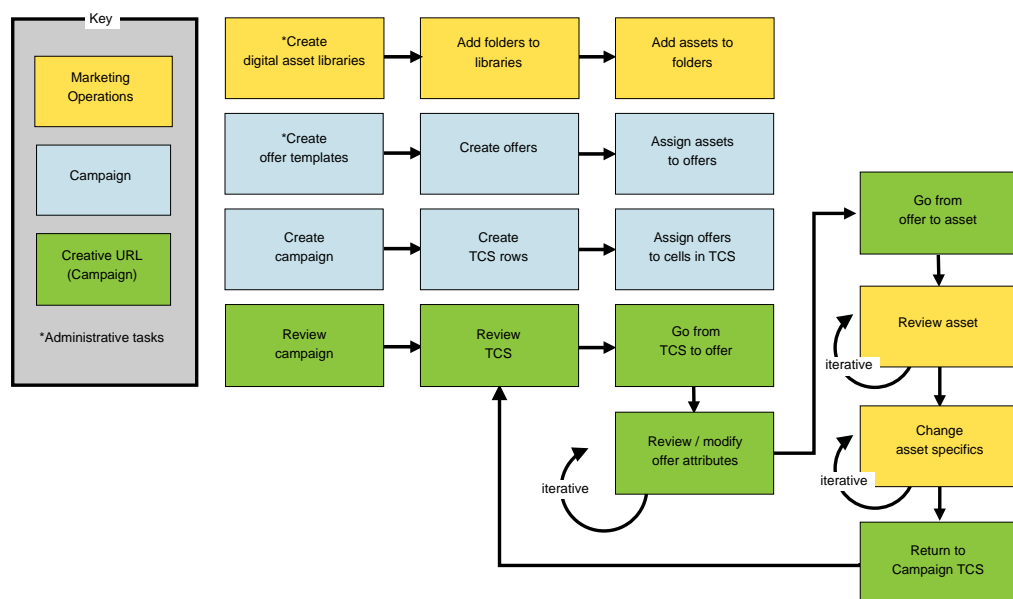
**Note:** Campaign may or may not be integrated with Marketing Operations.

To include an 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 Marketing Operations. The asset that the **CreativeURL** attribute points to is included in the offer.

The **CreativeURL** attribute allows users to move seamlessly from Campaign to Marketing Operations 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 Marketing Operations, 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 that does not integrate Marketing Operations with Campaign. Your workflow might differ.



## Guidelines for using Marketing Operations assets in Campaign offers

This topic lists the prerequisites and requirements for using Marketing Operations assets in Campaign offers. This functionality depends on the **CreativeURL** offer attribute.

- Both Marketing Operations and Campaign must be installed. (The **CreativeURL** attribute is installed with Campaign. However, you cannot use the feature unless Marketing Operations is also installed.)
- You must license the IBM Marketing Asset Management add-on for Marketing Operations.
- Campaign may or may not be integrated with Marketing Operations. Even if UMO-UC integration is off, users can assign an asset to an offer.
- **CreativeURL** is a standard 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 Marketing Operations 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 Marketing Operations.

**Note:** An offer can be related to only one asset. However, one asset can be related to multiple offers.

## Setting up Marketing Operations assets for use in Campaign offers

This topic explains what administrators must do to allow Campaign users to relate a digital asset from Marketing Operations to a 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 Campaign offer, you use the **CreativeURL** attribute. The **CreativeURL** attribute is a standard offer attribute that is installed with Campaign. A "Creative URL" is a pointer to a file in a Marketing Operations asset library.

Table 20. Setting up Marketing Operations assets for use in Campaign offers

Task	Details	For documentation
Create a library to hold digital assets.	This task is typically done by a Marketing Operations administrator.  In IBM Marketing Operations, select <b>Settings &gt; Marketing Operations Settings</b> , click <b>Asset Library Definitions</b> , and add a library.	<i>IBM Marketing Operations Administrator's Guide</i>
Add assets to the library.	This task is typically done by Marketing Operations users.  In IBM Marketing Operations, 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.	<i>IBM Marketing Operations User's Guide</i>

Table 20. Setting up Marketing Operations assets for use in Campaign offers (continued)

Task	Details	For documentation
Create an offer template that includes the <b>CreativeURL</b> attribute.	<p>This task is typically done by a Campaign administrator.</p> <p>Offer attributes are the fields that define an offer. <b>CreativeURL</b> is a standard attribute supplied with Campaign. Adding the <b>CreativeURL</b> attribute to a template makes the attribute available to any offers based on that template.</p> <p>For example, in a system that does not integrate Marketing Operations with 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. The URL to the asset is now included in the <b>Creative URL</b> field.</li> <li>4. Save the offer template.</li> </ol>	<p><i>Campaign Administrator's Guide:</i></p> <p>"To create an offer template" on page 55</p> <p>For systems that do integrate Marketing Operations and Campaign, see the <i>IBM Marketing Operations and Campaign Integration Guide</i></p>
Use Campaign to create an offer that includes an asset from Marketing Operations.	<p>Campaign users can now create offers based on a template that includes the <b>CreativeURL</b> attribute. When defining the offer, users can go to the asset library and select or create an asset.</p>	<i>Campaign User's Guide</i>

---

## Chapter 6. Audience level administration

As a Campaign administrator, you can perform the following tasks:

- Create the audience levels needed for your company's campaigns.
- Create database tables in the Campaign system database to support the new audience levels.
- Map system tables to the supporting database tables for the new audience levels in the 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 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 *Campaign User's Guide*.

---

## Why different audience levels are needed in 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 *Campaign User's Guide*.

---

## The default Customer audience level

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 Campaign system database contains the tables needed to support the Customer audience level. After installing Campaign, you must map these tables.

---

## Creating additional audience levels

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.

---

## About audience levels and system tables

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

These system table entries are automatically created when you create the audience level. You must then:

- Create the physical database tables in the Campaign system table database.
- Map these system tables to the database tables.

**Note:** IBM recommends that you map the segment membership table only if you are using strategic segments with Campaign flowcharts or Optimize sessions in Contact Optimization.

### Default Customer audience level system tables

Campaign is delivered with system table ddl scripts to create the tables that support the default Customer audience level.



After installing Campaign, you must map these system tables to tables in the Campaign system database as follows:

*Table 21. Default audience level system tables*

IBM 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 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:** IBM 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 Campaign flowcharts or Contact Optimization sessions is optional. If you map the segment membership table, each time you run the flowchart or Optimize session, Campaign or Contact Optimization 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 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 Appendix B, “Special characters in Campaign object names,” on page 329

## 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”
- “Required fields in the detailed contact history table”
- “Required fields in the response history table” on page 67
- “Required fields in the segment membership table” on page 67

### Required fields in the contact history table

The contact history table for each audience level in the Campaign system database must minimally contain the fields described in this section.

*Table 22. 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
No	ContactDateTime	datetime	8	Yes
No	UpdateDateTime	datetime	8	Yes
No	ContactStatusID	bigint	8	Yes
No	DateID	bigint	8	Yes
No	TimeID	bigint	8	Yes

**Note:** 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 Campaign system database must minimally contain the fields described in this section.

*Table 23. 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	DateID	bigint	8	No
No	TimeID	bigint	8	No

## Required fields in the response history table

The response history table for each audience level in the Campaign system database must minimally contain the fields described in this section.

Table 24. 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
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 Campaign or Contact Optimization, 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 25. 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

This section contains the following information:

- “User tables with a single audience level”
- “User tables with multiple audience levels” on page 68

### 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, 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 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. 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.

---

## Working with audience levels

This section contains the following information:

- "Workflow for setting up a new audience level"
- "Removing an audience level" on page 71

### Workflow for setting up a new audience level

The following steps 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 69
- "Task 2: Create the new audience levels in Campaign" on page 69
- "Task 3: Map the IBM Campaign system tables to the database tables" on page 70
- "Task 4: Map the user tables containing relevant data to the appropriate audience level(s)" on page 71

- “Task 5: Save the mapped tables in a table catalog” on page 71

### Task 1: Create the required database tables for each new audience level

You must create physical database tables in the 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 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. IBM 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 Campaign

1. Select **Settings > Campaign Settings**.  
The Campaign Settings page appears.
2. On the Campaign Settings page, under **Data Source Operations**, click **Manage Audience Levels**.  
The Audience Levels window opens, displaying existing audience levels.
3. Click **Create New**.  
The Create New Audience Level window opens.
4. Enter a unique **Audience Level Name**, which should reflect the group of identifiers at that audience level.  
Audience level names have no 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. Fields can only be mapped in “Task 3: Map the IBM Campaign system tables to the database tables” on page 70 if Campaign finds an exact match for the field names.

For example, if you are creating the new audience level "Household" and you specify one field for unique audience level identifier called "HouseholdID," you must make sure 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 window, when you select the new audience level, you see the required tables listed as "Not Mapped." You perform the mapping in "Task 3: Map the IBM Campaign system tables to the database tables."

### **Task 3: Map the IBM Campaign system tables to the database tables**

After creating the physical database tables for each new audience level and the audience levels in Campaign, you must map the IBM Campaign system tables to these database tables.

You can map user tables to your created audience levels without mapping the IBM 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.

IBM recommends that you map segment membership system tables to physical database tables only for audiences used in Campaign flowcharts or Contact Optimization sessions that contain strategic segments. Using strategic segments in Campaign and Contact Optimization is optional.

1. Select **Settings > Campaign Settings**.  
The Campaign Settings page appears.
2. On the Campaign Settings page, under **Data Source Operations**, click **Manage Audience Levels**.  
The Audience Levels window opens, displaying existing audience levels.
3. Select the audience level for which you are mapping database tables, and click **History Tables**.
4. On the Table Mappings window, select each IBM Campaign system table and click **Map Table**.
5. On the table mapping window, select the database table corresponding to the IBM Campaign system table for that audience level. The Source Table Fields list is populated with the fields from the database table you selected. The Required Fields list is populated with Selected Fields (from your source database table) and the corresponding Required Fields (from the IBM Campaign system table).

**Important:** Fields can only be mapped if 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 IBM 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)**

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**

(Optional). Save your mapped tables in a table catalog so that you can reload the catalog without remapping individual tables.

### **Removing an audience level**

**Important:** Do not remove an audience level that has been used within Campaign, as this will result in significant system problems, as described below.

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:** Before removing an audience level, IBM recommends backing up the entire 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**

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**

**Note:** Specifying and managing global suppression segments requires the "Manage Global Suppressions" permission in Campaign.

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 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 *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”
- “To specify the segment as a global suppression segment” on page 73

### To create the global suppression segment in a flowchart

**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

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 *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 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

1. After creating the segment that you want to use as a global suppression segment, in 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 Marketing 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 *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 *Campaign User's Guide*.

When the flowchart that created the global suppression segment is deleted, the segment is also deleted.

## Logging for global suppressions

The following information relating to global suppressions is included in the flowchart log:

- 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

Before you begin working with contact history, you should read all audience level administration topics and set up required audience levels.

Contact history is stored in the Campaign system database in separate tables by audience level; you therefore need to set up audience levels prior to working with contact history.

In addition, you can find basic concepts about contact history, and information about setting up flowcharts to record contact history, in the *Campaign User's Guide*.

---

### Contact history concepts

This section contains the following information:

- “What is contact history?”
- “What is detailed contact history?” on page 76
- “What is contact status?” on page 76
- “About updating contact status” on page 76
- “How does contact history relate to audience levels?” on page 77
- “How does contact history relate to database and system tables?” on page 77
- “What is offer history?” on page 77
- “What is treatment history?” on page 77

### 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 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 Campaign system database.

Contact history is recorded and stored in the 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\_DtlContactHist) 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.

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.

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 CountsAsContactfield 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?

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 Campaign system database.

## How does contact history relate to database and system tables?

Contact history tables, which must exist in the 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 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 Campaign system database. You can use the tables for the example Customer audience level as templates.

After you create the tables in the 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 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 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.

---

## Working with contact history

This section contains the following information:

- "Creating contact history tables for new audience levels"
- "Mapping contact history tables to system tables"
- "To add a contact status code"
- "Logging history" on page 79
- "Updating contact history" on page 80
- "Clearing contact history" on page 80

### Creating contact history tables for new audience levels

When you create a new audience level, you may need to create a table in the 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 )
```

### Mapping contact history tables to system tables

Each time you create a new audience level, you must map the contact history and detailed contact history system tables for the new audience level.

**Important:** You can delete contact statuses you do not intend to use, but you should not delete a contact status that is in use.

### To add a contact status code

You can add your own contact status codes to supplement the contact statuses that are delivered with Campaign. Define new contact status codes in the

UA\_ContactStatus table in the Campaign system database. A contact status indicates the type of contact made (for example, Delivered, Undelivered, Control).

Before you add a contact status, see *Default contact status codes* to determine whether the existing status codes meet your needs.

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 add contact statuses if the ones supplied with Campaign do not meet your needs.

1. Log in to the database management system that contains the 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 Marketing 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 Contact Optimization 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.

## Logging history

To log contact history, users configure one or more contact processes (Call List or Mail List). Then, when the flowchart is run in production (not test) mode, contact history is written to the tables related to the audience-levels used in the flowchart.

When logging to contact history is enabled, offer and treatment history are written as well.

**Note:** If the process is configured to log contact history, but runs on a cell with no selected targets, no history records are written.

Users can optionally choose to not have contact history logged by the Call List or Mail List process.

For more information, see the *Campaign User's Guide*.

## Updating contact history

To update contact history (for example, to update a contact status or additionally tracked field in contact history), users configure the Track process. For example, the input to the Track process may be an updated contact list received from a mail house with the list of targets who could not be contacted. When the flowchart containing the Track process is run in production mode, contact history is updated to the tables related to the audience-levels used.

For more information, see the *Campaign User's Guide*.

## 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 *Campaign User's Guide*.

---

## Default contact status codes

Campaign is delivered with the following contact statuses, defined in the UA\_ContactStatus table.

Table 26. 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 Campaign system database in separate tables by audience level; you therefore need to set up audience levels prior to working with response history.

In addition, you can find basic concepts about contact and response history, and information about setting up flowcharts to use the Response process, in the *Campaign User's Guide*.

---

### Response history concepts

This section contains the following information:

- “What is response history?”
- “What is the response type?”
- “How does response history relate to audience levels?”
- “How does response history relate to database tables?” on page 82
- “Foreign key constraints in response history tables” on page 82
- “What is an action table?” on page 82
- “What does an action table contain?” on page 82
- “Why use an action table?” on page 82

#### What is response history?

Response history is the historical record of responses to campaigns, either by targeted respondents, or by members of hold-out control groups, who might have performed the desired action despite not having been contacted.

For more information about response history in Campaign, and how to design flowcharts to record responses, see the *Campaign User's Guide*.

#### What is the response type?

Response type is the indicator of the type of response made by the target. Campaign users specify the response type to use when configuring a Response process.

Campaign is delivered with a set of default response types. As an administrator, you can add additional types.

For more information about response types in Campaign, see the *Campaign User's Guide*.

#### How does response history relate to audience levels?

Campaign records and maintains a separate response history for each audience level you have defined. Each audience level has its own related response history table in the Campaign system database, as well as a related IBM Campaign system table.

## How does response history relate to database tables?

Response history tables, which must exist in the 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 Campaign system database.

If you create an additional audience level, you must create the response history table for it in the Campaign system database.

After you create the table in the Campaign system database for a new audience level, you must map the new table to the IBM 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 you create for a new audience level must have a foreign key constraint on the UA\_Treatment table's TreatmentInstID field. See the DDL file that creates the system tables for details on how to set up this constraint.

## What is an action table?

An action table is an optional table containing data about targets' responses to a campaign. An action table typically provides each target's response type and actions of interest, as well as other campaign-specific data. The action table then serves as the source data of the input cell for the Response process.

An action table is audience level-specific; you typically create one action table for each audience level in Campaign.

**Important:** Ensure that any action table used for response tracking is locked during response processing. You must also clear rows after response processing to ensure that responses are not credited multiple times.

## What does an action table contain?

Each row of the action table represents a single event, which must minimally contain the audience IDs, the response type, and the response date. It 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 attributes and any fields that are NULL are ignored.

## Why use an action table?

Using an action table is a best practice meant to ensure that sufficient data about targets' responses are recorded and available to use. Campaign is delivered with a sample action table for the Customer audience level, called UA\_ActionCustomer in the system database.

---

## Working with response history

This section contains the following information:

- “Creating response history tables for new audience levels”
- “Mapping response history tables to IBM Campaign system tables”
- “To set the number of days after an offer expires to record responses”
- “Adding a response type”
- “Logging response history” on page 84

### Creating response history tables for new audience levels

When you create a new audience level, you must create a table in the 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 )
```

### Mapping response history tables to IBM Campaign system tables

After you create the response history table for a new audience level, you must map it to the IBM Campaign system table for the audience level response history.

### To set the number of days after an offer expires to record responses

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

On the Configuration page, set the allowResponseNDaysAfterExpiration property in the applications > Campaign > partitions > partition[n] > server > flowchartConfig category to the desired number of days.

### Adding a response type

You define response types in the UA\_UsrResponseType table in the Campaign system database.

Before adding response types, see “Default response types” on page 84 to determine which existing types meet your needs and which you must create.

To add a response type:

1. Log in to the database management system that contains the Campaign system database.
2. Open the UA\_UsrResponseType table.
3. Add rows for response types that you need to add. For each new type:
  - 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, or 2 if it represents a reject.
  - f. In the IsDefault column, enter 0 if that type is not to be the default, or 1 if it is. Ensure that only one row has the value 1 in this column.
4. Save the table changes.
  5. Remap the UA\_UsrResponseType system table.  
See the documentation for your database management system for detailed instructions on modifying data in tables.

## 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 *IBM Campaign User's Guide*.

---

## Response history references

This section contains the following information:

- “Default response types”
- “Sample UA\_ActionCustomer Table” on page 85

## Default response types

Campaign is delivered with the following response types, defined in the UA\_UsrResponseType table.

For each response type, valid values for the CountsAsResponse field are:

0 - does not count as a response

1 - counts as a positive response

2 - counts as a negative response

**Note:** 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.

*Table 27. Default response types*

Response-TypeID	Name	Description	Response-StatusCode	Counts-AsResponse	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

## Sample UA\_ActionCustomer Table

Campaign is delivered with the sample action table UA\_ActionCustomer. The fields in this table are intended as examples of fields that may be useful for generating a response history.

*Table 28. 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
ProductID	bigint	8	No
ResponseTypeCode	varchar	64	Yes



---

## Chapter 9. Operational monitoring

You use operational monitoring to see a single view of all active flowcharts.

Operational monitoring is an administration function. Only users with the Access monitoring page or Perform monitoring tasks security permission are allowed to view the Operational Monitoring page. Only users with the Perform monitoring tasks security permission are allowed to start, stop, or suspend flowcharts.

With the Perform monitoring tasks permission, a user is allowed to control all displayed flowcharts, regardless of the normal access rights they may have to each individual flowchart. Do not give this permission to end-users unless you intend to give them the right to stop, pause, and resume any running flowchart.

---

### To configure operational monitoring

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

You must configure operational monitoring appropriately for your environment, including setting parameters for how long monitoring information is stored and displayed for historical flowchart runs.

On the Configuration page, set the properties in the Campaign > monitoring category as required. For information on the properties, see the context help or the *Marketing Platform Administrator's Guide*.

---

### To access the All Monitored Runs page

**Note:** You must have appropriate permissions to access the Monitoring page. For more information, see the *Marketing Platform Administrator's Guide*.

Select **Campaign > Monitoring**. The All Monitored Runs page appears.

---

### Viewing the All Monitored Runs page

On the All Monitored Runs page, Campaign groups active flowcharts by the campaigns that they belong to.

The status for each flowchart is indicated two ways: in the **Status** column, and by the colored status indicator. The action buttons that are available for each flowchart depend on the flowchart's status.

For information about the colors and valid actions corresponding to each status, refer to the table in "Flowchart states and actions" on page 89.

**Note:** The action buttons are only available to you if you have the Perform monitoring tasks security permission.

---

## To sort the list of flowcharts on the All Monitored Runs page

By default, flowcharts are sorted in ascending order by campaign name. You can also sort the list of flowcharts by the **Status**, **Run By**, **Start Time**, or **End Time** column.

To sort the list of flowcharts, click the column name by which you want to sort.

The direction of the arrow to the right indicates if the column is sorted in ascending or descending order:

- An up arrow indicates the column is sorted in ascending order.
- An down arrow indicates the column is sorted in descending order.

To reverse the sorting order, click the column name again.

**Note:** If you leave the All Monitored Runs page and return to it, the flowcharts are listed in the default sort order (ascending by campaign name).

## To view associated campaigns or flowcharts

From the All Monitored Runs page, you can open a flowchart or campaign summary. Blue underlining indicates that campaign or flowchart names are hypertext links.

To view a campaign summary, click the name of the campaign, which appears to the left in the Campaign and Flowchart(s) column.

To view a flowchart in **Read Only** mode, click the name of the flowchart, which appears to the right of the campaign name, in italics.

## To refresh the All Monitored Runs page display

The Refresh feature allows you to update the contents of the All Monitored Runs page, to ensure that you are viewing current operational details.

To refresh the All Monitored Runs page, click **Refresh** at the top right. The page refreshes with current data.

---

## Working with flowcharts through the All Monitored Runs page

**Note:** You must have permissions to work with flowcharts from the All Monitored Runs page.

If you have the Perform monitoring tasks security permission, you can perform the following actions on flowcharts on the All Monitored Runs page. The actions that you can perform on a flowchart depend on its current status.


**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 more details, see the *Campaign User's Guide*.

## To stop a running flowchart

You can perform the Stop action only on a running flowchart.

1. On the All Monitored Runs page, locate the flowchart that you want to stop. You see its status and the available action buttons.




2. Click the Stop button  next to the flowchart status.  
The flowchart stops. Its status on the All Monitored Runs page changes to **Stopped** and the status indicator color changes to red.

## To suspend 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. On the All Monitored Runs page, locate the flowchart that you want to suspend. You see its status and the available action buttons.

2. Click the Suspend button  next to the flowchart status.  
The suspend process starts. The flowchart's status on the All Monitored Runs page 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's status changes to **Suspended**; the color of the status indicator remains yellow.

## To resume a suspended flowchart

You can resume a suspended flowchart. This restarts the flowchart and continues running it at the point at which you suspended it.

1. On the All Monitored Runs page, locate the suspended flowchart that you want to resume. You see its status and the available action buttons.

2. Click the Resume button  next to the flowchart status.  
The flowchart resumes running. Its status on the All Monitored Runs page changes to **Running** and the color of the status indicator changes to green.

---

## Operational monitoring references

This section contains the following reference information:

- “Flowchart states and actions”
- “Properties related to operational monitoring” on page 91

## Flowchart states and actions

This section describes the valid flowchart states and the actions that are available for each status, as shown on the All Monitored Runs page.

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 29. Flowchart states and actions

Status (Status Indicator 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 Monitoring 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> <p>For details about pausing and continuing a flowchart run, see the <i>Campaign User's Guide</i>.</p>	None from the Monitoring page (Run>Continue from the flowchart)
Suspending (Yellow)	The flowchart "Suspend" action has been initiated from the Monitoring 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 Monitoring 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 <i>Campaign User's Guide</i> .	None

Table 29. Flowchart states and actions (continued)

Status (Status Indicator 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

## Properties related to operational monitoring

Use the properties in the Campaign > monitoring category on the Marketing Platform Configuration page to modify the behavior of operational monitoring. For information about the properties, see the context help or the *IBM Marketing Platform Administrator's Guide*.

- cacheCleanupInterval
- cacheRunCompleteTime
- monitorEnabled
- serverURL
- monitorEnabledForInteract
- protocol
- port

## All Monitored Runs page icons

The All Monitored Runs page uses the following icons



The icons, left to right, are described in the following table.

Table 30. Icons used on the All Monitored Runs page

Icon Name	Description
Print this Item	Select one or more monitored runs by clicking the check box next to each item, then click this icon to print the selected items.
Refresh	Click this icon to refresh the list of monitored runs on the page.



---

## Chapter 10. Dimension hierarchy administration

This section contains the following information:

- “What is a dimension hierarchy?”
- “Why use dimension hierarchies?”
- “About dimension hierarchies and cubes” on page 94
- “About dimension hierarchies and database tables” on page 94

---

### 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).

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 *Campaign User's Guide*.

---

## About dimension hierarchies and database tables

When you create a dimension hierarchy in 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 IBM EMM 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 31. Dimension hierarchy database table*

Dimension-Name	Dim1Name	Dim2Name	Dim3Name	Expression	Datasource
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

Table 31. Dimension hierarchy database table (continued)

Dimension-Name	Dim1Name	Dim2Name	Dim3Name	Expression	Datasource
MemberAge	All Ages	Over 50	Over 60	age > 60	Your datamart

## Working with dimension hierarchies

This section contains the following information:

- “About working with dimension hierarchies”
- “Designing dimension hierarchies”
- “Accessing dimension hierarchies in Campaign”
- “To create a dimension hierarchy” on page 96
- “To load a stored dimension hierarchy” on page 96
- “To edit a dimension hierarchy” on page 97
- “To update a dimension hierarchy” on page 97
- “To remove a dimension hierarchy” on page 97

### About working with dimension hierarchies

To use dimension hierarchies in Campaign, you must do the following:

- Define and create a dimension hierarchy in a database table in your data mart or defined in a delimited or fixed-width flat file.
- Map this table or flat file to a dimension hierarchy in Campaign.

When this dimension hierarchy is mapped into Campaign, the dimension hierarchy is available in a Cube process for building dynamic data cubes on strategic segments.

You or an IBM consulting team must create the dimension hierarchy definition in the data mart or flat file. This is an operation external to Campaign. Also, the lowest level of the dimension hierarchy must use either raw SQL or a pure IBM EMM expression (no custom macros, user variables, or derived fields) to define the individual audience ID membership for each bin.

### Designing dimension hierarchies

You should consider the following 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.

### Accessing dimension hierarchies in Campaign

You can work with dimension hierarchies:

- When editing a flowchart, by clicking the **Admin** icon and selecting **Dimension Hierarchies**.
- Through the Campaign Settings page, by clicking **Manage Dimension Hierarchies**.

**Note:** When dimension hierarchies are used to create cubes, we recommend that you use the Cube process to create dynamic data cubes from a flowchart in the Sessions area of the application.

## To create a dimension hierarchy

These instructions assume that a table with a dimension hierarchy definition is already available.

1. Open the Dimension Hierarchies window.
2. In the Dimension Hierarchies window, click **New Dimension**.  
The Edit Dimension window opens.
3. Enter the following 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 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. 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 window 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 “To map a base record table to an existing database table” on page 28.  
After you finish mapping your dimension hierarchy, you return to the Edit Dimension window, which now includes details for the new dimension hierarchy.
6. Click **OK**.  
You return to the Dimensions window.
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.

## To load a stored dimension hierarchy

Dimension hierarchies are stored in table catalogs with any other mapped tables in the flowchart.

1. Open the Dimension Hierarchies window.
2. Click **Load**.
3. Select the table catalog containing the dimension hierarchies that you want to load.
4. Click **Load Catalog**. The dimension hierarchies are loaded.



## To edit a dimension hierarchy

1. Open the Dimension Hierarchies window.
2. You may need to load the dimension hierarchy you want to edit.
3. Select the dimension hierarchy you want to edit.
4. Click **Edit**.
5. Modify the following details for the dimension hierarchy:
  - 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 “To map a base record table to an existing database table” on page 28.
8. After you finish mapping your 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**.

## To update a dimension hierarchy

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. Open the Dimension Hierarchies window.
2. You may need to load the dimension hierarchy you want to edit.
3. Select the table catalog containing the dimension hierarchies that you want to update.
4. Click **Update**.

## To remove a dimension hierarchy

**Important:** 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 (as those flowcharts contain a copy of the dimension hierarchy definition).

1. Open the Dimension Hierarchies window.
2. You may need to load the dimension hierarchy 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

Campaign allows you to define inbound and outbound triggers that can be used in all flowcharts in a partition.

**Note:** Triggers defined in Campaign flowcharts are not used with the IBM Scheduler. For information on using triggers with the IBM Scheduler, see the *Marketing 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?

Following are examples of events that might cause an inbound trigger to start a process in Campaign:

- 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 an acquisition campaign, which is waiting for the latest scores, to run.
- 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 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 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.

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.

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 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.

---

## Working with trigger folders and triggers

This section contains the following tasks:

- “To create folders to organize triggers”
- “To move a trigger folder”
- “To edit a trigger folder” on page 102
- “To delete a trigger folder” on page 102
- “To create a trigger” on page 102
- “To edit or move a trigger” on page 103
- “To delete a trigger” on page 104

## To create folders to organize triggers

**Note:** You must have permissions to create folders for triggers.

1. When editing a flowchart, select **Options > Stored Triggers**.
2. Click **New Folder**.
3. Enter a **Name** for the folder.
4. Optionally, enter a **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**.

## To move a trigger folder

**Note:** You must have permissions to move trigger folders.

1. When editing a flowchart, click the **Options** icon and select **Stored Triggers**.  
The Stored Trigger Definitions window opens.
2. In the left pane, select the folder you want to move.

3. Click **Edit/Move**. The Edit Folder window opens.
4. In the **Create Under** drop-down 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 sub-folder automatically inherits its security policy from its parent folder.
6. Click **Save**.

## To edit a trigger folder

**Note:** You must have permissions to edit trigger folders.

1. When editing a flowchart, click the **Options** icon and select **Stored Triggers**.  
The Stored Trigger Definitions window opens.
2. In the left pane, select the folder you want to edit.
3. Click **Edit/Move**.  
The Edit Folder window opens.
4. Edit the **Name** for the folder.
5. Edit the **Note**.
6. In the **Create Under** drop-down list, select the folder under which you want to move the selected folder, or select **None** to make the folder a top-level folder.
7. If you are editing a top-level folder, select a security policy.  
A sub-folder automatically inherits its security policy from its parent folder.
8. Click **Save**.

## To delete a trigger folder

**Note:** You must have permissions to delete trigger folders.

1. When editing a flowchart, click the **Options** icon and select **Stored Triggers**.  
The Stored Trigger Definitions window opens.
2. In the left pane, select the folder you want to delete.
3. Click **Remove**.  
You are prompted to confirm the deletion.
4. Click **OK**.

## To create a trigger

**Note:** You must have permissions to create triggers.

1. When editing a flowchart, select **Options > 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.
3. 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.

4. 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.

5. If you are creating a trigger in the top-level folder, select a security policy, or keep the default.
6. 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.
7. In the **Command** field, enter the path relative to the current partition root and file-name of the executable file on the Campaign server. You can click **Browse** to visually select the 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 (&).
8. Click **Save** to save the trigger.
9. Click **Close** to exit the Stored Trigger Definitions window.

## To edit or move a trigger

**Note:** You must have permissions to edit or move triggers.

1. When editing a flowchart, select **Options > Stored Triggers**.  
The Stored Trigger Definitions window opens, displaying all triggers defined in the current Campaign partition.
2. Locate and select the trigger to edit in the **Items List**.
3. Click **Edit/Move**.  
The data fields for the trigger appear on the right of the window.
4. Optionally, change the 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.

5. Optionally, modify 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.

**Important:** If you change the trigger name, any processes referring to that trigger will become unconfigured and will be unable to run. You will need to edit each process to refer to the new trigger name.

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 Campaign server. You can click **Browse** to visually select the 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** to save the trigger.
10. Click **Close** to exit the Stored Trigger Definitions window.

## To delete a trigger

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, select **Options > Stored Triggers**.  
The Stored Trigger Definitions window opens, displaying all triggers that are defined in the current Campaign partition.
2. Locate and select the trigger to delete in the **Items List**.
3. Click **Remove**.  
You are prompted to confirm the deletion.
4. Click **OK** to delete the trigger.
5. Click **Close**.

---

## Setting up outbound triggers

You must have permissions to use triggers in a flowchart.

### Setting up a process to execute an outbound trigger

You can have outbound triggers execute when one of the following processes runs:

- 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 *Campaign User's Guide*.

### To set up a flowchart to execute an outbound trigger on success

1. When editing 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 Success**.  
To use multiple triggers, enter the name of each trigger, separated by a comma and a space.
3. Click **OK**.  
The selected triggers execute when the flowchart runs successfully, on both production and test runs.

### To set up a flowchart to execute an outbound trigger on failure

1. When editing 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**.

The selected triggers execute when the flowchart encounters an error when running, on both production and test runs.

---

## Setting up inbound triggers

This section contains the following:

- “To set up inbound triggers”
- “Configuring the Schedule process to run with an inbound trigger”
- “To broadcast a trigger to all flowcharts of a campaign” on page 106
- “To broadcast a trigger to a flowchart” on page 106
- “To broadcast a trigger to all campaigns” on page 106

**Note:** 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 “To create a trigger” on page 102.
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.”
3. Use the Campaign Trigger Utility `unica_actrg` (in the folder `Campaign_home/bin`) to broadcast triggers, as described in:
  - “To broadcast a trigger to all flowcharts of a campaign” on page 106
  - “To broadcast a trigger to a flowchart” on page 106
  - “To broadcast a trigger to all campaigns” on page 106

### 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 *Campaign User’s Guide*.

## To broadcast a trigger to all flowcharts of a campaign

Run the 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.

## To broadcast a trigger to a flowchart

Run the Campaign Trigger Utility with the following syntax:

```
unica_actrg -n flowchart_name trigger_name
```

For example:

```
unica_actrg -n account_inquiry_flowchart web_hit
```

The trigger is broadcast only to all running flowcharts with the specified name. 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.

## To broadcast a trigger to all campaigns

Run the 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 for sending triggers to a Campaign installation on UNIX. Follow these steps to set up the `unica_actrg` utility and required files on a remote Windows machine:

1. Obtain the required files.

You can copy the files from another Campaign installation on Windows, or install Campaign to obtain the files.

For a list of required files, see “`unica_actrg` utility: required files” on page 107.

For information on installing Campaign, see the installation documentation.

If you run the installer to obtain the trigger utility files and you want to remove the unnecessary files, copy the files required for the trigger utility to another directory, then uninstall Campaign. For information on uninstalling Campaign, see the *IBM Campaign 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:\Unica\Campaign
```

When you run unica\_actrg.exe, specify the port and server name of the machine where the Campaign installation is located.

## unica\_actrg utility: required files

The following files are required for executing the Campaign trigger utility (unica\_actrg) on a remote Windows machine:

Table 32. Required files for unica\_actrg utility

Directory	File name
<CAMPAIGN_HOME>\bin	iconv.dll
	intl.dll
	libeay32.dll
	ssleay32.dll
	tls4d.dll
	unica_actrg.exe
	xerces-c_1_4.dll
<CAMPAIGN_HOME>\conf	config.xml

---

## Reference information for trigger administration

References in this section include:

- “Tokens supported by triggers”
- “Campaign trigger utility options” on page 109
- “Campaign trigger utility syntax” on page 109

## 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 33. Tokens supported by triggers

Token	Description	Where Used
<AMUSER>	The IBM EMM user name of the user running the flowchart.	Processes that support outbound triggers.

Table 33. 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>	Contact list specified in a contact process.  If the Contact List is written to a file, the appropriate full path name and file name replace the trigger token.  If the Contact List is written to a database table, the token is simply removed.	<b>Call List and Mail List</b> processes.
<CONTACTLOG>	The log of the particular contact process.  When the Log is written to a file, the appropriate full path name and file name replace the trigger token.	<b>Call List and 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 Distributed Marketing user.	Processes that support triggers, trigger on failure, trigger on success.
<OUTPUTTEMPTABLE>	A token for use in raw SQL in pre- and post-processing under the Advanced window to create a temp table. For example: Create <OUTPUTTEMPTABLE> as SELECT CustIDs from CustomerTable WHERE ...	<b>Select</b> process.
<OWNER>	The Marketing 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.
<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.

## Campaign trigger utility syntax

```
[-p <port>] [-s <server_name>] [-v] [<campaign_code> | -n  
"<flowchart_name>"] "<trigger1>" "<trigger2>"...
```

## Campaign trigger utility options

The unica\_actrg utility supports the following options.

Table 34. Campaign trigger utility options

Parameter	Use
-p <port>	The port that you want to use to run the utility.
-s <server_name>	The name of the Campaign server.
-v	Reports the version of the Campaign Trigger Utility.
<campaign_code>	The identifier of the campaign that contains all the 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 you want to use. You must specify at least one trigger. You can optionally specify multiple triggers, separated by spaces.



---

## Chapter 12. Logging administration

Campaign can provide the following types of logs:

- “The Campaign Listener log”
- “The Campaign web application log”
- “The Campaign Server Manager log” on page 112
- “The Campaign session utility log” on page 112
- “The cleanup utility log” on page 112
- “Flowchart logs” on page 112
- “The sessions log” on page 112
- “The web connections log” on page 112
- “Windows event logs” on page 112 (when the Campaign server is installed on Windows)
- “The log4j log file” on page 113

---

### The Campaign Listener log

The Campaign Listener log file contains events generated by the Campaign Listener.

The log is in the file `unica_aclsnr.log`, located in the *Campaign\_home/logs* directory.

Depending on the system's logging settings, the *Campaign\_home/logs* directory may contain multiple historical Campaign Listener logs, each ending with an extension number, for example `unica_aclsnr.log.1`, `unica_aclsnr.log.2`, and so on.

The number of logs retained and the maximum size of each log depend on the value of the `Applications>Campaign>unicaACLListener>logMaxBackupIndex` and the `Applications>Campaign>unicaACLListener>logMaxFileSize` properties, respectively.

---

### The Campaign web application log

The Campaign web log file contains events generated by the Campaign web application.

The log is in a file called `campaignweb.log`, located in the *Campaign\_home/logs* directory by default.

Depending on the system's logging settings, the *Campaign\_home/log* directory may contain multiple historical Campaign web application logs, each ending with an extension number, for example `campaignweb.log.1`, `campaignweb.log.2`, and so on.

You can configure Campaign web application logging properties in the `campaign_log4j.properties` file, located in *Campaign\_home/conf* directory by default.

---

## The Campaign Server Manager log

The 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 in the *Campaign\_home/logs* directory.

---

## The Campaign session utility log

The Campaign session utility log file (*unica\_acsesutil.log*) is generated if an error occurs when running the *unica\_acsesutil* utility. This log file is located in the *Campaign\_home/partitions/partition\_name/logs* directory.

---

## The cleanup utility log

The cleanup utility log file (*unica\_acclean.log*) is generated if an error occurs when running the *unica\_acclean* utility. This log file is located in the *Campaign\_home/partitions/partition\_name/logs* directory.

---

## Flowchart logs

Each campaign flowchart generates flowchart-specific logging information when it is run.

The log is in a file called *campaign\_name\_campaign\_code\_flowchart\_name.log*. By default, this log file is located in the *Campaign\_home/partitions/partition\_name/logs* directory. However, when you are editing a flowchart you can customize the log location by clicking the Options menu and selecting Change Log Path. Note that you cannot select Change Log Path unless AllowCustomLogPath is enabled in the Campaign properties.

---

## The sessions log

When a user views a flowchart, before editing it, session information for that flowchart is logged in the *ac\_sess.log* file, located in the *Campaign\_home/partitions/partition\_name/logs* directory.

The *ac\_sess.log* file records information about server connections when flowcharts are opened.

---

## The web connections log

When a user log into Campaign, information is logged in the *ac\_web.log* file, located in the *Campaign\_home/partitions/partition\_name/logs* directory.

The *ac\_web.log* file records information about the user's connections to the Campaign system database.

---

## Windows event logs

When Campaign is installed on a Windows computer, you have the option to log Campaign listener and flowchart events to the Windows event log. Whether the Windows event log is used is determined by the following configuration properties.

- Applications > Campaign > unicaACListener > enableWindows-EventLogging, for the Campaign listener.



- Applications > Campaign > partitions > *partition\_name* > server > logging > enableWindowsEventLogging, for flowcharts in that partition.

---

## The log4j log file

The Campaign web application uses the Apache log4j utility for logging configuration, debugging, and error information.

---

## Working with logs

To work with logs, see the following:

- “Campaign Listener logging tasks”
- “Campaign web application logging tasks”
- “Flowchart logging tasks” on page 114
- “Windows event logging tasks” on page 116

### Campaign Listener logging tasks

Campaign Listener logging tasks include:

- “To configure Campaign Listener logging”
- “To view the Campaign Listener log file”

#### To configure Campaign Listener logging

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

On the Configuration page, set the following properties in the Campaign > unicaACLlistener category as needed. For information on the properties, see the context help or the *Marketing Platform Administrator's Guide*.

- enableWindowsEventLogging
- loggingLevels
- logMaxBackupIndex
- logMaxFileSize
- windowsEventLoggingLevels

#### To view the Campaign Listener log file

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

The Campaign Settings page opens, presenting links for various administration tasks.

2. Click **View System Log**.

The current Campaign Listener log opens in a new browser window. Events that occur after you open the log file are not listed.

### Campaign web application logging tasks

Campaign web application logging tasks include:

- “To configure Campaign web application logging” on page 114
- “To change the file name and location of the Campaign web application log” on page 114

## To configure Campaign web application logging

1. Locate the file specified by the applications>Campaign>logging> property. By default, the file is *Campaign\_home/conf/campaign\_log4j.properties*.
2. Use the comments within this file to change the web application logging settings.
3. Save the file and restart the web application.

## To change the file name and location of the Campaign web application log

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

On the Configuration page, modify the value of the log4jConfig property in the Campaign > logging category to specify location and name for the file used to define the properties of the Campaign web application log.

## Flowchart logging tasks

Flowchart logging tasks include:

- “To configure flowchart logging”
- “To enable or disable flowchart logging” on page 115
- “To modify flowchart logging levels” on page 115
- “To customize the flowchart log file location” on page 115
- “To view flowchart log files” on page 115
- “To clear a flowchart log file” on page 116

## To configure flowchart logging

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

On the Configuration page, modify the following properties in the Campaign > partitions > partition[n] > server > logging category as needed. For information on the properties, see the context help or the *Marketing Platform Administrator's Guide*.

- allowCustomLogPath
- enableLogging
- enableWindowsEventLogging
- keepFlowchartLogOpen
- logFileBufferSize
- loggingCategories
- loggingLevels
- logMaxBackupIndex
- logMaxFileSize
- logProcessId
- windowsEventLoggingCategories
- windowsEventLoggingLevels

## To enable or disable flowchart logging

1. When editing a flowchart, click the **Options** icon.
2. In the **Options** menu, examine the **Enable Logging** command:
  - When this command shows a check mark, logging is enabled.
  - When no check mark appears, logging is disabled.
3. Select **Enable Logging** to toggle the current setting.

## To modify flowchart logging levels

1. When editing a flowchart, select **Options > Logging Options**.  
The **Logging Options** window opens.
2. Check the message types to log. In increasing order of verbosity, the four levels of logging are:
  - **Error** - Flowchart errors.
  - **Warning** - Flowchart warnings.
  - **Information** - Informational messages.
  - **Debug** - Verbose debugging information.Log files can grow quickly when informational or debugging messages are logged, so you may want to clear these options unless debugging.
3. Check the message categories to log.
4. Check **Include the Process ID in Log Entries** to have the process ID included for each entry.
5. Click **OK** to save the settings.

## To customize the flowchart log file location

Before you can customize the log file location, the `AllowCustomLogPath` configuration property for the Campaign server must be enabled in the Campaign > partitions > partition[n] > server > logging category. See the *Campaign Administrator's Guide* for details on setting configuration properties.

By default, each flowchart log file is stored in the `Campaign_home/partitions/partition_name/logs` directory. However, when you are editing a flowchart, you can specify that the flowchart's log should be stored in a different location.

1. When editing a flowchart, select **Tools > Change Log Path**.  
The Select Log Path window appears.  
If the **Change Log Path** option is not enabled, be sure that the server's `AllowCustomLogPath` configuration property is enabled, then edit the flowchart to try again.
2. Use the **Directory** list to locate the directory in which you want to store the flowchart log files. Double-click any directory name to see the directories it contains.
3. Optionally, you can create a new directory on the Campaign server by clicking the **New Folder** icon above the **Directory** list.
4. When you have selected the directory in the list, click **Open** to accept your choice. The Select Log Path window closes automatically.

The flowchart's log files will now be stored in the directory you specified.

## To view flowchart log files

When editing a flowchart, select **Options > View Logs**.

The flowchart log file opens in a new browser window.

## To clear a flowchart log file

When editing a flowchart, select **Options >Clear Logs**.

The content of the log file is deleted.

## Windows event logging tasks

Windows event logging tasks include:

- “To configure Windows event logging for the Campaign listener”
- “To configure Windows event logging for flowcharts”

### To configure Windows event logging for the Campaign listener

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

On the Configuration page, set the `enableWindowsEventLogging` and `windowsEventLoggingLevels` properties in the `applications>Campaign>unicaACLlistener` category as needed. For information on the properties, see the context help or the *Marketing Platform Administrator's Guide*.

### To configure Windows event logging for flowcharts

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

On the Configuration page, modify the following properties in the `Campaign > partitions > partition[n] > server > logging` category as needed:

- `enableWindowsEventLogging`
- `windowsEventLoggingCategories`
- `windowsEventLoggingLevels`

For information on the properties, see the context help or the *Marketing Platform Administrator's Guide*.

## log4j logging tasks

Using log4j logging in Campaign includes the following tasks:

- “To configure logging in log4j”
- “To change the location of the `campaign_log4j.properties` file” on page 117

### To configure logging in log4j

To configure logging in log4j for IBM Campaign, you set property values in the `campaign_log4j.properties` file located in the `conf` directory under your Campaign installation. For example, to change the location of the log file, open the `campaign_log4j.properties` file and change the value of the `log4j.appender.FILE.File` property to the fully qualified path where you want the log file to be written.

For information about changing log4j property settings, see the following sources:

- The comments in the `campaign_log4j.properties` file.
- The log4j documentation on the Apache web site:  
<http://logging.apache.org/log4j/1.2/manual.html>

**Note:** After you change values in the `campaign_log4j.properties` file, you must restart the IBM Campaign web application.

**To change the location of the `campaign_log4j.properties` file**

On the Configuration page, change the value of the `log4jConfig` property in the Campaign > logging category to the desired location.



---

## Chapter 13. Unique code administration

Each campaign, offer, treatment, and cell in Campaign has an identifying code that is generated by code generators, and conforms to a specified format.

As a Campaign administrator, you can:

- Set configuration parameters to control how each type of code is generated, and valid formats for codes.
- Create and set up a custom code generator if the default generators do not meet your needs.

---

### 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 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.

---

### About offer codes

An offer code is the globally unique identifier for an offer. Each offer in Campaign must have a code, and no two offer codes in the same Campaign partition should be the same.

An offer code can have one to five parts, which you specify when creating the offer template.

When users create an offer, the Offer Code field(s) 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.

---

### 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 *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 *Campaign User's Guide*.

---

## About treatment codes

The unique combination of a cell and an offer used at a particular point in time is referred to in Campaign as a treatment. Each treatment is uniquely identified with a treatment code.

For more information about treatments, see the *Campaign User's Guide*.

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:** Once treatment codes are generated, they cannot be overridden.

---

## Code formats

The default and valid format for each type of generated code uses a series of characters to represent the character types.

The following table lists the characters you can use to control code formats.

*Table 35. Control code formats*

Characters	Treated As
A-Z, any symbol, b, d-m, o-w, y, z (or, b-z except c, n, x)	A constant value in the generated code
a	Any uppercase letter A-Z



Table 35. Control code formats (continued)

Characters	Treated As
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 allowVariableLengthCodes property must be set to "TRUE."
n	Any number 0-9

## Example

Table 36. Example of control code format

Format definition	Example of generated code
CAMP_aaannn	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 IBM Campaign built-in code generators.

Table 37. Default code formats

Code type	Default Value	Where Defined
Campaign	Cnnnnnnnnn	campCodeFormat parameter on the Marketing Platform Configuration page
Cell	Annnnnnnnn	cellCodeFormat parameter on the Marketing Platform Configuration page
Offer	nnnnnnnnn	In each offer template defined in Campaign
Treatment	nnnnnnnnn	In each offer template defined in Campaign

## Code format requirements

Unique campaign, cell, treatment, and offer codes must be 32 characters or less. This applies to codes generated by both default and custom code generators as well as any manually entered codes.

In offer codes, you cannot use the space character.

## About changing the default code formats

You can override the default formats for the codes generated by the Campaign built-in code generators.

Before modifying the default code formats, be aware of code format restrictions in Campaign.

## To change the campaign code format

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

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.

On the Configuration page, set the `campCodeFormat` property in the Campaign > partitions > partition[n] > server > systemCodes category as required. Follow the formatting guidelines as described in “Code formats” on page 120.

## To change the cell code format

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

**Important:** Do not change the cell code format after users have created flowcharts. Doing so will invalidate existing flowcharts.

On the Configuration page, set the `cellCodeFormat` property in the Campaign > partitions > partition[n] > server > systemCodes category as required. Follow the formatting guidelines as described in “Code formats” on page 120.

## To change the offer or treatment code format in an existing offer template

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.

**Note:** 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.

1. Log in to 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 or treatment code format you want to change.
4. On offer template definition page, modify the Offer Code Format or Treatment Code Format as required, following the formatting requirements in Code format requirements.

**Important:** You cannot use the space character in an offer code format.

5. Click **Finish**.

---

## About code configuration properties

All properties for configuring campaign and cell codes, code generators, and certain attributes of offer codes in Campaign are set on the Marketing Platform Configuration page.

Offer code formats are defined in offer templates rather than configured using parameters.

---

## About code generators

Code generators are the programs used to automatically generate campaign, cell, offer, and treatment codes of the required format in Campaign.

In addition to its built-in code generators, Campaign supports custom code generators that you develop.

### Default code generators in Campaign

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 38. 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 Campaign is installed.

If the code generators built-in to Campaign do not meet your company's needs, you can develop and use custom code generators.

---

## About custom code generators

If the 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 Campaign web application is deployed.

**Important:** If the 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 the following 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 Marketing Platform Configuration page.

## About configuring Campaign to use custom code generators

You specify campaign and cell code formats and generators using a property on the Marketing Platform Configuration page.

**Note:** To complete this task, you must have the appropriate permissions in IBM EMM. For information, see the *Marketing 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

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

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.

### To specify the cell code generator

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

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.

### To specify the offer code generator

1. Log in to 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

1. Log in to 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 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 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 Campaign installation.

You can place custom offer code generators in a location of your choice, then specify the location using IBM EMM.

## 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 Campaign web application home.

**Note:** To complete this task, you must have appropriate permissions to use Marketing Platform. For information, see the *Marketing Platform Administrator's Guide*.

---

## Code generation references

This section contains the following reference topics:

- “Properties related to code generation”
- “Parameters for the default campaign and cell code generators” on page 127
- “Parameters for the default offer code generator” on page 127
- “Parameters for custom code generators” on page 128

## Properties related to code generation

Use the following properties to customize code formats and generators. To access and modify these properties, use the Marketing Platform Configuration page. For information about these properties, see the context help or the *Marketing Platform Administrator's Guide*.

Table 39. Properties to customize code formats and generators

Property	Path
allowVariableLengthCodes	Campaign> partitions> <i>partition[n]</i> > server>systemCodes>
campCodeFormat	Campaign> partitions> <i>partition[n]</i> > server>systemCodes>
campCodeGenProgFile	Campaign> partitions> <i>partition[n]</i> > server>systemCodes>
cellCodeFormat	Campaign> partitions> <i>partition[n]</i> > server>systemCodes>

Table 39. Properties to customize code formats and generators (continued)

Property	Path
cellCodeGenProgFile	Campaign> partitions> <i>partition[n]</i> > server>systemCodes>
displayOfferCodes	Campaign> partitions> <i>partition[n]</i> > server>systemCodes>
offerCodeDelimiter	Campaign> partitions> <i>partition[n]</i> > server>systemCodes>
allowDuplicateCellcodes	Campaign> partitions> <i>partition[n]</i> > server>flowchartConfig>
defaultGenerator	Campaign> partitions> <i>partition[n]</i> > offerCodeGenerator>
offerCodeGeneratorClass	Campaign> partitions> <i>partition[n]</i> > offerCodeGenerator>
offerCodeGeneratorClasspath	Campaign> partitions> <i>partition[n]</i> > offerCodeGenerator>
offerCodeGeneratorConfigString	Campaign> partitions> <i>partition[n]</i> > offerCodeGenerator>

## Parameters for the default campaign and cell code generators

The following parameters are supported by the uaccampcodegen program, located in the <Campaign\_home>/bin directory (where <Campaign\_home> is the Campaign installation directory, such as C:\Unica\Campaign\bin, or /Unica/Campaign/bin).

Table 40. 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 following parameters are supported by the uacoffercodegen program, located in the <Campaign\_home>/bin directory (where <Campaign\_home> is the Campaign installation directory).

Table 41. 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 "nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn"
-a 5 -u 3 -y 2008 -m 1 -d 14
```

## Parameters for custom code generators

Campaign supports custom parameters as input to any custom code generator you configure for use in Campaign.

No validation is performed for these parameters; however, the following restrictions apply:

- You cannot reuse the flags for the default 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 14. Advanced settings administration

You use advanced settings to manage some Campaign settings and server optimization features.

---

### To access Advanced settings

1. Open a flowchart in Edit mode.
2. Click the **Admin** icon and select **Advanced Settings**.

The Advanced Settings window opens.

The Advanced Settings window has three tabs:

- General
- Server Optimization
- Test Run Settings

---

### About the General settings

The General tab settings allow you to specify:

- Whether results from runs of this flowchart are saved;
- Whether flowchart processing is done "in database" or on the Campaign server;
- Whether global suppression is disabled for this flowchart;
- The Y2K (Year 2000) threshold;
- Whether Campaign automatically saves this flowchart;
- Whether Campaign uses checkpoints during runs of this flowchart;
- The maximum number of data errors Campaign allows;
- The triggers to send if a run of this flowchart results in an error;
- The triggers to send if a run of this flowchart completes successfully.

### Save flowchart run results

Select this checkbox to specify that all output cells from flowchart runs are saved when the run is finished. If you save results, 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.

By default, this checkbox is selected.

### Use in-DB optimization during flowchart run

Select this checkbox to specify that you want Campaign to operate in the database as much as possible when running flowcharts, rather than on the IBM server. This can improve performance because data does not have to be pulled from the database to the IBM server after each process runs. By default, this checkbox is cleared.

**Note:** In-database processing is not supported for all databases. Your Campaign administrator can confirm whether this option is available for your data sources.

For more details about in-database optimization, see the *Campaign User's Guide*.

## 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 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** value determines how 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 to prevent confusion.

Valid values are 0 to 100; any values higher than 100 are set to 100. The default setting for Y2K Threshold is 20.

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 Y2K 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 sent the Y2K 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

You can set the auto-save feature to automatically save your work periodically (for example, every 5 minutes) for recovery purposes. If the Campaign server terminates while you are editing a flowchart, if you had Auto-Save enabled, when you re-open the flowchart, you will see the last auto-saved version.

**Note:** You must have previously saved the current flowchart (provided a file name) for this feature to work.

Campaign stores auto-save files in a temporary directory so that the original flowchart files are not changed. Therefore, 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

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**, and 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, 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.

The default setting for Checkpoint is **Never**.

## Maximum data errors allowed

When 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 allows Campaign to continue working on the file (if less 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 want to write the errors to a log file.

## Send trigger on flowchart run errors

This option allows you to select one or more triggers from a list of outbound triggers that run when a campaign encounters errors during a flowchart run (indicated by a red X). You most commonly use this option to trigger an email to alert an administrator of the problem. Trigger on Failure runs for each process run that fails.

## Send trigger on flowchart success

This option allows you to select one or more triggers from a list of outbound triggers that run when a session succeeds. You most commonly use this option to trigger an email to alert an administrator of the successful run. Trigger on Success runs only if the entire flowchart run completes successfully.

---

## About the Server Optimization settings

The **Server Optimization** tab allows you to specify the Campaign **Virtual Memory Usage Limit** and override use of temporary tables for the current flowchart.

## Campaign Virtual Memory Usage

The **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** checkbox allows you to specify that temporary tables should not be used for the current flowchart.

This option overrides the `allow_temp_tables` property in the central configuration repository provided by IBM EMM for the user database. This setting does not affect the system data source.

---

## About the Test Run Settings

The **Test Run Settings** tab allows you to specify whether to write test run results to your database.

Select the **Enable Output** checkbox to specify that you want to output the results of your test runs to your database.

In general, Campaign does not write test run results to your database; however, you might want to verify that results are being recorded properly. To do this, limit your cell size and then select the **Enable Output** checkbox, so that you are using a limited amount of data to test your flowchart run and its output.

---

## Chapter 15. IBM Campaign utilities

This section describes the administrative utilities available with Campaign.

---

### Campaign Listener (`unica_aclsnr`)

The Campaign listener (`unica_aclsnr`) is a utility that allows clients to connect to the Campaign web application. The Campaign listener must be running, in addition to the web application server in which Campaign is deployed and running, before users who log in to the IBM EMM can work with any Campaign features.

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.exe`.

You can start and stop the listener manually or automatically.

To have the Campaign server start automatically with the system on which Campaign is running:

- If Campaign is installed on a Windows server, set up the listener as a service. See “To install the Campaign server as a Windows service” on page 135 for details.
- If 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.

### Campaign listener requirements

The Campaign listener requires that Marketing Platform be running.

The listener connects to Marketing Platform using the value of the `configurationServerBaseURL` property in the `config.xml` file, located in the `conf` directory of your Campaign installation. Typically, this value is `http://hostname:7001/Unica`. If Marketing Platform is not running, the Campaign listener will fail to start.

As the listener is dependent on Marketing Platform to start successfully, you should ensure that your web application server is running and the Marketing Platform web application is deployed before starting the listener.

### Campaign listener syntax

The `unica_aclsnr` utility has the following syntax:

```
unica_aclsnr [-i] {[-n] | [-r]} [-u] [-v]
```

## Campaign listener options

The unica\_acslnr utility supports the following options:

Table 42. Campaign listener options

Option	Description
-i	This option installs the listener utility as a service (Windows only).
-n	This option is the opposite of -r. It prevents the listener from checking the unica_acslnr.udb file.
-r (the default)	<p>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 you have flowcharts (that is, acsvr processes) still running. The listener stores flowchart information in a text file (unica_acslnr.udb). When you use the -r option, the listener checks the file for running flowcharts and re-establishes the connections.</p> <p>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.</p>
-u	This option uninstalls the listener utility as a service (Windows only).
-v	This option displays the current version of the listener.

## Starting and stopping the listener

If you have set up the listener as a service (on Windows) or as part of the init process (on UNIX) then the listener is automatically started when you start the server. You can also start and stop the listener manually, as described in this section.

### To start the Campaign listener on a Windows system

To start the Campaign listener on a supported Windows system:

1. Make sure that the web application server on which Campaign is deployed is up and running.
2. Start the Campaign listener by running the cmpServer.bat script located in the bin directory under your Campaign installation.

When the unica\_acslnr.exe process appears on the Windows Task Manager Processes tab, the server has started successfully.

### To stop the Campaign listener on a Windows system

To stop the Campaign listener on a supported Windows system:

1. Go to the Campaign bin directory and run the following command: svrstop -p 4664

If you are prompted for the CAMPAIGN\_HOME environment variable, set it as shown in this example, then run the svrstop command again:

```
set CAMPAIGN_HOME=C:\<installation_path>\Campaign
```

2. At the Login prompt, enter the user name of a Campaign user.
3. At the Password prompt, enter the password for the Campaign user you have entered.

The Campaign listener process closes. When the listener is not running, users who connect to IBM EMM cannot open any Campaign features.

## To install the Campaign server as a Windows service

To install the Campaign server as a Windows service that starts automatically whenever the Windows system starts:

1. Add the bin directory under the 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 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 Campaign server as a service.

2. If you are upgrading from a previous version of 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 Campaign installation.
4. Run the following command to create the Campaign server service:  
`unica_aclsnr -i`

The service is created.

**Note:** Make sure CAMPAIGN\_HOME has been created as a system environment variable before starting the Campaign server service.

## To start the listener on UNIX systems

Enter the following command at the system prompt:

```
rc.unica_ac start
```

## To stop the listener on UNIX systems

Enter the following command at the system prompt:

```
rc.unica_ac stop
```

## Campaign listener log

The listener process creates a log file called `unica_aclsnr.log`.

---

## Campaign Listener Shutdown utility (svrstop)

Use the Campaign listener shutdown utility (svrstop) to perform the following tasks:

- Shut down the Campaign listener
- Shut down the Contact Optimization listener

Note that the best practice is to start up and shut down the Contact Optimization listener using the `ACOServer` script, which uses the `svrstop` utility. See the *IBM Contact Optimization Installation Guide* for details.

**Note:** 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.

## Campaign svrstop utility reference

Use the svrstop utility to stop the Campaign listener or the Contact Optimization 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 Campaign server in the <install\_dir>/Campaign/bin directory, where <install\_dir> is the parent IBM directory in which Campaign is installed.

The svrstop utility uses the following syntax:

```
svrstop [-g] [-p <port> [-S]] [-s <serverName>] [-y <user>] [-z <password>]
[-v] [-P <product>]
```

Each argument is described in the following table:

Table 43. 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 <port> to 4664 to shut down the Campaign listener. Set <port> to 2882 to shut down the Optimize listener.
-S	Specifies that the listener specified by the -p or -P argument is using SSL.
-s <serverName>	The host name of the server on which the listener is running, such as optimizeServer or campaignServer.example.com. If you omit this argument, the utility attempts to shut down the specified listener on your local server.
-y <user>	The IBM EMM user with 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.
-z <password>	The password for the IBM EMM user 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 svrstop 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 Contact Optimization listener. Any other value for this argument, or omitting this argument, shuts down the Campaign listener.

```
svrstop -y asm_admin -z password -p 4664
```

## To use the svrstop utility to shut down the Campaign listener

From a command prompt on the Campaign server, you can run the svrstop utility to stop the Campaign listener running on that server. To stop the Campaign 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 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 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 Campaign uses internally to accept connections from the web client, so the `-p 4664` argument indicates you are stopping the Campaign listener.

4. When prompted, provide the name and password of any IBM EMM 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 Campaign listener is shut down.

## To use the `svrstop` utility to shut down the Contact Optimization listener

From a command prompt on the Campaign server, you can run the `svrstop` utility to stop the Contact Optimization listener running on that server. To stop the Contact Optimization 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 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 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 Contact Optimization listener.

4. When prompted, provide the name and password of any IBM EMM 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 Contact Optimization listener is shut down.

---

## Campaign Server Manager (`unica_svradm`)

The Campaign Server Manager (`unica_svradm`) is a command-line server management utility that allows you to perform the following tasks:

- Connect to the Campaign listener
- View all currently open flowcharts and their states
- View and set environment variables
- Run flowcharts
- Suspend/resume flowcharts
- Stop flowcharts
- Kill runaway flowcharts

When you start the `unica_svradm` utility, it checks whether the listener is running.

If the listener is running, the connection automatically establishes and the names of the server and the port number are displayed.

## To run the Campaign Server Manager

Before you can run the Campaign Server Manager:

- The listener must be running.
- The UNICA\_PLATFORM\_HOME and CAMPAIGN\_HOME environment variables must be set for the command window you are using.
- Your IBM EMM login must have the **Run Svradm cmdline** 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 described in “Campaign Server Manager commands.”

## Campaign Server Manager commands

The Campaign Server Manager supports the commands described in the following section. To view the list of all available commands for `unica_svradm`, use the `Help` command.

**Note:** Commands that take a flowchart name as an argument will operate on all flowcharts in all campaigns and sessions that have the same name. Use relative flowchart paths for commands that take flowchart paths.

The Campaign Server Manager commands are not case-sensitive.

### Cap (Distributed Marketing)

Cap

The Cap command prevents additional Distributed Marketing flowcharts from starting, while allowing those 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. You might use this command, for example, if you are deleting or disabling a user and want to re-assign ownership of that user's campaigns to a new user.

Option	Description
<code>-o &lt;olduserid&gt;</code>	User ID of the current owner of the campaign.
<code>-n &lt;newuserid&gt;</code>	User ID of the new owner you want to assign to the campaign.
<code>-p &lt;policyid&gt;</code>	Policy ID of the security policy to apply to the campaign.

### Connect

`Connect[-f] [-s server] [-p port][-S]]`

The Connect command connects to the listener running on the *server* at *port* number. You can connect to only one server at a time. To connect to another server, use -f (forced) connection.

If a port is specified using the -p option, you may also include the -S option to indicate that an SSL connection should be established. If a port is specified using the -p option but -S is not included, then the connection will not use SSL.

## Disconnect

Disconnect

The Disconnect command disconnects from the server. This command is available only if you are connected to a server.

**Note:** To connect to another server, you can use the -f parameter, or disconnect first, and then connect to the new server.

## Exit

Exit

The Exit command logs you out of the Campaign Server Manager.

## Help

Help

The Help command displays the available commands.

## Kill

Kill -p pid

The Kill command issues a "kill-p" to the *pid* specified (the Windows NT equivalent is issued on Windows NT). This is intended for run-away processes.

## Loglevel

Loglevel [high | low | medium | all]

The Loglevel command sets the listener logging level for Campaign or, if you enter the command without any arguments, displays the current logging level. If you modify the logging level, the change takes effect immediately, so there is no need to restart the listener after entering this command.

## Quit

Quit

The Quit command logs you out of the Campaign Server Manager.

## Resume

Resume {-s flowchart\_name | -p pid | -a}

The Resume command resumes the running of one or more flowcharts.

- Use -s to resume a single, specific flowchart by name
- Use -p to resume the specified process ID
- Use -a to resume all suspended flowcharts

## Run

Run -p relative-path-from-partition-root -u  
Unica\_Marketing\_Platform\_user\_name [-h partition] [-c catalogFile] [-s]  
[-m]

The Run command opens and runs a specific single flowchart file, where relative flowchart path and file name, partition, catalog file and user name are given.

You can use the following syntax:

[-S dataSource -U db\_User -P db\_Password]\*

**Note:** On Unix platforms, flowcharts will be executed by the Unix account specified as the username's alternate login. On Windows NT, the flowchart is run as the administrator's user login.

## Options for the Run command

Option	Description
-h	Specifies the partition name
-l	Indicates an alternative location in which to store the process log file. This option should be followed by a path relative to the Campaign installation, as in \partition1\logs. Do not specify a file name with this option, because the file name is assigned automatically. <b>Note:</b> Using this option also requires that the AllowCustomLogPath configuration property in the Campaign > partitions > partition[n] > server > logging category is enabled. See the <i>Marketing Platform Administrator's Guide</i> for details on setting configuration properties.
-m	Specifies that you are running multiple flowcharts. This option is not supported for batch flowcharts.
-p	Specifies the relative path from the partition root
-P	Specifies the data source password
-s	Specifies a synchronous run
-S	Specifies the data source
-u	Specifies the IBM EMM user name
-U	Specifies the data source user name
-v	Specifies the user variable values for a flowchart directly in the command, using this syntax:  [-v "varname=[']value[']"*]
-x	Specifies the user variable values for a flowchart in an XML file, using this syntax:  [-x xml-filename]

## Example of XML file for -x argument

This sample XML file sets the user variable named UVAcctType to the value Gold.

**Note:** 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.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
    <UserVariables>
        <UserVar Name="UVAcctType">
            <Values>
                <Option>Gold</Option>
            </Values>
        </UserVar>
    </UserVariables>
```

## Save

Save {-s flowchart\_name|-p pid|-a}

The Save command saves the current state of an active flowchart.

Option	Description
-s	Saves a single, specific flowchart by name defined by the <i>flowchart-name</i> .
-p	Saves a flowchart defined by the pid.
-a	Saves all running flowcharts.

## 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 displays, asking for confirmation of shutdown.

To override and force shutdown, use -f .

## Status

Status [-d |-i] [-u] [-v | -c]]

The Status command displays information both about running and suspended flowcharts (with flowchart name, owner, and file location).

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 the unica_acsvr processes before displaying output, which prevents crashed processes from being displayed in the Status list.
c	Verifies the existence of the unica_acsvr processes before displaying output, which prevents crashed processes from being displayed in the Status list. Option c also instructs the listener to clean up any temporary files in the partition temp 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}

The Stop command checks for active clients, warns if any are present (this can be overridden with the -f force option), and stops the IBM server processes.

Option	Description
-s	Stops a single, specific flowchart by name defined by the <i>flowchart_name</i>
-p	Stops flowcharts by <i>pid</i> .
-a	Stops all running flowcharts.

To override and force stop, use -f.

## Suspend

Suspend [-f] {-s flowchart\_name | -p pid | -a}

The Suspend command allows you to "quiesce" a running campaign and save the state for later restart with the matching command, Resume. All flowcharts currently running an output process complete data export activities and the flowchart is then saved as a paused flowchart. 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.

Option	Description
-s	Suspends a single, specific flowchart by name defined by the <i>&lt;flowchart_name&gt;</i>
-p	Suspends flowcharts specified by <i>pid</i>
-a	Suspends all running flowcharts

The system finishes running any currently running processes, and does not allow any subsequent processes to start. The flowchart is saved and written to the list of suspended flowcharts.

The -f parameter allows you to force a suspend. Once suspended, the flowchart is written to the listener as a suspended flowchart.

**Note:** If the flowchart is not running at the time of Suspend, it is saved, but not written to the listener and cannot be started with Resume.

## Uncap (Distributed Marketing)

### Uncap

The Uncap command revokes the Cap (Distributed Marketing) command. See Cap (Distributed Marketing).

## Version

### Version

The Version command displays the version of unica\_svradm and the version of the listener process to which you are connected. You can use this command to help troubleshoot a version mismatch error.

## To kill a running flowchart

You may need to kill a flowchart to immediately stop it. Because flowchart names can be the same in different campaigns and sessions, you should follow the instructions in this section.

1. At the command prompt, enter the following command to obtain a list of running flowcharts on the server:  

```
% unica_svradm status
```

Note that even if the flowchart names are the same, 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
```

When a flowchart is killed, its buffer is not flushed to disk. Instead, the last checkpoint copy is saved.

---

## Campaign session utility (unica\_acsesutil)

Use the 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.

**Note:** The unica\_acsesutil utility only supports importing and exporting objects between servers with the same version of Campaign installed.

## Environment variables required for unica\_acsesutil

You must set some environment variables to run unica\_acsesutil.

For all operating systems, you must set the following environment variables.

- UNICA\_PLATFORM\_HOME
- CAMPAIGN\_HOME

On UNIX only, set the database-specific library path appropriate to your UNIX platform, as follows.

- LIBPATH for AIX®

- SHLIB\_PATH for HP-UX
- LD\_LIBRARY\_PATH for Linux or Sun Solaris

## Campaign session utility use cases

Use the session utility (unica\_acsesutil) to perform the following tasks:

- “Exporting and importing objects between servers”
- “Backing up sessions” on page 146
- “Updating record counts and lists of distinct values” on page 146
- “Manipulating table catalogs” on page 146
- “Documenting catalog contents” on page 147

### Exporting and importing objects between servers

**Note:** The unica\_acsesutil utility only supports importing and exporting objects between servers with the same version of Campaign installed.

Use unica\_acsesutil to export and import campaigns, sessions, and flowcharts from one server to another, with the following limitations:

- Exporting campaigns or sessions exports only the associated system tables and metadata. Associated flowcharts must be exported separately; you can export flowcharts only one at a time.
- Before you can import a flowchart to a target system, its flowchart (.ses) file, as well as its associated campaign or session, must already exist on the target system. Therefore you should export and import all campaigns and sessions before exporting and importing their associated flowcharts, and you must manually copy the flowchart (.ses) files to the target server.
- Using the -s parameter to specify a Campaign flowchart (.ses) file on which to operate is required regardless of the object type you are exporting or importing. When exporting or importing a campaign or session with multiple associated flowcharts, any of the associated .ses files can be used.
- When you attempt to import eMessage or Distributed Marketing flowcharts, unica\_acsesutil checks to see that the relevant application is installed on the target system. If the required application is not installed on the target system, unica\_acsesutil generates an error and the selected objects are not imported.

**About the exporting and importing process:** The process of moving objects between servers using unica\_acsesutil is done in several stages, with some manual steps required:

1. Using the information in the flowchart (.ses) file specified with the -s option, the unica\_acsesutil utility writes exported objects and information to an intermediate output file, specified by the -e option.
2. You manually move (copy) the output file to the target server.
3. On the target server, you import the unica\_acsesutil output file using the -i option.

### About conflicts with existing data when importing:

When unica\_acsesutil imports data into the system tables (for example, session info, triggers, or custom macros), it checks whether each object already exists on the target system.

The check is done based on internal object ID. That is, for campaigns, 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 a conflicting object (such as a campaign, session, or offer) already exists on the target system prior to import, you should consider deleting the object before running the import, to avoid being prompted to resolve the conflict.

#### To export a campaign, session, or flowchart:

```
unica_acsesutil -s <sesFileName> -h <partitionName>
-e <exportFileName> [-f { flowchart | campaign | session }]
[-S <datasource> -U <DBusername> -P <DBpassword>]
```

#### Example 1: export campaign

```
unica_acsesutil -s "campaigns/Campaign C000001_C000001.ses" -h partition1
-e campaign.exp -f campaign
```

This example 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: export flowchart

```
unica_acsesutil -s "campaigns/Campaign C000001_C000001_Flowchart1.ses"
-h partition1 -e flowchart.exp -f flowchart
```

This example 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`.

#### To import a campaign, session, or flowchart:

**Note:** Before you can import a flowchart, its flowchart (`.ses`) file, as well as its associated campaign or session, must exist on the target system. Therefore, before importing a flowchart, you must: 1) manually copy the associated `.ses` file from the source system to the target system, and 2) import the associated campaign or session to the target system.

```
unica_acsesutil -s <sesFileName> -h <partitionName>
-i <importFileName> [-f { flowchart | campaign | session }]
[-b { abort | replace | skip }]
[-S <datasource> -U <DBusername> -P <DBpassword>]
```

#### Example 1: import campaign

```
unica_acsesutil -s "campaigns/Campaign C000001_C000001.ses" -h partition1
-i campaign.exp -f campaign
```

This example 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: import flowchart

```
unica_acsesutil -s "campaigns/Campaign C000001_C000001_Flowchart1.ses" -h partition1 -i import.exp -f flowchart
```

This example 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 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 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 -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 Campaign session utility to manipulate table catalogs outside of 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 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 two techniques to document catalog contents:

- Use an XML catalog file to generate a report
- Print table mappings using the Campaign session utility

## 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 IBM 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
```

## Campaign session utility return values

The `unica_acsesutil` utility returns a value of 0 if it runs successfully. It returns a 1 if no files are found with the specified flowchart or catalog file name.

## Campaign session utility syntax

```
unica_acsesutil -s sesFileName -h partitionName
[-r | -c | -x [-o outputFileName]] [-u] [-v]
[{-e exportFileName [-f {flowchart | campaign | session}}]
| {-i importFileName [-t catFileName]}
[-b {abort | replace | skip}}]
[-p] [-a | -n | -l]
[-S dataSource -U DBUser -P DBPassword]*
[-y userName] [-z password]
[-j owner] [-K policy]
```

## Campaign session utility options

The `unica_acsesutil` utility supports the following options.

Table 44. Campaign session utility options

Option	Syntax	Description
-a	-a	Recomputes record counts and the list of distinct values for all tables.
-b	-b {abort   replace   skip}	<p>Applies only for the import option (-i). Specifies that the import be done in batch mode.</p> <p>Requires one of the following arguments to specify how to handle duplicate objects (if there is an ID conflict):</p> <ul style="list-style-type: none"> <li>• abort - If a duplicate object is detected, the import stops.</li> <li>• replace - If a duplicate object is detected, replace it with the imported object.</li> <li>• skip - If a duplicate object is detected, do not replace it and continue the import.</li> </ul>
-c	-c <outputFileName>	Generates a table catalog in <i>outputFileName</i> in .cat format (Campaign internal format). This option is ignored with the -s option.
-e	-e <exportFileName>	<p>Exports the object type specified by the -f option to a file named <i>exportFileName</i>.</p> <p>If the -f option is not used, by default a flowchart is set for export.</p>
-f	-f {flowchart   campaign   session}	<p>Specifies the type of object to export. If this option is omitted, by default a flowchart is set for export.</p> <p>If -f is used, requires one of the following arguments: flowchart, campaign, session.</p>
-h	-h <partitionName>	Specifies the name of the partition in which the flowchart file (specified with -s) is located. This parameter is required.
-i	-i <importFileName>	Specifies the name of the file being imported. This should be a file that was exported using the -e option in a previous export operation.
-j	-j <owner>	Specifies the owner of the file being imported or exported.
-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 <i>outputFileName</i> . If unspecified, the default is <i>catFileName.xml</i> or <i>catFileName.cat</i> , depending on whether you use the -x or -c 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 -U and -S options.
-p	-p	Print table mappings to the console.

Table 44. Campaign session utility options (continued)

Option	Syntax	Description
-r	-r <outputFileName>	Generate a flowchart XML report in <i>outputFileName</i> . This parameter is ignored when you use the -t option (using a table catalog as input).
-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 Campaign flowchart (.ses) file on which to operate. Specifying an .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> <p>"campaign/Campaign C00001_C00001_Flowchart 1.ses"</p> <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 <i>outputFileName</i> . If the input table catalog is a .cat file, it generates a corresponding .xml file, and vice versa.
-y	-y <userName>	Specifies the IBM EMM user name.
-z	-z <password>	Specifies the password for the IBM EMM user specified by the -y option.

## 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 both the Campaign system tables database and on user tables databases.

**Note:** When you run the unica\_acclean utility, you must stop all the flowcharts that are currently running or the flowcharts that are scheduled to run.

Users running this utility must have the "Perform Cleanup Operations" permission, granted by their Campaign administrator. If users attempt to run this utility without the appropriate privileges, the tool terminates after displaying an error.

**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.

The following items can be identified and cleaned up by the utility:

- 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 have been 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.

## Campaign cleanup utility use cases

Use the cleanup utility (unica\_acclean) to perform the following tasks:

- “Generating a list of orphaned files and tables”
- “Deleting the files and tables listed in a file” on page 151
- “Deleting all orphaned temp files and tables” on page 151
- “Generating a list of files and tables selectively by object type and criteria” on page 152
- “Deleting files and tables selectively by object type and criteria” on page 152

### 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:** IBM 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 determines whether files and tables are orphaned in the following way:

### 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 Marketing 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 Marketing 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 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.

## Campaign cleanup utility syntax

```
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>]
[-S <dataSource> -U <DB-user> -P <DB-password>]*
```

The cleanup utility is non-interactive unless user name or password are not specified. If user name is not specified, the tool prompts for user name and password. If password is not specified, the tool prompts for password.

## Campaign cleanup utility options

The unica\_acclean utility supports the following options.

Table 45. Campaign cleanup utility options

Option	Syntax	Description
-d	-d	Deletes temp tables and files. All flowchart files are scanned, and based on the result, 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 unica_acclean.log. You can modify the log file name, but specifying a different location is not currently supported.
-h	-h	Displays usage help. Any invalid command-line invocation also displays the help.
-i	-i <clean file name>	Specifies the file listing the items to be deleted. Best practice is to use the same file generated by the cleanup tool, using the -o option.
-l	-l {low   medium   high   all}[-f <logFileName>]	Specifies logging level and the log file name. If no level is specified, medium 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.
-o	-o <listfilename>	Outputs the list of tables and files to the specified file, but does not delete them.
-P	-p	Print table mappings to the console.
-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.

Table 45. Campaign cleanup utility options (continued)

Option	Syntax	Description
-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, the unica_acclean tool 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>Specifies the name of the data source for the object being operated on. Use with the -U &lt;database_user&gt; and -P &lt;database_password&gt; options. These options allow you to override the credentials stored in Marketing Platform or provide authentication for data sources whose ASMSaveDBAuthentication is set to FALSE.</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 specifying a campaign folder or session folder as the object, criteria are based on columns in the UA_Folder table.</li> <li>• When specifying a campaign as the object, criteria are based on columns in the UA_Campaign table.</li> <li>• When specifying a flowchart as the object, criteria are based on columns in the UA_Flowchart table.</li> <li>• When specifying a session as the object, criteria are based on columns in the UA_Session table.</li> </ul>
-U	-U <DBUserName>	<p>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.</p>
-u	-u <user name>	<p>Must be used when the -p option is used. Use this option to supply the IBM EMM user name of the user running the utility.</p>
-v	-v	<p>Displays version and copyright information for the cleanup utility.</p>

Table 45. Campaign cleanup utility options (continued)

Option	Syntax	Description
-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". For details, see -s.</p> <p>Optionally use the -r option to recursively search subfolders. For details, see -r.</p>

## Campaign report generation utility (unica\_acgenrpt)

unica\_acgenrpt is a command-line report generation utility that 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

For more information about these reports, see the *IBM 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 Chapter 2, "Managing security in IBM Campaign," on page 3.

### 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.

## IBM Campaign report generation utility syntax

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]
```

## IBM Campaign report generation utility options

The unica\_acgenrpt utility supports the following options.

*Table 46. Campaign report generation utility options*

Option	Syntax	Description
-s	-s <sesFileName>	Specifies a Campaign flowchart (.ses) 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 -h option). For example, a valid value for -s is:  "campaign/Campaign C00001_C00001_Flowchart 1.ses"  The <sesFileName> can contain wildcard characters to operate on multiple matching flowcharts.
-h	-h <partitionName>	Specifies the name of the partition in which the flowchart file (specified with -s) is located.
-r	-r <reportType>	Specifies the type of report to be generated. Valid values include: <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 <name>=<value>	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 "Parameters used with the unica_acgenrpt -p option" on page 157.
-d	-d <delimiter>	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.

Table 46. Campaign report generation utility options (continued)

Option	Syntax	Description
-o	-o <outputFileName>	Specifies the output file name. The default is <sesFileName> with .ses replaced by .csv; specifies a destination directory when using wildcards.
-y	-y <user>	Specifies a login user name for Campaign.
-z	-z <password>	Specifies the password for the user login.
-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 the following 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
cell	Required	Name of a cell you want to profile.
field	Required	Name of the field you want to use to profile the cell.
cell2	Optional	Name of an additional cell to profile.
bins	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.
meta	Optional	Specifies whether you want to profile by meta type. Valid values are TRUE and FALSE. The default is TRUE.

#### Cell Variable Crosstab report

Parameter name	Usage	Description
cell	Required	Name of a cell you want to profile.
field1	Required	Name of the first field you want to use to profile the cell.
field2	Required	Name of the second field you want to use to profile the cell.
cell2	Optional	Name of an additional cell to profile.
bins	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.
meta	Optional	Specifies whether you want to profile by meta type. Valid values are TRUE and FALSE. The default is TRUE.

## Cell Content report

Parameter name	Usage	Description
cell	Required	Name of the cell you want to include in the report.
field	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).
records	Optional	Number of records you want to include in the report. The default is 100.
skipdups	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 TRUE and FALSE. The default is FALSE.

---

## Database testing utilities

Campaign supports the following command-line database testing utilities, which you can use to test connections to a target database, to execute queries and to perform a variety of tasks:

- “cxntest utility”
- “odbctest utility” on page 159
- “db2test utility” on page 160
- “oratest utility” on page 161

These utilities are in the /Campaign/bin directory on the Campaign server.

**Note:** If your operating system does not provide the db2test utility, use the cxntest utility to test connections to the target database.

### cxntest utility

The cxntest utility allows you to test connections to a target database and, once connected, to issue a variety of commands.

#### To use the cxntest utility

1. From a command prompt on the Campaign server, run the cxntest utility.
2. The cxntest utility is prompt-driven. You must enter the following information at the prompts:
  - a. The name of the connection library for your database
  - b. The name of the data source
  - c. A database user ID
  - d. The password associated with the database user IDThe 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]`

Performs an array fetch on a list of tables, 500 at a time. Can optionally specify a search *pattern*.

- *describetable*  
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. 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.

## odbcetest utility

The odbcetest utility allows you to test Open DataBase Connectivity (ODBC) connections to a target database and, once connected, to issue a variety of commands. It is supported on AIX, Solaris, Windows and HP-UX systems (32-bit only).

**Note:** For Oracle and DB2® databases, use their native utilities.

### To use the odbcetest utility

1. From a command prompt on the Campaign server, run the odbcetest utility.  
The odbcetest 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. The odbcetest utility is prompt-driven. You must enter the following information exactly at the prompts:
  - a. The name of the database 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 odbcetest 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.
  - *descriesSQL\_command*

- Describes the columns returned by the SQL command specified by *SQL\_command*.
- *describepattern*  
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.

## 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.

### To use the db2test utility

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 Campaign server, run the db2test utility.  
The db2test utility returns a list of databases (registered data sources) that you can connect to.
2. The db2test utility is prompt-driven. You must 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 db2test 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:
  - *describepattern*  
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.

## oratest utility

The `oratest` utility allows you to test connections to an Oracle server.

### To use the oratest utility

1. From a command prompt on the Campaign server, run the `oratest` utility.
2. The `oratest` utility is prompt driven. You must enter the following information exactly at the prompts:
  - a. The name of the Oracle server 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 `oratest` utility prints a **Connection Successful** message and exits with a return value of zero (0).

---

## Database load utilities

Campaign offers support for using database load utilities to increase performance for pushing ID lists into temporary tables, and for exporting data back into the database. This functionality works with most common database load utilities, which are available directly from the database vendors. It is your responsibility to obtain licensed copies of these utilities.

The Campaign load support is controlled through a series of properties defined on the Marketing Platform Configuration page. For more information, see the *Marketing Platform Administrator's Guide*.

When you use most database load utilities, you must also specify a control file. Campaign can generate this file dynamically based on a control file template that you configure. You only need to configure these files once; no changes are required from the user interface.

When Campaign needs to populate data to the database (for example, for a Snapshot process or a contact process such as MailList, or an ID list into temporary tables), it does the following:

1. Creates a temporary data file as fixed width or delimited text.

If specified by the LoaderControlFileTemplate property, a temporary control file is dynamically created based on the template file and the list of fields that need to be sent to the database.

2. Issues the command specified by the LoaderCommand property. This can be either a direct call to the database load utility executable or a call to a script that launches the database load utility.
3. Cleans up the temporary data file and control file.

This functionality allows you to load data into a new or empty database table and append data to an existing database table.

**Note:** Campaign does not support using load utilities to update records in an existing database table.

## 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 47. Fast loader repeated tokens*

Token	Description
<CONTROLFILE>	This token is replaced with the full path and file name to the temporary control file that Campaign generates according to the template that is specified in the LoaderControlFileTemplate parameter.
<DSN>	This token is replaced with the value of the DSN property. If the DSN 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 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 Campaign during the loading process. This file is in the 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 Campaign is loading data into. This is the target table from your Snapshot process or the name of the Temp Table being created by 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.



---

## Chapter 16. IBM Digital Analytics and Campaign integration

When IBM Digital Analytics and Campaign are integrated, online segments and associated data from Digital Analytics can be combined with offline profile data in Campaign.

You can select segments defined in Digital Analytics and target them in marketing campaigns, based on web activity and behavior.

Using an integrated system provides the following benefits:

- Web analysts can quickly follow up on trends noticed online, by defining segments to be targeted by Campaign.
- Campaign managers can align their campaign tactics with marketers' demands.
- Business marketers can track and follow up on campaign tactics, by measuring success and ROI of cross-channel campaigns.
- If the optional eMessage and post-click analytics tools are configured, marketing analysts can track customer and prospect behavior after targeting them with email campaigns.

---

### Integrating IBM Digital Analytics and Campaign

When Digital Analytics is integrated with Campaign, online segments defined in Digital Analytics can be used in campaigns.

**Note:** This topic pertains specifically to IBM Digital Analytics, not IBM Digital Analytics for On Premises.

The integration between Digital Analytics and Campaign relies on several components:

- An integration service that provides access to Digital Analytics APIs and acts as the integration point between the two products.
- A translation table that tells Campaign which Digital Analytics keys correspond to which Campaign Audience IDs.
- A Marketing Platform user account configured with the necessary credentials for Campaign to access the integration service.
- Configuration settings that inform Campaign about the integration service, translation table, and credentials.

The following table explains how to configure the necessary components.

*Table 48. Integrating Digital Analytics and Campaign*

Task	Details	For documentation
Optionally, configure SSO so users can easily navigate between products.	Single sign-on (SSO) lets users access Digital Analytics from within the IBM EMM user interface without being prompted to log in.	<i>IBM Marketing Platform Administrator's Guide</i>

Table 48. Integrating Digital Analytics and Campaign (continued)

Task	Details	For documentation
Configure a translation table to translate Digital Analytics keys to Campaign Audience IDs.	<p>This task is typically performed by IT or other technical personnel.</p> <p>The translation table consists of at least two columns, one for the Digital Analytics registrationid (online key) and one or more for each Campaign Audience ID (offline key). The translation table must be configured on the user data source from which Campaign selections take place.</p> <p>Note the table name, because you need to specify it in the Campaign configuration settings.</p>	"About the translation table" on page 168
Configure the segment integration	<p>For each Campaign partition where you want to enable integration, choose <b>Settings &gt; Configuration &gt; Campaign   partitions   partition[n]   Coremetrics</b> and configure these settings:</p> <ul style="list-style-type: none"> <li>• <b>ServiceURL</b>: Identifies the integration service (<a href="https://export.coremetrics.com/eb/segmentapi/1.0/api.do">https://export.coremetrics.com/eb/segmentapi/1.0/api.do</a>).</li> <li>• <b>CoremetricsKey</b>: Identifies the value used in the translation table (registrationid).</li> <li>• <b>ClientID</b>: The unique Digital Analytics ID assigned to your company.</li> <li>• <b>TranslationTableName</b>: The name of the translation table.</li> <li>• <b>ASMUserForCredentials</b>: The Marketing Platform account allowed to access the integration service. The default is asm_admin.</li> <li>• <b>ASMDatasourceForCredentials</b>: The data source assigned to the Marketing Platform account identified in the <b>ASMUserForCredentials</b> setting. The default is UC_CM_ACCESS.</li> </ul>	"Campaign   partitions   partition[n]   Coremetrics" on page 317
Assign credentials to a Marketing Platform account	<p>Choose <b>Settings &gt; Users</b>, select the user that is defined in the <b>ASMUserForCredentials</b> configuration setting, click the <b>Edit Data Sources</b> link, and add a new data source:</p> <ul style="list-style-type: none"> <li>• The Data Source name must exactly match the <b>ASMDatasourceForCredentials</b> defined in the configuration settings (for example, UC_CM_ACCESS).</li> <li>• The Data Source Login and Password are the credentials associated with your Digital Analytics Client ID.</li> </ul> <p>This "data source" is the mechanism that Marketing Platform uses to store the credentials that provide access to the integration service.</p>	<i>IBM Marketing Platform Administrator's Guide</i>

Table 48. Integrating Digital Analytics and Campaign (continued)

Task	Details	For documentation
Map the translation table.	<p>Mapping a table is how you make Digital Analytics data accessible to Campaign.</p> <p>Choose <b>Settings &gt; Campaign Settings &gt; Manage Table Mappings</b>, then follow the prompts to specify the data source corresponding to the user database where the user tables reside (<i>not</i> the "data source" you defined for ASMDatasourceForCredentials).</p> <p>Select the translation table, the table fields, and audience levels.</p> <p>Store the table mapping in the default catalog (default.cat) if you want it to be available for use in all flowcharts.</p>	"Mapping the translation table" on page 170
Specify which Campaign users can use Digital Analytics segments in flowcharts.	<p>Choose <b>Settings &gt; User Roles &amp; Permissions &gt; Campaign &gt; Partition[n] &gt; Global Policy</b>. Click <b>Add Roles and Assign Permissions</b>, then <b>Save and Edit Permissions</b>. Under <b>Campaigns</b>, adjust access for <b>Access Coremetrics Segments</b>.</p> <p>By default, access is "Granted" for the Folder Owner, Owner, and Admin roles, and "Not Granted" for the Execute, Design, and Review roles.</p> <p><b>Note:</b> When determining access, SSO is not considered. If you are using single sign-on and you want Campaign users to be able to access Digital Analytics segments, you still have to provide segment access by setting Global Policy.</p>	"Implementing security policies" on page 10
Turn on the integration for individual Campaign partitions.	Choose <b>Settings &gt; Configuration &gt; Campaign   partitions   partition[n]   server   internal   UC_CM_integration</b> . When this option is set to Yes, the Select process box provides the option to use <b>Digital Analytics Segments</b> as Input.	"Campaign   partitions   partition[n]   server   internal" on page 313
Configure permissions in your Digital Analytics product(s).	If the Campaign-related permissions are not set, <b>Digital Analytics Segments</b> are not available as input in a Select process box in a Campaign flowchart.	The Digital Analytics product documentation.
Define segments in Digital Analytics and make them available to Campaign.	<p>Use one of the following products to export segments to Campaign:</p> <ul style="list-style-type: none"> <li>Digital Analytics</li> <li>Explore</li> <li>Export</li> </ul>	The Digital Analytics product documentation.

Table 48. Integrating Digital Analytics and Campaign (continued)

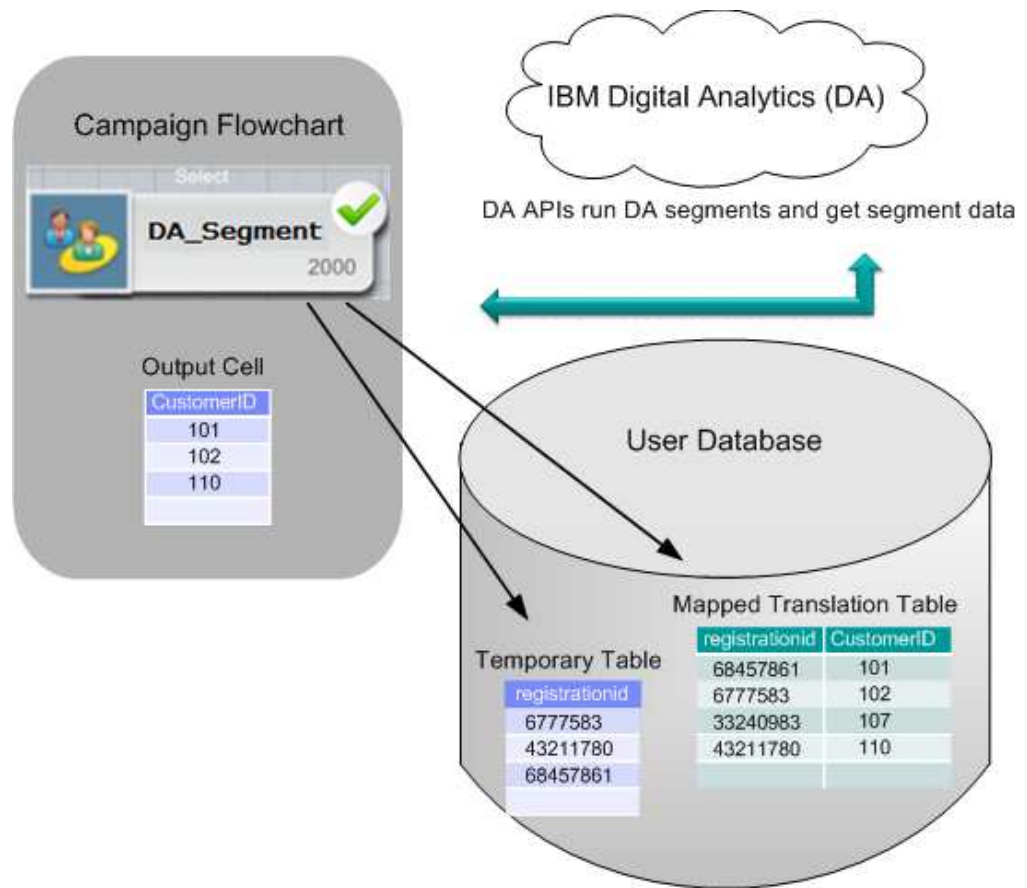
Task	Details	For documentation
Now Campaign users can start using Digital Analytics segments in flowcharts.	<p>Add a Select process box to a flowchart and choose <b>Digital Analytics Segments</b> as the <b>Input</b>. Select a <b>Client ID</b> and a segment, specify a date range, then run the flowchart.</p> <p>When the Select process runs:</p> <ul style="list-style-type: none"> <li>• It pulls data from Digital Analytics via the integration service. The segment data is simply a list of registration IDs.</li> <li>• Using the mapped translation table, the registration IDs are converted to Campaign Audience IDs.</li> <li>• The Audience IDs are then available for use in downstream processes.</li> </ul>	<i>IBM Campaign User's Guide</i>

## About the translation table

A translation table is typically configured by IT or other technical personnel when the Digital Analytics-Campaign integration is initially configured.

The translation table tells Campaign which Digital Analytics IDs correspond to which Campaign Audience IDs. A translation table is required when running a flowchart that includes Digital Analytics segments as input. Without the table, Campaign does not know how to translate IDs from one product to the other.





A translation table must adhere to the following guidelines:

- The translation table must be configured on the user data source from which Campaign selections take place. This data source must allow table creation permission so Campaign can create a temporary table for the list of IDs that meet the segment definition.
- A translation table consists of at least two columns, one for the Digital Analytics key and one or more for the corresponding Campaign Audience ID.
- For this release, the Digital Analytics key must be the exact value: `registrationid`.
- The data type of the `registrationid` in the translation table must be the same data type defined for the `registrationID` in Digital Analytics. For example, they must both be `VARCHAR`.
- The Campaign Audience ID is defined by the customer when the integration is configured. For example, it might be `CustomerID`.
- If your primary Campaign audience consists of multiple physical keys (compound keys), the translation table must contain the same number of columns for that audience. For example, if your primary audience consists of the keys `CustomerID` and `AccountID`, then the translation table must have three columns: 1) The Digital Analytics key 2) `CustomerID` 3) `AccountID`. This requirement pertains only if you are mapping for a compound audience.

**Note:** For performance and storage reasons, the best practice is to use single-key audiences.

- You must ensure that the translation table name and the Digital Analytics key (`registrationid`) defined in the table are specified in the Campaign

configuration settings. The values used in the translation table must exactly match the values defined in the configuration settings. See “Campaign | partitions | partition[n] | Coremetrics” on page 317.

- The method for populating a translation table depends on each customer's specific requirements and configuration. A common logic needs to be determined that allows for identifying which Digital Analytics registration ID matches which Campaign Audience ID. Your implementation partner can assist you in this process.
- The translation table must be mapped in Campaign. See “Mapping the translation table.”
- When a flowchart runs, Campaign detects if there is a mismatch between the number of Digital Analytics keys and Campaign Audience IDs in the mapped translation table. This situation can occur, for example, if the ETL routines are still running). In this case, Campaign writes a message to the flowchart log file asking you to verify that the mapped translation table contains updated records. To resolve this situation, (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.

---

## Mapping the translation table

Mapping the translation table is how you make Digital Analytics segment data accessible in Campaign. The table mapping identifies the data source, the translation table name and location, the table fields, audience levels, and data for Campaign to use.

Before mapping the table, you must configure the translation table, configure integration settings, and assign credentials to a Marketing Platform account. For instructions, see “Integrating IBM Digital Analytics and Campaign” on page 165.

This task is typically performed by IT or other technical personnel when the Digital Analytics-Campaign integration is initially configured. However, tables can be mapped or remapped by any user with appropriate permissions, at any time.

1. Confirm that Campaign is configured properly to access the necessary user tables: See “To test user table access” on page 22.
2. Map the translation table by following the instructions and guidelines in “Working with user tables” on page 26.

To summarize: You map a new base record table to make data accessible to processes in flowcharts. You can map (create) a new base record table when editing a flowchart (using **Admin > Tables**) or by selecting **Settings > Campaign Settings > Manage Table Mappings**.

3. Save the mapping information in a table catalog for reuse. This way, the table mapping needs to be done only once, and Campaign users who want to include Digital Analytics segments can load the saved catalog to get the mapping information directly. If you want the information to be available for use in all flowcharts, store it in the default catalog (default.cat). See “Working with table catalogs” on page 39.
4. Whenever 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 using 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 **IBM Campaign Table Name** assigned in the Table Definition wizard must exactly match the

TranslationTableName defined in the Campaign configuration settings (see “Campaign | partitions | partition[n] | Coremetrics” on page 317). If you do not edit the table name when using the Table Definition wizard, the names will match.



---

## Chapter 17. Configuring Campaign for non-ASCII data

Campaign and PredictiveInsight support the use of localized data and non-United States locales, including multiple user-preferred locales within the same installation of the IBM 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. IBM strongly recommends that you avoid using your IBM 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 IBM EMMapplication. This section includes the following:

- “About character encodings.”
- “About interactions with non-ASCII databases.”
- “About the multi-locale feature” on page 174.

#### About character encodings

A character encoding is the means by which human language is represented on a computer. Many different encodings are used to represent different languages. In order to configure your IBM 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. Special cases in character encoding occur in some text formats. For more details, see “The encoding of text in character-based fields” on page 174.

The supported encodings are listed in “Character encodings in Campaign” on page 335.

#### About interactions with non-ASCII databases

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

Campaign and PredictiveInsight communicate directly with the database client and the client communicates with the database. Each database varies in how it handles language-sensitive data. You must understand the encoding and date format that your database server and client use, and ensure that you configure your IBM application correctly for these settings.

#### The format of date and time fields

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 340. For more details about the multi-locale feature, see “About the multi-locale feature.”

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 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 might specify the encoding that will be used throughout the database. Campaign and PredictiveInsight can be configured to use one of many different character encodings on a database-wide scale. Per-column encodings are not supported.

One feature that is common in modern databases is that the database client will transcode between the encoding of the database and the encoding that an application uses. This is common in the case where an application uses some form of Unicode, but the database uses a language-specific encoding.

## About the multi-locale feature

Campaign supports multiple languages and locales for a single installation. 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 IBM EMM.

Setting a user's locale preference is optional. Unless a preferred locale is set explicitly in IBM EMM for a user, there is no "preferred" locale at the user level, and when that user logs in, Campaign uses the suite-level locale set in IBM EMM.

When a preferred locale is explicitly set for a user, this preference overrides the suite-level setting, and when this user logs in to 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 Campaign and work simultaneously, each in their preferred language and locale. For details about setting user locale preferences in IBM EMM, see the *IBM Marketing Platform Administrator's Guide*.

To configure your system for multi-locale functionality, see “Configuring Campaign for multiple locales” on page 181. You complete the tasks in that section after configuring 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 Campaign. The following areas are not affected by user locale preferences:

- Parts of the 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 Campaign in any language.
- Campaign command line tools -- these display in the default language. You can override the Campaign default language with the language specified in your system's LANG environment variable. If you change the LANG environment variable, the following 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 following limitations:

- It is not supported for Japanese operating systems. To install Campaign with a single locale on a Japanese OS, contact IBM Technical Support.

**Note:** The multi-locale feature installed on non-Japanese operating system environments does correctly support ja as a user locale preference.

- It is not supported by all IBM applications. See the documentation for each application for information about multi-locale support.
- In multi-locale installations of 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 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 Campaign for a non-ASCII language or a non-US locale

To configure Campaign for localized data or non-ASCII locales, first read all of “About using non-ASCII data or non-US locales” on page 173. Then, complete the tasks in the following list. Each step is described in detail later in this section.

1. “Set the operating system language and regional settings.”
2. “Set encoding parameters for the web application server (WebSphere only)” on page 177.
3. “Set Campaign language and locale property values” on page 177.
4. “Un-map and re-map system tables” on page 178.
5. “Test your database and server configurations” on page 178.

**Important:** Do not skip any of the tasks or steps. Skipping a step could result in incorrect or incomplete configuration, causing errors or corrupted data.

### Set the operating system language and regional settings

On the server running the Campaign Sever and the system where the Campaign web application is deployed, configure the operating system language and regional settings.

**Note:** 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 Campaign does not require that you install support for X Fonts and translations.

If the language 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>)
- AIX 5.3 National Language Support Guide and Reference (<http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.nls/doc/nlsgrdf/nlsgrdf.htm>)
- HP-UX 11 Internationalization Features White Paper (<http://docs.hp.com/en/5991-1194/index.html>)

#### About 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.

## Set encoding parameters for the web application server (WebSphere only)

For WebSphere® only, if you are using 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 on doing this, see the IBM WebSphere documentation.

## Set Campaign language and locale property values

Campaign supports multiple languages and locales for a single installation.

Use the Campaign configuration settings in Marketing Platform to set the values of configuration parameters that control how 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 used for translated Campaign messages (for example, text in the 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:** Campaign supports non-ASCII column names, table names, and database names. However, 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.

Choose **Settings > Configuration** to adjust the following properties. 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 > unicaAListener > logStringEncoding
- Campaign > unicaAListener > systemStringEncoding

## Un-map and re-map system tables

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 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. Campaign retains its existing settings for a data source until the data source is no longer used (that is, until it is unmapped).

## Test your database and server configurations

Before you start creating campaigns or other objects, you should ensure that your database and server settings are correctly configured.

Perform the following tests to ensure that your configurations are correct:

- “Test the database configuration”
- “Test that your attribute table is correctly configured”
- “Test campaigns and flowcharts that have ASCII and non-ASCII characters” on page 179
- “Test flowchart input and output that contains ASCII and non-ASCII characters” on page 179
- “Test that the correct language directory is used” on page 180
- “Test the date formats in calendar reports” on page 180
- “Test that the currency symbol displays correctly for your locale” on page 181

### 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 Marketing 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 Campaign Settings page appears.
2. Select **Manage Table Mappings**
3. In the **Table Mappings** dialog, in the list of IBM 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 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," 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 IBM 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 Campaign, select **Analytics > 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 Campaign, click **Analytics > 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 Campaign for multiple locales

Configuring 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: Campaign must be installed

The information in the rest of this section is presented under the assumption that Campaign is already installed AND that the 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 IBM EMM and configure the encoding properties listed in the following table. Set the values for the properties as specified here.

Property	Value
Campaign > partitions > partition[n] > dataSources > [data_source_name] > StringEncoding	WIDEUTF-8
Campaign > partitions > partition[n] > server > encoding > stringEncoding	UTF-8
Campaign > unicaACListener > logStringEncoding	UTF-8
Campaign > unicaACListener > 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 Campaign Listener.

### Configure encoding properties (Oracle)

Log in to Marketing Platform and configure the encoding properties listed in the following table. Set the values for the properties as specified here.

Property	Value
Campaign > partitions > partition[n] > dataSources > [data_source_name] > StringEncoding	UTF-8
Campaign > partitions > partition[n] > server > encoding > stringEncoding	UTF-8
Campaign > unicaACListener > logStringEncoding	UTF-8
Campaign > unicaACListener > systemStringEncoding	UTF-8

### Configure date/time settings (Oracle)

Property	Value
Campaign > partitions > partition[n] > [data_source_name] > DateFormat	DELIM_Y_M_D
Campaign > partitions > partition[n] > [data_source_name] > DateOutputFormatString	%Y-%m-%d
Campaign > partitions > partition[n] > [data_source_name] > DateTimeFormat	DT_DELIM_Y_M_D
Campaign > partitions > partition[n] > [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>
Campaign > partitions > partition[n] > [data_source_name] > SQLOnConnect	<p>ALTER SESSION SET</p> <p>NLS_LANGUAGE='American'</p> <p>NLS_TERRITORY='America'</p> <p>NLS_TIMESTAMP_FORMAT='YYYY-MM-DD hh24:mi:ss'</p> <p>NLS_DATE_FORMAT='YYYY-MM-DD'</p>

### Configure environment variables (Oracle)

On 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 Campaign client machines, modify the 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
```

## 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).

## Configuring for multiple locales on DB2

To configure IBM 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 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)

Log in to Marketing Platform and configure the encoding properties listed in the following table. Set the values for the properties as specified here.

For important information, see the property descriptions in “Campaign configuration properties” on page 227.

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)

On the Marketing Platform Configuration page, set the following values for the following date/time properties:

Property	Value
Campaign > partitions > partition[n] > [data_source_name] > DateOutputFormatString	%Y-%m-%d
Campaign > partitions > partition[n] > [data_source_name] > DateTimeFormat	DT_DELIM_Y_M_D
Campaign > partitions > partition[n] > [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 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 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),” 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
```



---

## Appendix A. Configuration properties on the configuration page

This section describes the configuration properties found on the Configuration page.

---

### Marketing Platform configuration properties

This section describes the Marketing Platform configuration properties on the Configuration page.

#### General | Navigation

##### TCP port for secure connections

###### Description

Specifies the SSL port in the web application server on which the Marketing Platform is deployed. This property is used internally for communication among IBM EMM products.

###### Default value

7001

##### TCP port for standard connections

###### Description

Specifies the HTTP port in the web application server on which the Marketing Platform is deployed. This property is used internally for communication among IBM EMM products.

###### Default value

7001

##### IBM Marketing Platform URL

###### Description

Specifies the URL used for IBM EMM. This is set at installation time and normally should not be changed. Note that the URL contains the domain name, as shown in the following example.

```
protocol://machine_name_or_IP_address.domain_name:port_number/  
context-root
```

The machine name should not be localhost.

###### Default value

Not defined

###### Example

In an environment configured for SSL, the URL might look like this:

```
https://machineName.companyDomain.com:8080/unica
```

## General | Data filtering

### Default table name

#### Description

In conjunction with Default audience name, determines the set of data filters (that is, the data configuration) from which the data filter user interface in IBM EMM reads filters and assignments.

#### Default value

Undefined

#### Valid Values

Physical name of the customer table that contains the fields used as data filter criteria. Maximum of 50 characters of type varchar.

### Default audience name

#### Description

In conjunction with Default table name, determines the set of data filters (that is, the data configuration) from which the data filter user interface in IBM EMM reads filters and assignments.

#### Default value

Undefined

#### Valid Values

When configuring data filters for Distributed Marketing, the name must be the same as the name given to an audience level in Campaign. Maximum of 50 characters of type varchar.

## General | Password settings

Properties in this category specify the policies that apply to IBM EMM passwords. Most of these password options apply only to passwords for internal users (created within the Marketing Platform), not to external users (imported from an external system). The exception is the Maximum failed login attempts allowed property, which affects both internal and external users. Also note that this property does not override any similar restriction set in an external system.

### Maximum failed login attempts allowed

#### Description

Specifies the maximum number of times an invalid password may be entered each time a user logs in. If the maximum is reached, the user is disabled in the IBM EMM system, and no one can log in as that user.

If set to zero or less, the system allows an infinite number of consecutive failures.

#### Default value

3

#### Valid Values

Any integer

## Password history count

### Description

Specifies the number of old passwords the system retains for a user. The user is not allowed to reuse any passwords within this list of old passwords. If the value is set to zero or less, then no history is retained, and the user may reuse the same password repeatedly. Note that the password history count does not include the password initially assigned to a user account when it is created.

### Default value

0

### Valid Values

Any integer

## Validity (in days)

### Description

Specifies the number of days before a user's password expires.

If the value is zero or less, then the password never expires.

If the value is greater than zero, users are required to change their password the first time they log in, and the expiration interval is counted from the date of the first login.

If you change this value after users and passwords have been created, the new expiration date takes effect for existing users the next time they change their password.

### Default value

30

### Valid Values

Any integer

## Blank passwords allowed

### Description

Specifies whether the a blank password is allowed.If you set this to true you should also set Minimum character length=0.

### Default value

true

### Valid Values

true | false

## Allow identical user name and password

### Description

Specifies whether the user's password is allowed to be the same as the user's login name.

### Default value

false

**Valid Values**

true | false

**Minimum number of letter characters****Description**

Specifies the minimum number of letters required in a password. If the value is zero or less, then there is no minimum requirement.

**Default value**

0

**Valid Values**

Any integer

**Minimum number of numeric characters****Description**

Specifies the minimum number of numbers required in a password. If the value is zero or less, then there is no minimum requirement.

**Default value**

0

**Valid Values**

Any integer

**Minimum character length****Description**

Specifies the minimum length of a password. If the value is zero or less, then there is no minimum requirement. If you set the value to greater than 0, you should also set Blank passwords allowed=false.

**Default value**

4

**Valid Values**

Any integer

**General | Miscellaneous**

Properties in this category specify values that are used internally, as well as a value you may need to set for the locale.

**TokenLifetime****Description**

Specifies the length of time, in seconds, that a token generated by the Marketing Platform is valid. It is part of the suite sign-on implementation, and you should not change this value.

**Default value**

15

**Valid Values**

Any positive integer

## Default region

### Description

Specifies the default locale for the Marketing Platform. If you plan to install Campaign, you should set this value to match the locale set for Campaign in Campaign's defaultLocale property.

### Default value

English

### Valid Values

Supported locales

## Trusted application enabled

### Description

When this value is set to True, the Marketing Platform must be deployed in an environment that has an SSL port and the IBM Marketing Platform URL property in the General > Navigation category must be set to use https.

### Default value

False

### Valid Values

True | False

## Platform

### Region setting

#### Description

Specifies the locale preference for IBM EMM users. When you set this property on the Configuration page, the setting you apply is the default setting throughout IBM EMM for all users, except those whose locale preference is set individually through the Marketing Platform's User page. When you set this property for an individual user, the setting you apply for that user overrides the default setting.

This preference setting affects display of the language, time, numbers, and dates in IBM EMM applications.

Availability of locales may vary depending on the IBM EMM application, and not all IBM applications support this locale setting in the Marketing Platform. See specific product documentation to determine availability and support for the Region setting property.

#### Default value

English (United States)

## Help server

### Description

The URL of the server on which IBM hosted online help is installed. If IBM EMM users have internet access, you should not change the default value, which points to the online help server maintained and updated by IBM .

**Default value**

The URL of the hosted help server.

**Valid Values**

Any server on which IBM hosted help is installed.

**IBM Marketing Operations - Campaign integration****Description**

A flag indicating whether Marketing Operations and Campaign are installed together and integrated. For more information about configuring this integration, see the *IBM Marketing Operations and Campaign Integration Guide*.

**Default value**

False

**Valid Values**

True | False

**IBM Marketing Operations - Offer integration****Description**

For systems that integrate Marketing Operations with Campaign, this flag indicates whether offer integration is also enabled. Offer integration enables the ability to use Marketing Operations to perform offer lifecycle management tasks. For more information about configuring this integration, see the *IBM Marketing Operations and Campaign Integration Guide*.

**Default value**

False

**Valid Values**

True | False

**Start page****Description**

The URL of the page that appears when users log in to IBM EMM. The default is the default dashboard.

**Default value**

The default dashboard.

**Valid Values**

Any IBM EMM URL except form submissions pages, edit pages, and search result pages.

**Domain name****Description**

The name of the domain where IBM EMM is installed. The value is set during installation. You should not change this unless the domain name changes.

**Default value**

Not defined



## Disable Page Tagging

### Description

When set to the default value of `False`, IBM uses the Site ID code that was entered during Marketing Platform installation to gather basic statistics that track overall product usage trends to develop and improve IBM products. If you do not want to have such information collected, set this property to `True`.

### Default value

`False`

### Valid Values

`True` | `False`

## Platform | Scheduler

### Client polling interval

#### Description

Campaign polls the IBM EMM Scheduler for jobs at regular intervals, specified in milliseconds by this value. The default value is 60 seconds. You should not set this property to any value less than 10000 (10 seconds), because this can decrease campaign performance.

#### Default value

`60000`

### Client initialization delay

#### Description

The amount of time, expressed in milliseconds, that the Campaign scheduler thread waits before polling the IBM EMM Scheduler for jobs when Campaign first starts up. Set this value to be at least as long as it takes for Campaign to fully start up on your system. The default value is five minutes.

#### Default value

`300000`

#### Valid Values

Any integer

### Maximum Unknown Status Polling Count

#### Description

Specifies the number of times the scheduler checks the status of a scheduled run whose status cannot be determined. After this limit is reached, the run status is listed as Unknown on the Settings > Scheduled Tasks page.

#### Default value

`5`

#### Valid Values

Any integer

## Enable Scheduler

### Description

Specifies whether the scheduler is enabled. Set this property to False if you do not want users to be able to use the scheduler. The False setting turns off the scheduler for all products that use it.

### Default value

True

### Valid Values

True | False

## Platform | Scheduler | Recurrence definitions

Properties in this category set the recurrence patterns for the IBM EMM Scheduler. These appear in the dialog box you use if you set a recurrence pattern when you create a schedule. You can use the Recurrence template to create your own recurrence pattern, using any valid Cron expression.

### Every hour

#### Description

The job is triggered every hour.

#### Default value

0 0 0/1 \* \* ?

### Every day

#### Description

The job is triggered every 24 hours.

#### Default value

0 0 0 \* \* ?

### Every [day of week] at 12:00 am

#### Description

The job is triggered on the specified day of the week at 12:00 am.

#### Default value

- Monday - 0 0 0 ? \* MON
- Tuesday - 0 0 0 ? \* TUE
- Wednesday - 0 0 0 ? \* WED
- Thursday - 0 0 0 ? \* THU
- Friday - 0 0 0 ? \* FRI
- Saturday - 0 0 0 ? \* SAT
- Sunday - 0 0 0 ? \* SUN

### [First|Last] day of every month at 12:00 am

#### Description

The job is triggered on the specified day of the month (first or last) at 12:00 am.

**Default value**

- First day of every month - 0 0 0 1 \* ?
- Last day of every month - 0 0 0 L \* ?

**[First|Last] day of every quarter at 12:00 am****Description**

The job is triggered on the specified day of the calendar quarter (first or last day) at 12:00 am.

**Default value**

- First day of every quarter - 0 0 0 1 \* JAN,APR,JUL,OCT
- Last day of every quarter - 0 0 0 L \* MAR,JUN,SEP,DEC

**[First|Last] day of every year at 12:00 am****Description**

The job is triggered on the specified day of the year (first or last) at 12:00 am.

**Default value**

- First day of every year - 0 0 0 1 ? JAN \*
- Last day of every year - 0 0 0 L ? DEC \*

**Every [month] at 12:00 am****Description**

The job is triggered on the first day of the specified month at 12:00 am.

**Default value**

- Every January - 0 0 0 1 ? JAN \*
- Every February - 0 0 0 1 ? FEB \*
- Every March - 0 0 0 1 ? MAR \*
- Every April - 0 0 0 1 ? APR \*
- Every May - 0 0 0 1 ? MAY \*
- Every June - 0 0 0 1 ? JUN \*
- Every July - 0 0 0 1 ? JUL \*
- Every August - 0 0 0 1 ? AUG \*
- Every September - 0 0 0 1 ? SEP \*
- Every October - 0 0 0 1 ? OCT \*
- Every November - 0 0 0 1 ? NOV \*
- Every December - 0 0 0 1 ? DEC \*

## **Platform | Scheduler | Schedule registrations | Campaign | [Object type]**

A different category exists for each of the object types that can be scheduled with the IBM scheduler. Properties in these categories should not normally be changed.

**Executor class name****Description**

The class that the IBM Scheduler uses to trigger a flowchart or mailing run.

**Default value**

**Status polling interval**

**Description**

At regular intervals, the IBM Scheduler polls Campaign for the run status of scheduled objects that have not reported status. The interval is specified here in milliseconds. The default value is 10 minutes. Setting a more frequent polling interval (a smaller value) affects the system performance. Setting a less frequent polling interval (a larger value) reduces the load on the system. For example, you might want to set a less frequent polling interval when you have a large number of Campaign flowcharts that take more than 10 minutes to complete.

**Default value**

600000

**Platform | Scheduler | Schedule registrations | Campaign | [Object type] | [Throttling group]**

Default throttling groups exist for each of the object types that can be scheduled with the IBM scheduler. You can use the throttling group template to create additional groups.

**Throttling threshold**

**Description**

The greatest number of schedules associated with this group that can run concurrently. The configured scheduler groups appear in the **Scheduler Group** drop-down list in the Scheduler user interface for creating and editing schedules. The default throttling group is set to 999, which is effectively no limit. Because all schedules must belong to a throttling group, you should leave this value unchanged so that schedules that you do not want to throttle can be assigned to this group.

**Default value**

**Valid Values**

Any positive integer.

**Platform | Security**

**Login method**

**Description**

Specifies the authentication mode for all IBM EMM products installed and configured to work together, as follows:

- If you set the value to Windows integrated login, IBM EMM products use Windows Active Directory for authentication.
- If you set the value to IBM Marketing Platform, IBM EMM products use the Marketing Platform for authentication and authorization.
- If you set the value to LDAP, IBM EMM products use an LDAP server for authentication.

- If you set the value to Web access control, IBM EMM products use web access control software for authentication.

#### **Default value**

IBM Marketing Platform

#### **Valid Values**

Windows integrated login | IBM Marketing Platform | LDAP | Web access control

## **Platform | Security | Login method details | Windows integrated login**

### **Domain**

#### **Description**

Sets the value of JCIFS SMB client library property `jcifs.smb.client.Domain`. Specifies the domain that is used if no domain is specified in an SMB URL. Set this value to the Windows domain name. For most environments, set either this property or the Domain Controller property.

#### **Default value**

Undefined.

#### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory server and Windows integrated login is enabled.

### **Client Timeout**

#### **Description**

Sets the value of JCIFS SMB client library property `jcifs.smb.client.soTimeout`. Specifies the amount of time, in milliseconds, before sockets are closed if there is no activity between the client and server. This number should be as small as possible but long enough to allow the protocol handshaking to complete, which depends on network characteristics.

#### **Default value**

1000

#### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory server and Windows integrated login is enabled.

### **Cache Policy**

#### **Description**

Sets the value of JCIFS SMB client library property `jcifs.netbios.cachePolicy`. Specifies the amount of time, in seconds, that the NetBIOS name is cached to reduce redundant name queries. If the

value is set to 0 is no caching takes place. If the value is set to -1 the cache is never cleared. This property is used when SMB signing is enabled and required in a Windows 2003 domain.

**Default value**

0

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory server and Windows integrated login is enabled.

## **Domain Controller**

**Description**

Sets the value of JCIFS SMB client library property `jcifs.http.domainController`. Specifies the IP address of a server that should be used to authenticate HTTP clients (used by `NtlmHttpFilter` and `NetworkExplorer`). You may use the IP address of a workstation in the domain specified in the `Domain` property. For most environments, set either this property or the `Domain` property.

**Default value**

Undefined.

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory server and Windows integrated login is enabled.

## **IP of the WINS server**

**Description**

Sets the value of JCIFS SMB client library property `jcifs.netbios.wins`. Specifies the IP address of the WINS server. You may enter multiple IP addresses, separated by commas (for example `192.168.100.30, 192.168.100.31`). The WINS server is queried to resolve the domain specified in the `Domain` property to an IP address of a domain controller. This property is required when accessing hosts on different subnet (such as a domain controller by name) and it is highly recommended if a WINS server is available.

**Default value**

Undefined

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory server and Windows integrated login is enabled and Windows integrated login is enabled.

## **Strip Domain**

**Description**

Specifies whether the Marketing Platform removes a domain from users' login names when they access the IBM EMM. If your Windows

configuration requires a domain to be included with users' login names when they log in, set this value to False.

**Default value**

True

**Valid Values**

True | False

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory server and Windows integrated login is enabled.

## **Retry on Authentication Failure**

**Description**

If a user login fails, the system allows another login attempt if this value is set to True. Set to False if you want to disallow more than one login attempt.

**Default value**

True

**Valid Values**

True | False

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory server and Windows integrated login is enabled.

## **Platform | Security | Login method details | LDAP**

### **LDAP server host name**

**Description**

Specifies the name or IP address of the LDAP server. Set the value to the machine name or IP address of the LDAP server. For example:  
machineName.companyDomain.com

If you are integrating with Windows Active Directory, use the server name instead of the DNS name.

**Default value**

Undefined

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

### **LDAP server port**

**Description**

Specifies the port on which the LDAP server listens. Set the value to the appropriate port number. Typically, the port number is 389 (636 if SSL is used).

**Default value**

389

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**User search filter****Description**

Specifies the filter to use to search for users. Valid values are any valid LDAP search filter (see RFC 2254). Note that you must XML-escape any XML characters in this value.

Typically, the value for the user login attribute is `uid` for LDAP servers and `sAMAccountName` for Windows Active Directory servers. You should verify this on your LDAP or Active Directory server. If your LDAP server is Windows Active Directory, you should change the default value of this property to use `sAMAccountName` rather than `uid`. For example:

```
(&(|(objectClass=user)(objectClass=person))(sAMAccountName={0}))
```

**Default value**

```
(&(|(objectClass=user)(objectClass=person))(uid={0}))
```

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**Use credentials stored in IBM Marketing Platform****Description**

Specifies whether the Marketing Platform uses credentials from the Marketing Platform database when searching the LDAP or Windows Active Directory server during user authentication (at login time).

If this value is `true`, the Marketing Platform uses credentials from the Marketing Platform database, and you must set the appropriate values for the IBM Marketing Platform user for LDAP credentials and Data source for LDAP credentials properties in this category.

If your LDAP or Windows Active Directory server does not allow anonymous access, set this value to `true`.

If this value is `false`, the Marketing Platform connects with the LDAP or Windows Active Directory server anonymously. You may set this value to `false` if your LDAP or Windows Active Directory server allows anonymous access.

**Default value**

false

**Valid Values**

true | false

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.



## IBM Marketing Platform user for LDAP credentials

### Description

Specifies the name of the IBM EMM user that has been given LDAP administrator login credentials. Set this value if you set the Use credentials stored in UIBM Marketing Platform property in this category to true.

Set the value of this property to the user name you created for the IBM EMM user when you configured LDAP integration. This property works in conjunction with the Data source for LDAP credentials property in this category.

### Default value

asm\_admin

### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## Data source for LDAP credentials

### Description

Specifies the Marketing Platform data source for LDAP administrator credentials. Set this value if you set the Use credentials stored in IBM Marketing Platform property in this category to true.

Set the value of this property to the data source name you created for the IBM EMM user when you configured LDAP integration. This property works in conjunction with the IBM Marketing Platform user for LDAP credentials property in this category.

### Default value

Undefined

### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## Base DN

### Description

Specifies the base distinguishing name (DN) pointing to the root of the LDAP directory structure.

### Default value

[CHANGE ME]

### Valid Values

Any valid DN (see RFC 1779, RFC 2253)

### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## Require SSL for LDAP connection

### Path

Platform | Security | LDAP

### Description

Specifies whether the Marketing Platform uses SSL when it connects to the LDAP server to authenticate users. If you set the value to true , the connection is secured using SSL.

### Default value

false

### Valid Values

true | false

## Platform | Security | Login method details | Web access control

### Username pattern

#### Description

Java™ regular expression used to extract the user login from the HTTP header variable in web access control software. Note that you must XML-escape any XML characters in the regular expression. The recommended value for SiteMinder and Tivoli® Access Manager is \w\*

#### Default value

Undefined

#### Valid Values

Any Java regular expression.

#### Availability

This property is used only when the Marketing Platform is configured to integrate with web access control software.

### Web access control header variable

#### Description

Specifies the HTTP header variable configured in the web access control software, which is submitted to the web application server. By default, SiteMinder uses sm\_user and Tivoli Access Manager (TAM) uses iv-user. For TAM, set this value to the user name component of the IBM Raw string, not the IBM HTTP string.

#### Default value

Undefined

#### Valid Values

Any string

#### Availability

This property is used only when the Marketing Platform is configured to integrate with web access control software.

## Platform | Security | LDAP synchronization

### LDAP sync enabled

#### Description

Set to true to enable LDAP or Active Directory synchronization.

#### Default value

false

#### Valid Values

true | false

#### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

### LDAP sync interval

#### Description

The Marketing Platform synchronizes with the LDAP or Active Directory server at regular intervals, specified in seconds here. If the value is zero or less, the Marketing Platform does not synchronize. If the value is a positive integer, the new value takes effect without a restart within ten minutes. Subsequent changes take effect within the configured interval time.

#### Default value

600, or ten minutes

#### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

### LDAP sync delay

#### Description

This the time (in 24 hour format) after which the periodic synchronization with the LDAP server begins, after the Marketing Platform is started. For example an LDAP sync delay of 23:00 and an LDAP sync interval of 600 mean that when the Marketing Platform starts, the periodic synchronization starts to execute at 11:00 PM and executes every 10 minutes (600 seconds) thereafter.

#### Default value

23:00, or 11:00pm

#### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

### LDAP sync timeout

#### Description

The LDAP sync timeout property specifies the maximum length of time, in minutes, after the start of a synchronization before the Marketing Platform marks the process ended. The Platform allows only one synchronization

process to run at a time. If a synchronization fails, it is marked as ended whether it completed successfully or not.

This is most useful in a clustered environment. For example, if the Marketing Platform is deployed in a cluster, one server in the cluster might start an LDAP synchronization and then go down before the process is marked as ended. In that case, the Marketing Platform will wait for the amount of time specified in this property, and then it will start the next scheduled synchronization.

**Default value**

600, (600 minutes, or ten hours)

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **LDAP sync scope**

**Description**

Controls the scope of the initial query to retrieve the set of users. You should retain the default value of SUBTREE for synchronizing with most LDAP servers.

**Default value**

SUBTREE

**Valid Values**

The values are standard LDAP search scope terms.

- OBJECT - Search only the entry at the base DN, resulting in only that entry being returned
- ONE\_LEVEL - Search all entries one level under the base DN, but not including the base DN.
- SUBTREE - Search all entries at all levels under and including the specified base DN.

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **LDAP provider URL**

**Description**

For most implementations, set to the LDAP URL of the LDAP or Active Directory server, in one of the following forms:

- ldap://IP\_address:port\_number
- ldap://machineName.domain.com:port\_number

On LDAP servers, the port number is typically 389 (636 if SSL is used).

If IBM EMM is integrated with an Active Directory server, and your Active Directory implementation uses serverless bind, set the value of this property to the URL for your Active Directory server, using the following form:

ldap:///dc=example,dc=com

**Default value**

Undefined

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**Require SSL for LDAP connection****Path**

Platform | Security | LDAP synchronization

**Description**

Specifies whether the Marketing Platform uses SSL when it connects to the LDAP server to synchronize users. If you set the value to true, the connection is secured using SSL.

**Default value**

false

**Valid Values**

true | false

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**LDAP config IBM Marketing Platform group delimiter****Description**

In the LDAP reference to IBM Marketing Platform group map category, if you want to map one LDAP or Active Directory group to multiple Marketing Platform groups, use the delimiter specified here. It can be any single character that does not appear in the names it is separating.

**Default value**

; (semicolon)

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**LDAP reference config delimiter****Description**

Specifies the delimiter that separates the SEARCHBASE and FILTER components that make up the LDAP or Active Directory reference (described in the LDAP references for IBM Marketing Platform user creation category).

FILTER is optional: if omitted, the Marketing Platform server dynamically creates the filter based on the value of LDAP user reference attribute name.

**Default value**

; (semicolon)

**Valid Values**

Any single character that does not appear in the names it is separating.

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**IBM Marketing Platform user for LDAP credentials****Description**

Specifies the name of IBM EMM user that has been given LDAP administrator login credentials.

Set the value of this property to the user name you created for the IBM EMM user when you configured LDAP integration. This property works in conjunction with the Data source for LDAP credentials property in this category.

**Default value**

asm\_admin

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**Data source for LDAP credentials****Description**

Specifies the Marketing Platform data source for LDAP administrator credentials.

Set the value of this property to the data source name you created for the IBM EMM user when you configured LDAP integration. This property works in conjunction with the IBM Marketing Platform user for LDAP credentials property in this category.

**Default value**

Undefined

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**LDAP user reference attribute name****Description**

Specifies the name that your LDAP or Active Directory server uses for the user attribute in the Group object. Typically, this value is uniquemember in LDAP servers and member in Windows Active Directory servers.

**Default value**

member

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## LDAP BaseDN periodic search enabled

### Description

When this property is set to True, the Marketing Platform performs the LDAP synchronization search using the distinguished name set in the Base DN property under the **IBM EMM | Platform | Security | LDAP** category. If this property is set to False, the Marketing Platform performs the LDAP synchronization search using the groups mapped to LDAP groups under **LDAP reference to IBM Marketing Platform group map**.

### Default value

True

### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## User login

### Description

Maps the IBM EMM user's login to the equivalent user attribute in your LDAP or Active Directory server. User login is the only required mapping. Typically, the value for this attribute is uid for LDAP servers and sAMAccountName for Windows Active Directory servers. You should verify this on your LDAP or Active Directory server.

### Default value

uid

### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## First name

### Description

Maps the First Name user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### Default value

givenName

### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## Last name

### Description

Maps the Last Name user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### Default value

sn

### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **User title**

### **Description**

Maps the Title user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### **Default value**

title

### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **Department**

### **Description**

Maps the Department user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### **Default value**

Undefined

### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **Company**

### **Description**

Maps the Company user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### **Default value**

Undefined

### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **Country**

### **Description**

Maps the Country user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### **Default value**

Undefined

### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.



## **User email**

### **Description**

Maps the Email Address attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### **Default value**

mail

### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **Address 1**

### **Description**

Maps the Address user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### **Default value**

Undefined

### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **Work phone**

### **Description**

Maps the Work Phone user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### **Default value**

telephoneNumber

### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **Mobile phone**

### **Description**

Maps the Mobile Phone user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

### **Default value**

Undefined

### **Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **Home phone**

### **Description**

Maps the Home Phone user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

**Default value**

Undefined

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

**Alternate login**

**Description**

Maps the Alternate Login user attribute in the Marketing Platform to the equivalent user attribute in your LDAP or Active Directory server.

**Default value**

Undefined

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

## **Platform | Security | LDAP synchronization | LDAP reference to IBM Marketing Platform group map**

### **LDAP reference map**

**Description**

Users who are members of the LDAP or Active Directory group specified here are imported to the Marketing Platform group specified in the IBM Marketing Platform group property.

Set the value of this property using the following syntax: SEARCHBASE DELIMITER FILTER where:

SEARCHBASE is the Distinguished Name (DN) of the object.

DELIMITER is the value of the LDAP config AM group delimiter property.

FILTER is the LDAP or Active Directory attribute filter. FILTER is optional: if omitted, the Marketing Platform server dynamically creates the filter based on the value of the LDAP user reference attribute name property.

**Default value**

Undefined

**Availability**

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

### **IBM Marketing Platform group**

**Description**

Users who are members of the LDAP or Active Directory group specified in the LDAP reference group property are imported to the Marketing Platform group specified here.

**Default value**

Undefined

#### Availability

This property is used only when the Marketing Platform is configured to integrate with a Windows Active Directory or other LDAP server.

---

## Reporting configuration properties

For reporting, the IBM EMM suite integrates with IBM Cognos, a third-party business intelligence application. You use the Cognos properties to identify the IBM Cognos system used by your IBM installation. Then, for Campaign, eMessage, and Interact, there are additional configuration properties that you use to set up and customize reporting schemas.

### Reports | Integrations | Cognos [version]

This page displays properties that specify URLs and other parameters for the IBM Cognos system used by this IBM system.

#### Integration Name

##### Description

Read-only. Specifies that IBM Cognos is the third-party reporting or analytical tool used by the IBM EMM 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 specified by the Integration Name property.

##### Default value

Cognos

#### Version

##### Description

Read-only. Specifies the product version of the application 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 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`

Note that 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 IBM Authentication Provider, which means it relies on the Marketing Platform for authentication.

#### **Default value**

anonymous

#### **Valid Values**

- anonymous: means authentication is disabled.
- authenticated: means that the communications between the IBM 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 IBM Authentication Provider.

#### **Default value**

Unica

### **Authentication user name**

#### **Description**

Specifies the login name for the reporting system user. The IBM applications log in to Cognos as this user when Cognos is configured to use the Unica<sup>®</sup> Authentication provider. Note that this user also has access to IBM EMM.

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 is true:

- When the IBM EMM is not installed in the same domain as the IBM Cognos applications.
- When IBM Cognos is accessed using an IP address (within the same network domain) instead of the Fully Qualified Hostname (which is being used to access the IBM EMM applications), even if both the IBM EMM 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 IBM EMM 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 should install IBM Cognos and IBM EMM in the same domain.

### Default value

False

### Valid Values

True | False

## Reports | Schemas | [product] | [schema name] | SQL Configuration

### Table/View Name

#### Description

Specifies the name of the view or table that the SQL script you generate for this reporting schema will create. 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 UPPER-CASE letters

Following is the naming convention you should use:

- Start the name with the letter "UAR"
- Add a one-letter code to represent the IBM EMM 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 list of codes, below.

For example: UARC\_COPERF\_DY is the name of the reporting view or table for Campaign Offer Performance by Day.

Following is the list of IBM EMM application codes.

- Campaign: C
- eMessage: E
- Interact: I
- Distributed Marketing: X
- Marketing Operations: 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 Input Datasource (JNDI)

### Description

Specifies the name of the JNDI data source that identifies the 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 you select when you generate the SQL scripts for the 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 (or table) across all time.

### Audience Key

#### Description

Specifies the name of the column that is the Audience Key for the audience level 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 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 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 supported by this reporting schema.

**Default value**

UA\_ResponseHistory

**Over Time Variations****Description**

Specifies the calendar time periods 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 this form 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 or table for the column 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 UPPER-CASE 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 this form to add the response metrics 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 or table for the column 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 UPPER-CASE 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 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 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 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 supported by this reporting schema.

#### Default value

UA\_ResponseHistory

### Over Time Variations

#### Description

Specifies the calendar time periods 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

This schema supports reporting on campaign-detailed responses, 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.

### Response History Table

**Description**

Specifies the name of the Response History table for the audience level supported by this reporting schema.

**Default value**

UA\_ResponseHistory

## Reports | Schemas | Campaign | Campaign Offer Response Breakout | Columns | [Response Type]

Use this form to add to the reporting schema any custom response types you want to include in your reports.

### Column Name

**Description**

Specifies the name to use in the reporting view or table for the column 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 UPPER-CASE letters, and it cannot have spaces.

### Response Type Code

**Description**

The response type code for the specified response type. This is the value 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 Campaign installation may have additional custom response type codes.

## Control Treatment Flag

### Description

If you use the standard IBM Cognos reports provided in the IBM EMM 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

This schema supports reporting on campaign-detailed contacts, 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 Campaign reports use this schema.

## Audience Key

### Description

Specifies the name of the column that is the Audience Key for the audience level 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 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 supported by this reporting schema.

### Default value

UA\_DtlContactHist

## Reports | Schemas | Campaign | Campaign Offer Contact Status Breakout | Columns | [Contact Status]

### Column Name

### Description

Specifies the name to use in the reporting view or table for the column 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 UPPER-CASE letters, and it cannot have spaces.

## Contact Status Code

### Description

The name of the contact status code. This is the value 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 Campaign installation may have additional custom contact status types.

## Reports | Schemas | Campaign | Campaign Custom Attributes | Columns | [Campaign Custom Column]

Use this form to add to the reporting schema any custom campaign attributes that you want to include in your reports.

### Column Name

#### Description

Specifies the name to use in the reporting view or table for the attribute 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 UPPER-CASE 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 Campaign, select StringValue.

## Reports | Schemas | Campaign | Campaign Custom Attributes | Columns | [Offer Custom Column]

Use this form to add to the reporting schema any custom offer attributes that you want to include in your reports.

### Column Name

#### Description

Specifies the name to use in the reporting view or table for the attribute 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 UPPER-CASE 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 this form to add to the reporting schema any custom cell attributes that you want to include in your reports.

#### Column Name

##### Description

Specifies the name to use in the reporting view or table for the attribute 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 UPPER-CASE 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 Interact reporting schemas reference three separate databases: the design time, run time, and learning databases. Use the properties from this page to specify the JNDI names of the data sources for those databases.

The data sources 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.

Note that the database type of the data sources must match the database type 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 Interact design time database, which is also the Campaign system tables.

#### **Default value**

campaignPartition1DS

### **Interact Runtime Datasource (JNDI)**

#### **Description**

Specifies the name of the JNDI data source that identifies the Interact runtime database.

#### **Default value**

InteractRTDS

### **Interact Learning Datasource (JNDI)**

#### **Description**

Specifies the name of the JNDI data source that identifies the 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 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 supported by this reporting schema.

#### Default value

UA\_Dt1ContactHist

### Response History Table

#### Description

Specifies the name of the Response History table for the audience level supported by this reporting schema.

#### Default value

UA\_ResponseHistory

### Over Time Variations

#### Description

Specifies the calendar time periods used by the "over time" reports supported by this schema.

#### Default value

Hour, Day

#### Valid Values

Hour, Day, Week, Month, Quarter, Year

## Reports | Schemas | eMessage eMessage Tracking Datasource (JNDI)

#### Description

Specifies the name of the JNDI data source that identifies the eMessage tracking tables, which are located in the 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 you select when you generate the SQL scripts for the views or reporting tables.

**Default value**

campaignPartition1DS

---

## Campaign configuration properties

This section describes the Campaign configuration properties found on the Configuration page.

### Campaign

To specify the locales and component applications that your installation supports, choose **Settings > Configuration**, then click the Campaign category.

#### currencyLocale

**Description**

The currencyLocale property is a global setting that controls how currency is displayed in the Campaign web application, regardless of the display locale.

**Important:** No currency conversion is performed by 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 Campaign supports. The value of this property is set by the installer when you install Campaign. For example:  
de,en,fr,ja,es,ko,pt,it,zh,ru.

**Default value**

All languages/locales into which 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 Campaign. The value of this property is set by the installer when you install Campaign.

**Default value**

en

**acoInstalled**

**Path**

**Description**

The `acoInstalled` property specifies whether Contact Optimization is installed.

When Contact Optimization is installed and configured, set the value to `yes`, which causes the Contact Optimization process to be displayed in flowcharts. If the value is `true` and Contact Optimization 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 Distributed Marketing is installed. When Distributed Marketing is installed and configured, set the value to `true`, which causes the Distributed Marketing features to be available in the Campaign user interface.

**Default value**

false

**Valid Values**

true | false

## Campaign | collaborate

The properties in this category pertain to Distributed Marketing configuration.

**CollaborateIntegrationServicesURL**

**Description**

The `CollaborateIntegrationServicesURL` property specifies the server and port number of Distributed Marketing. This URL is used by Campaign when a user publishes a flowchart to Distributed Marketing.

**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 IBM applications. It specifies the Uniform Resource Identifier of the 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 IBM 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 IBM 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 Campaign web application server. If your installation of 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 Campaign web application server for secure connections. If your installation of 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 Campaign. If your installation of 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*

### 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 Campaign web application when SiteMinder is used; this property is also used for internal communication with other IBM EMM applications, such as eMessage and 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 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 IBM applications. It specifies the Uniform Resource Identifier of the 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=`

**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=`

**displayName**

**Configuration category**

Campaign|navigation

**Description**

The `displayName` property specifies the link text used for the Campaign link in the drop-down menu that exists in the GUI of each IBM product.

**Default value**

Campaign

## Campaign | caching

The properties in the caching category specify the length of time that cached data for channels, initiatives, campaigns, sessions, and offers is retained.

### **offerTemplateDataTTLSeconds**

#### **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**

#### **Description**

The campaignDataTTLSeconds property specifies the length of time, in seconds, that the system retains the Campaign cache data (Time to Live). An empty value means the cache data is never purged.

#### **Default value**

600 (10 minutes)

### **sessionDataTTLSeconds**

#### **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**

#### **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**

#### **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**

### **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**

### **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**

### **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 used to configure all Campaign partitions, including the default partition, which is named `partition1`. One category should be created for each Campaign partition. This section describes the properties in the `partition[n]` category, which apply to all partitions you configure in Campaign.

### **Campaign | partitions | partition[n] | eMessage**

Define properties in this category to define characteristics of recipient lists and specify the location of resources that upload the lists to IBM EMM Hosted Services.

### **eMessagePluginJarFile**

#### **Description**

Complete path to the location of the file that operates as the Recipient List Uploader (RLU). This plug-in to Campaign uploads OLT data and associated metadata to the remote services hosted by IBM. The location that you specify must be the full local directory path in the file system for the computer that hosts the Campaign web application server.

The IBM installer populates this setting automatically for the default partition when you run the installer. For other partitions, you must configure this property manually. Because there is only one RLU for each eMessage installation, all partitions must specify the same location for the RLU.

Do not change this setting unless IBM instructs you to do so.

**Default value**

No default value defined.

**Valid Values**

Full local directory path to where you installed the Campaign web server.

**defaultSeedInterval****Description**

The number of messages between seed messages if defaultSeedType is Distribute list.

**Default value**

1000

**defaultSeedType****Description**

The default method that eMessage uses to insert seed addresses into a recipient list.

**Default value**

Distribute IDS

**Valid Values**

- Distribute IDS - Distribute IDs evenly, based on the size of the recipient list and the number of seed addresses available, inserts seed addresses at equal intervals throughout the entire recipient list.
- Distribute list - Insert seed address for every defaultSeedInterval IDs in main list. Inserts the entire list of available seed addresses at specified intervals throughout the recipient list. You must specify the interval between insertion points.

**oltTableNamePrefix****Description**

Used in the generated schema for the output list table. You must define this parameter.

**Default value**

OLT

**Valid Values**

The prefix can contain no more than 8 alphanumeric or underscore characters, and must start with a letter.

**oltDimTableSupport****Description**

This configuration parameter controls the ability to add dimension tables to output list tables (OLT) created in the eMessage schema. Dimension tables are required to use advanced scripting for email to create data tables in email messages.

The default setting is False. You must set this property to True so that marketers can create dimension tables when they use the eMessage process

to define a recipient list. For more information about creating data tables and working with advanced scripts for email, see the *IBM eMessage User's Guide*.

**Default value**

False

**Valid Values**

True | False

**Campaign | partitions | partition[n] | reports**

These configuration properties define 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 using 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 using 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 using 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 using 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 using 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 using 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 using XPath notation.

#### **Default value**

```
/content/folder[@name='Affinium Campaign - Object Specific Reports']/folder[@name='campaign']/folder[@name='cached']
```

### **campaignAnalysisTabEmessageOnDemandFolder**

#### **Description**

The campaignAnalysisTabEmessageOnDemandFolder property specifies the location of the folder that contains the eMessage reports listed on the Analysis tab of a campaign. The path is specified using XPath notation.

#### **Default value**

```
/content/folder[@name='Affinium Campaign']/folder[@name='eMessage Reports']
```

### **campaignAnalysisTabInteractOnDemandFolder**

#### **Description**

Report server folder string for Interact reports.

#### **Default value**

```
/content/folder[@name='Affinium Campaign']/folder[@name='Interact Reports']
```

### Availability

This property is applicable only if you have installed 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 have installed Interact.

## **Campaign | partitions | partition[n] | validation**

The Validation Plugin Development Kit (PDK), delivered with Campaign, allows third parties to develop custom validation logic for use in 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 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 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 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 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 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), 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, 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 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 the this category define an audience level field. These properties are populated when a user creates audience levels on the Administration page in 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 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 Campaign. You should not edit this property.

### **Default value**

No default value defined.

## **Campaign | Partitions | partition[n] | dataSources**

Choose Campaign | Partitions | partition[n] | dataSources to configure how IBM Campaign interacts with databases, including its own system tables.

These properties specify the databases that IBM Campaign can access and they control many aspects of how queries are formed.

Each data source that you add in Campaign is represented by a category under partition[n] > dataSources > DATA\_SOURCE\_NAME.

**Note:** The Campaign system tables data source for each partition must be named UA\_SYSTEM\_TABLES in the Marketing Platform, and a dataSources > UA\_SYSTEM\_TABLES category must exist in the Configuration page for every Campaign partition.

## **AccessLibrary**

### **Description**

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 Campaign environment. For example, 64-bit 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 Campaign to access, for example, Teradata).

## **Additional libraries for AIX**

### **Description**

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 as required (for example, to libodb4dDD.so, overriding the default selection of libodb4d.so).

### **Default value**

No default value defined.

## **AliasPrefix**

### **Description**

The AliasPrefix property specifies the way Campaign forms the alias name that 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

## **AllowBaseJoinsInSelect**

### **Description**

This property determines whether 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 Campaign server.

### **Default value**

TRUE

### **Valid Values**

TRUE | FALSE

## **AllowSegmentUsingSQLCase**

### **Description**

The AllowSegmentUsingSQLCase property specifies whether the Campaign 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 of the following conditions are met:

- Segments are mutually exclusive.
- All segments come from a single table.
- Criteria for each segment are based on the IBM macro language.

In this case, Campaign generates a single SQL CASE statement to perform segmentation, followed by segment-by-field processing on the Campaign application server.

### **Default value**

TRUE

### **Valid Values**

TRUE | FALSE

## **AllowTempTables**

### **Description**

The AllowTempTables property specifies whether 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.



When temporary tables are enabled, each time a query is issued against the database (for example, by the Segment process), the resulting IDs are written in a temporary table in the database. When an additional query is issued, Campaign may use that temporary table to retrieve rows from the database.

If temporary tables are not enabled, Campaign retains the selected IDs in the server memory. The additional query retrieves IDs from the database and matches them to the IDs in server memory.

For more information about controlling temporary table joins, see `MaxTempTableJoinPctSelectAll` and `MaxTempTableJoinPctWithCondition`.

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

**ASMSaveDBAuthentication**

**Description**

The `ASMSaveDBAuthentication` property specifies whether, when you log in to Campaign and map a table in a data source you did not previously log in to, Campaign saves your user name and password in IBM EMM.

If you set this property to TRUE, 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, 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**

**Description**

The `ASMUserForDBCredentials` property specifies the IBM EMM user name that is assigned to the Campaign system user (required for accessing the Campaign system tables).

This property is undefined by default.

**Default value**

No default value defined.

**BulkInsertBlockSize**

**Description**

The `BulkInsertBlockSize` property defines the maximum size of a data block, in number of records, that Campaign passes to the database at a time.

**Default value**

**BulkInsertRequiresColumnType****Description**

The BulkInsertRequiresColumnType property is required to support Data Direct ODBC data sources only. Set this property to TRUE for Data Direct 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****Description**

The BulkReaderBlockSize property defines the size of a data block, in number of records, that Campaign reads from the database at a time.

**Default value**

2500

**ConditionalSQLCloseBracket****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****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****Description**

The ConnectionCacheSize property specifies the number of connections that Campaign maintains in a cache for each data source.

By default (N=0), Campaign establishes a new connection to a data source for each operation; if Campaign maintains a cache of connections and a

connection is available for reuse, 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, 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

### Description

Campaign uses the value of the `DateFormat` property to determine how to parse data in date formats when using the Campaign macro language or when interpreting data from date columns.

Set the value of the `DateFormat` property to the format in which 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 your database uses, select a date from the database as described below.

#### Selecting a date by database

Table 49. Date formats

Database	To determine the correct setting
DB2	<p>Connect to the database from a machine that is running the 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>
Netezza®	<p>Connect to the database from a machine that is running the Campaign server. Use <code>odbcetest</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 <code>ANY_TABLE</code> is the name of any existing table</p>

Table 49. Date formats (continued)

Database	To determine the correct setting
Oracle	<p>Log in to the database from the machine that is running the 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 Campaign listener. Use odbctest, in the Campaign\bin directory, to connect and issue the following command:</p> <pre>SELECT getdate()</pre>

### Additional considerations

Note the following database-specific instructions.

#### Teradata

Teradata allows you to define the date format on a per-column basis. In addition to dateFormat and dateOutputFormatString, you must set SuffixOnCreateDateField. To be consistent with our system table settings, use:

- SuffixOnCreateDateField = FORMAT 'YYYY-MM-DD'
- DateFormat = DELIM\_Y\_M\_D
- DateOutputFormatString = %Y-%m-%d

#### SQL Server

If the **Use regional settings when outputting currency, numbers, dates, and times** 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.

#### Default value

DELIM\_Y\_M\_D

#### Valid Values

Any of the formats that are specified in the DATE macro

### DateOutputFormatString

#### Description

The DateOutputFormatString property specifies the format of the date datatype to be used when 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\_strin 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 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

#### Description

The value of the `[data_source_name] > DateTimeFormat` property specifies the format in which Campaign expects to receive datetime/timestamp data from a database. It must match the format 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 in the table for "Selecting a date by database".

**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**

### **Description**

The DateTimeOutputFormatString property specifies the format of the datetime datatype to be used when 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**

### **Description**

The DB2NotLoggedInitially property determines whether Campaign uses the not logged initially SQL syntax when populating temporary tables in DB2. When set to TRUE, this property disables logging for inserts in to temp tables, which improves performance and decreases database resource consumption.

If your version of DB2 does not support the not logged initially syntax, set this property to FALSE.

### **Default value**

TRUE

### **Valid Values**

TRUE | FALSE

## **DB2NotLoggedInitiallyUserTables**

### **Description**

The DB2NotLoggedInitiallyUserTables property determines whether Campaign uses the not logged initially SQL syntax for inserts into DB2 user tables. When set to TRUE, this property disables logging for inserts into the user tables, which improves performance and decreases database resource consumption.

**Note:** 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.

**Note:** The DB2NotLoggedInitiallyUserTables property is not used for the Campaign system tables.

### **Default value**

FALSE

### **Valid Values**

TRUE | FALSE

## DefaultScale

### Description

The DefaultScale property is used when Campaign creates a database field to store numeric values from a flat file or 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 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. For example:

- 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

### Description

The DefaultTextType property is intended for ODBC data sources. This property tells 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

### Description

The DeleteAsRecreate property specifies whether, when an output process is configured to REPLACE TABLE and if TRUNCATE is not supported, Campaign drops and recreates the table or only deletes from the table.

When the value is TRUE, Campaign drops the table and recreates it.

When the value is FALSE, Campaign executes a DELETE FROM from the table.

**Default value**

FALSE

**Valid Values**

TRUE | FALSE

**DeleteAsTruncate**

**Description**

The DeleteAsTruncate property specifies whether, when an output process is configured to REPLACE TABLE, Campaign uses TRUNCATE TABLE or deletes from the table.

When the value is TRUE, Campaign runs a TRUNCATE TABLE from the table.

When the value is FALSE, 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**

**Description**

The DisallowTempTableDirectCreate property specifies the way Campaign adds data to a temp table.

When set to FALSE, 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, 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**

**Description**

Set this property to the data source name (DSN) as assigned in your ODBC configuration for this Campaign data source. This value is undefined by default.

Using the 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 would have a different name in Campaign, but if they refer to the same physical data source and they will have the same DSN value.

#### **Default value**

No default value defined.

### **DSNUsingOSAuthentication**

#### **Description**

The DSNUsingOSAuthentication property applies only when an 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**

#### **Description**

The EnableBaseDimSelfJoin property specifies whether the 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), Campaign assumes that you do not want to perform a join.

#### **Default value**

FALSE

### **EnableSelectDistinct**

#### **Description**

The EnableSelectDistinct property specifies whether the internal lists of IDs for Campaign are de-duplicated by the 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 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 Campaign application server to perform de-duplication to reduce resource consumption/burden on the database.

Regardless of what value you specify for this property, 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 Campaign server).

#### Default value

TRUE

#### Valid Values

TRUE | FALSE

### EnableSelectOrderBy

#### Description

The EnableSelectOrderBy property specifies whether the internal lists of IDs for Campaign are sorted by the 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 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

#### Description

The ExcludeFromTableDisplay parameter allows you to limit the database tables that are displayed during table mapping in Campaign. It does not reduce the number of table names retrieved from the database.

Table names matching the specified patterns are not displayed.

For example, if you set the value of this parameter to sys.\*, tables with names that begin with sys. are not displayed. Note that the values for this parameter are case-sensitive.

#### Default value

UAC\_\*, which excludes temp tables and Extract tables, when the ExtractTablePrefix property's value is the default value

### ExtractTablePostExecutionSQL

#### 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.

*Table 50. Tokens available to ExtractTablePostExecutionSQL*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

#### **Default value**

Not defined

#### **Valid Values**

A valid SQL statement

### **ExtractTablePrefix**

#### **Description**

The ExtractTablePrefix property specifies a string that is automatically prepended to all Extract table names in 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**

#### **Description**

The ForceNumeric property specifies whether Campaign retrieves numeric values as the data type double. When the value is set to TRUE, Campaign retrieves all numeric values as the data type double.

#### **Default value**

FALSE

**Valid Values**

TRUE | FALSE

**InactiveConnectionTimeout****Description**

The InactiveConnectionTimeout property specifies the number of seconds an inactive 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****Description**

The InsertLogSize property specifies when a new entry is entered in the log file while the 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****Description**

The JndiName property is used only when configuring the Campaign system tables (not for other data sources, such as customer tables). Set its value to the Java Naming and Directory Interface (JNDI) data source that is defined in the application server (WebSphere or WebLogic).

**Default value**

campaignPartition1DS

**LoaderCommand****Description**

The LoaderCommand property specifies the command issued to invoke your database load utility in Campaign. If you set this parameter, 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 parameter also invokes the database loader utility mode when 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.

Contact Optimization uses the same configuration settings as Campaign for implementing a database load utility. If you configure Campaign to work with a database load utility, you configure Contact Optimization to use the same commands. Similarly, if you configure Contact Optimization to work with a database load utility, you are configuring Campaign to work with a database load utility. Each assumes a different root directory in your IBM EMM installation directory. The root directory is /Campaign for Campaign or /ContactOptimization for Contact Optimization, so you can specify different commands in the loader command and template files.

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. 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 parameter is undefined by default.

Tokens available to LoaderCommand are described below.

*Table 51. Tokens available to LoaderCommand*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign generates according to the template that is specified in the LoaderControlFileTemplate parameter.
<DATABASE>	This token is replaced with the name of the data source that 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 Campaign during the loading process. This file is in the Campaign Temp directory, UNICA_ACTMPDIR.
<DBUSER>	This token is replaced with the database user name for the database.
<DSN>	This token is replaced with the value of the DSN property. If the DSN 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 51. Tokens available to LoaderCommand (continued)

Token	Description
<TABLE>	This token is obsolete, but is supported for compatibility with earlier versions. See <TABLENAME>, which replaced <TABLE> as of version 4.6.3.
<TABLENAME>	This token is replaced with the database table name that Campaign is loading data into. This is the target table from your Snapshot process or the name of the Temp Table being created by 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

### Description

The LoaderCommandForAppend parameter specifies the command issued to invoke your database load utility for appending records to a database table in Campaign. If you set this parameter, Campaign enters database loader utility mode for all output files from the Snapshot process that are used with the "append records" settings.

This parameter 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 parameter is undefined by default.

Tokens available to LoaderCommandForAppend are described below.

Table 52. Tokens available to LoaderCommandForAppend

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.

Table 52. Tokens available to LoaderCommandForAppend (continued)

Token	Description
<CONTROLFILE>	This token is replaced with the full path and file name to the temporary control file that Campaign generates according to the template that is specified in the LoaderControlFileTemplate parameter.
<DATABASE>	This token is replaced with the name of the data source that 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 Campaign during the loading process. This file is in the Campaign Temp directory, UNICA_ACTMPDIR.
<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 DSN 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 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, but is supported for compatibility with earlier versions. See <TABLENAME>, which replaced <TABLE> as of version 4.6.3.
<TABLENAME>	This token is replaced with the database table name that Campaign is loading data into. This is the target table from your Snapshot process or the name of the Temp Table being created by 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

#### Description

The LoaderControlFileTemplate property specifies the full path and file name to the control file template configured in Campaign. When this parameter is set, Campaign dynamically builds a temporary control file based on the template that you specify here. The path and name of this temporary control file is available to the <CONTROLFILE> token that is available to the LoaderCommand parameter.

Before you use 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 Campaign.

For the correct syntax required for your control file, see your database loader utility documentation.

This parameter 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.

*Table 53. Tokens available to LoaderControlFileTemplate*

Token	Description
<DBCOLUMNNUMBER>	This token is replaced with the column ordinal in the database.
<FIELDLENGTH>	This token is replaced with the length of the field being loaded into the database.
<FIELDNAME>	This token is replaced with the name of the field being loaded into the database.
<FIELDNUMBER>	This token is replaced with the number of the field being loaded into the database.
<FIELDTYPE>	This token is replaced with the literal "CHAR( )". The length of this field is specified between the (). If your database happens to not understand the field type, CHAR, you can manually specify the appropriate text for the field type and use the <FIELDLENGTH> token. For example, for SQLSVR and SQL2000 you would use "SQLCHAR(<FIELDLENGTH>)"
<NATIVETYPE>	This token is replaced with the actual database type that this field is loaded into.
<xyz>	This token places the specified character(s) on all fields being loaded into the database, except the last. A typical use is <,> which repeats a comma for all fields except the last.
<~xyz>	This token places the specified characters only on the last repeated line.
<!xyz>	This token places the specified character(s), including the angle brackets < >, on all lines.

#### Default value

No default value defined.

### LoaderControlFileTemplateForAppend

#### Description

The `LoaderControlFileTemplateForAppend` property specifies the full path and file name to the control file template configured in Campaign. When this parameter is set, Campaign dynamically builds a temporary control file based on the template that is specified here. 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 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 Campaign.

See your database loader utility documentation for the correct syntax required for your control file. Tokens available to your control file template are the same as those for the LoaderControlFileTemplate property.

This parameter is undefined by default.

#### **Default value**

No default value defined.

### **LoaderDelimiter**

#### **Description**

The LoaderDelimiter property specifies whether the temporary data file is a fixed-width or delimited flat file, and, if it is delimited, the characters Campaign uses as delimiters.

If the value is undefined, 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. 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 can be enclosed in double quotation marks, if wanted.

### **LoaderDelimiterAtEnd**

#### **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, Campaign uses delimiters at the end of each line.

FALSE

#### **Default value**

FALSE

#### **Valid Values**

TRUE | FALSE

### **LoaderDelimiterAtEndForAppend**

#### **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, Campaign uses delimiters at the end of each line.

**Default value**

FALSE

**Valid Values**

TRUE | FALSE

**LoaderDelimiterForAppend****Description**

The LoaderDelimiterForAppend property specifies whether the temporary 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, 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. 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.

**LoaderUseLocaleDP****Description**

The LoaderUseLocaleDP property specifies, when 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****Description**

Allows you to specify the maximum number of items that 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**

### **Description**

The MaxQueryThreads property specifies the upper limit on the number of simultaneous queries allowed to run against each database source from a single Campaign flowchart.

Campaign runs database queries using independent threads. Because 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 value specified by this property, the Campaign server automatically limits the number of simultaneous queries to this value.

The maximum value is unlimited. Note that when the maxReuseThreads property 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**

### **Description**

When the selected number of IDs is less than the value specified by the MaxRowFetchRecords property, Campaign passes the IDs to the database, one at a time in a separate SQL query. This process may be very time-consuming. If the number of selected IDs is greater than the value specified by this parameter, Campaign uses temporary tables (if allowed against the database source), or it pulls down all the values from the table, not including any unnecessary values.

For performance reasons, it is best to keep this number low.

### **Default value**

100

## **MaxTempTableJoinPctSelectAll**

### **Description**

When a query is issued, 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, 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****Description**

When a query is issued, 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**

**Description**

Use this property to set the threshold for using the bulk loader. Campaign invokes the script assigned to the `LoaderCommand` 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, 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, Campaign assumes that the value is zero.

**Default value**

0 (zero)

**Valid Values**

Integers

**MinReqForLoaderCommandForAppend**

**Description**

Use this property to set the threshold for using the bulk loader. 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, 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, Campaign assumes that the value is zero.

**Default value**

0 (zero)

**Valid Values**

Positive integers

## NumberOfRetries

### Description

The NumberOfRetries property specifies the number of times Campaign automatically retries a database operation on failure. Campaign automatically resubmits queries to the database this number of times before reporting a database error or failure.

### Default value

0 (zero)

## ODBCTableTypes

### Description

This property is empty by default, which is appropriate for all currently supported data sources.

### Default value

Not defined

### Valid Values

(empty)

## ODBCUnicode

### Description

The ODBCUnicode property specifies the type of encoding used in 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 - Campaign uses ANSI ODBC calls.
- UTF-8 - Campaign uses Unicode ODBC calls and assumes that a SQLWCHAR is a single byte. This is compatible with DataDirect ODBC drivers.
- UCS-2 - Campaign uses Unicode ODBC calls and assumes that a SQLWCHAR is 2 bytes. This is compatible with Windows and unixODBC ODBC drivers.

## ODBCv2

### Description

Use the ODBCv2 property to specify which ODBC API specification Campaign should use for the data source.

The default value of FALSE allows Campaign to use the v3 API specification, while a setting of TRUE causes 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, 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**

**Description**

Use this property to limit the table mapping display in IBM 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**

**Description**

When set to TRUE, the PadTextWithSpaces property causes 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**

**Description**

Use the PostExtractTableCreateRunScript property to specify a script or executable for Campaign to run after an Extract table has been created and populated.

Tokens available to PostExtractTableCreateRunScript are described below.

*Table 54. Tokens available to PostExtractTableCreateRunScript*

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the Extract table was created.
<AMUSER>	This token is replaced with the IBM EMM 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.

Table 54. Tokens available to PostExtractTableCreateRunScript (continued)

Token	Description
<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

#### Description

Specifies a script or executable that Campaign runs after a Segment temp table has been created and populated.

Tokens available to PostSegmentTableCreateRunScript are described below.

Table 55. Tokens available to PostSegmentTableCreateRunScript

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the Segment temp table was created.
<AMUSER>	This token is replaced with the IBM EMM 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

### Description

Use the PostSnapshotTableCreateRunScript property to specify a script or executable that Campaign runs after a Snapshot table has been created and populated.

Tokens available to PostSnapshotTableCreateRunScript are described below.

Table 56. Tokens available to PostSnapshotTableCreateRunScript

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 IBM EMM 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

### Description

Use the PostTempTableCreateRunScript property to specify a script or executable for 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.

Table 57. Tokens available to PostTempTableCreateRunScript

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 IBM EMM user name associated with the flowchart for which temp tables were created.

Table 57. Tokens available to PostTempTableCreateRunScript (continued)

Token	Description
<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

#### Description

Specifies a script or executable that Campaign runs after a User table has been created and populated.

Tokens available to PostUserTableCreateRunScript are described below.

Table 58. Tokens available to PostUserTableCreateRunScript

Token	Description
<DBUSER>	This token is replaced with the database user name for the database where the User table was created.
<AMUSER>	This token is replaced with the IBM EMM 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

### Description

Use the PrefixOnSelectSQL property to specify a string that is automatically prepended to all SELECT SQL expressions generated by Campaign.

This property applies only to SQL generated by 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.

*Table 59. Tokens available to PrefixOnSelectSQL*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

### Default value

No default value defined.

## QueryThreadSleep

### Description

The QueryThreadSleep property affects the CPU utilization of the Campaign server process (UNICA\_ACSVR). When the value is TRUE, the thread that the Campaign server process uses to check for query completion sleeps between checks. When the value is FALSE, the Campaign server process checks continuously for query completion.

### Default value

TRUE

## ReaderLogSize

### Description

The ReaderLogSize parameter defines when 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

**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

**ShareConnection**

**Description**

The ShareConnection property is no longer used and should remain set to its default value, FALSE.

**Default value**

FALSE

**Valid Values**

FALSE

**SQLOnConnect**

**Description**

The SQLOnConnect property defines a complete SQL statement that 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.

*Table 60. Tokens available to SQLOnConnect*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.

Table 60. Tokens available to SQLOnConnect (continued)

Token	Description
<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 Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

### StringEncoding

#### Description

The StringEncoding property specifies the character encoding of the database. When Campaign retrieves data from the database, the data is transcoded from the encoding specified to the internal encoding of Campaign (UTF-8). When Campaign sends a query to the database, character data is transcoded from the internal encoding of 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 Campaign* in the *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 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 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 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 Campaign supports only per-data source encoding. UTF-8 cannot be used with 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

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 Campaign to access non-English data within a database.

In the 7.0 version of Campaign, a new database encoding called WIDEUTF-8 was introduced as a value for the StringEncoding property. By using this encoding, 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

### Description

The SuffixOnAllOtherSQL property specifies a string that is automatically appended to every SQL expression, generated by Campaign, which are not covered by the SuffixOnInsertSQL, SuffixOnSelectSQL, SuffixOnTempTableCreation, SuffixOnUserTableCreation, or SuffixOnUserBaseTableCreation properties.

This property applies only to SQL generated by 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 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.

Table 61. Tokens available to SuffixOnAllOtherSQL

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.

Table 61. Tokens available to SuffixOnAllOtherSQL (continued)

Token	Description
<USER>	This token is replaced with the Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

### SuffixOnCreateDateField

#### Description

The SuffixOnCreateDateField property specifies a string that 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.

### SuffixOnInsertSQL

#### Description

The SuffixOnInsertSQL property specifies a string that is automatically appended to all INSERT SQL expressions generated by Campaign. This property applies only to SQL generated by 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 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.

Table 62. Tokens available to SuffixOnInsertSQL

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.



Table 62. Tokens available to SuffixOnInsertSQL (continued)

Token	Description
<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 Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

### SuffixOnSelectSQL

#### Description

The SuffixOnSelectSQL property specifies a string that is automatically appended to all SELECT SQL expressions generated by Campaign. This property applies only to SQL generated by 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.

Table 63. Tokens available to SuffixOnSelectSQL

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

## SuffixOnTempTableCreation

### Description

Use the SuffixOnTempTableCreation property to specify a string that is automatically appended to the SQL expression generated by Campaign when a temp table is created. This property applies only to SQL generated by 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.

Table 64. Tokens available to SuffixOnTempTableCreation

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

### Default value

No default value defined.

## SuffixOnSegmentTableCreation

### Description

Specifies a string that is automatically appended to the SQL expression generated by Campaign when a Segment temp table is created.

Tokens available to `SuffixOnSegmentTableCreation` are described below.

*Table 65. Tokens available to `SuffixOnSegmentTableCreation`*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

**Default value**

Not defined

**Valid Values**

Valid SQL

**SuffixOnSnapshotTableCreation**

**Description**

Use the `SuffixOnSnapshotTableCreation` property to specify a string that is automatically appended to the SQL expression generated by Campaign when a Snapshot table is created.

Tokens available to `SuffixOnSnapshotTableCreation` are described below.

*Table 66. Tokens available to `SuffixOnSnapshotTableCreation`*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.

Table 66. Tokens available to SuffixOnSnapshotTableCreation (continued)

Token	Description
<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 Campaign user name of the user running the flowchart.

#### Default value

Not defined

#### Valid Values

Valid SQL

### SuffixOnExtractTableCreation

#### Description

Use the SuffixOnExtractTableCreation property to specify a string that is automatically appended to the SQL expression generated by Campaign when an Extract table is created.

Tokens available to SuffixOnExtractTableCreation are described below.

Table 67. Tokens available to SuffixOnExtractTableCreation

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

#### Default value

Not defined

#### Valid Values

Valid SQL

## SuffixOnUserBaseTableCreation

### Description

Use the SuffixOnUserBaseTableCreation property to specify a string that is automatically appended to the SQL expression that Campaign generates when a user creates a Base table (for example, in an Extract process). This property applies only to SQL generated by 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.

*Table 68. Tokens available to SuffixOnUserBaseTableCreation*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

### Default value

No default value defined.

## SuffixOnUserTableCreation

### Description

Use the SuffixOnUserTableCreation property to specify a string that is automatically appended to the SQL expression that Campaign generates when a user creates a General table (for example, in a Snapshot process). This property applies only to SQL generated by 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.

*Table 69. Tokens available to `SuffixOnUserTableCreation`*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.
<TABLENAME>	This token is replaced with the temp table name.

#### Default value

No default value defined.

### SystemTableSchema

#### Description

Specifies the schema used for Campaign system tables.

The default value is blank. This parameter is only relevant for the `UA_SYSTEM_TABLES` data source.

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` (for example, `dbo.UA_Folder`). In this form, `X` is the schema and `Y` is the unqualified table name. This terminology for this syntax differs among the different database systems supported by Campaign.)

If multiple schemas exist in the system tables database, then set this value to the name of the schema in which the Campaign system tables were created.

#### Default value

No default value defined.

### TempTablePostExecuteSQL

#### Description

Use the TempTablePostExecuteSQL property to specify a complete SQL statement that Campaign runs immediately after the creation of a temporary table in a user data source or in the system tables database. The AllowTempTables property must be set to TRUE to enable the creation of temp tables in a data source.

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.

The TempTablePostExecuteSQL 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 the TempTablePostExecuteSQL property, be sure that you 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\;`

Tokens available to TempTablePostExecuteSQL are described below.

*Table 70. Tokens available to TempTablePostExecuteSQL*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

#### TableListSQL

#### 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, 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 (\*), 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 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
```



## UOSQLOnConnect

### Description

The SQLOnConnect property defines a complete SQL statement that Campaign runs immediately after each database connection. The UOSQLOnConnect property is similar to this, but specifically applicable to Contact Optimization.

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.

*Table 71. Tokens available to UOSQLOnConnect*

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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 Campaign user name of the user running the flowchart.

### Default value

No default value defined.

## UseSQLToRetrieveSchema

### Description

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 for this property is FALSE, indicating that Campaign should use its standard method (ODBC or native connection, for example) to retrieve the schema. Setting this property to TRUE causes 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 Campaign.

The following list describes the behavior of this setting for a number of data sources:

- For Netezza, you must set this property to TRUE to allow support for synonyms. Setting this property to TRUE tells Campaign to prepare a SQL query to retrieve the table schema. No other settings or values are needed to support synonyms in Netezza data sources.
- For SQL Server, to allow support for synonyms you must 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.
- For Oracle data sources, setting this property to TRUE tells 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 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 Campaign to work with Netezza or SQL Server synonyms:

UseSQLToRetrieveSchema=TRUE

### UserBaseTablePostExecuteSQL

#### 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 **UserTablePostExecuteSQL**.

## UserTablePostExecuteSQL

### Description

Use this property to specify a complete SQL statement that IBM 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.

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 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\;

Tokens available to **UserTablePostExecuteSQL** are described below.

Table 72. Tokens available to UserTablePostExecuteSQL

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.
<KEYCOLUMNS>	This token is replaced with the user table column name(s).

Table 72. Tokens available to UserTablePostExecuteSQL (continued)

Token	Description
<TABLENAME>	This token is replaced with the user table name.
<USER>	This token is replaced with the Campaign user name of the user running the flowchart.

#### Default value

No default value defined.

### UseTempTablePool

#### Description

WhenUseTempTablePool 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 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

### SegmentTablePostExecuteSQL

#### Description

Use the SegmentTablePostExecuteSQL property to specify a complete SQL statement that Campaign runs after a Segment temp table has been created and populated.

Tokens available to SegmentTablePostExecuteSQL are described below.

Table 73. Tokens available to SegmentTablePostExecuteSQL

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.

Table 73. Tokens available to SegmentTablePostExecuteSQL (continued)

Token	Description
<USER>	This token is replaced with the Campaign user name of the user running the flowchart.

**Default value**

Not defined

**Valid Values**

A valid SQL statement

**SnapshotTablePostExecuteSQL**

**Description**

Use the SnapshotTablePostExecuteSQL 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 SnapshotTablePostExecuteSQL are described below.

Table 74. Tokens available to SnapshotTablePostExecuteSQL

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.
<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 Campaign user name of the user running the flowchart.

**Default value**

Not defined

**Valid Values**

A valid SQL statement

**TempTablePrefix**

**Description**

This property specifies a string that is automatically prepended to the names of all temporary tables created by 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, Campaign prepends the letter "U" to the prefix.

Token	Description
<AMUSER>	This token is replaced with the IBM EMM 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.

Token	Description
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<USER>	This token is replaced with the Campaign user name of the user running the flowchart.

#### Default value

UAC

### TempTablePreTruncateExecutionSQL

#### Description

**Note:** This property is supported only for Teradata data sources. For all other supported databases, this property should not be set.

Use the TempTablePreTruncateExecutionSQL 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 TempTablePostExecuteSQL property.

For example, with the TempTablePostExecuteSQL 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

#### Description

**Note:** This property is supported only for Teradata data sources. For all other supported databases, this property should not be set.

Use the TempTablePreTruncateRunScript 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****Description**

The TeradataDeleteBeforeDrop parameter applies only to Teradata data sources. It specifies whether records are deleted before a table is dropped.

Set this value to TRUE to delete all records from a table before dropping the table.

**Note:** If Campaign is unable to delete the records for any reason, it will not drop the table.

Set this value to FALSE to drop a table without first deleting all records.

**Default value**

TRUE

**TruncateSQL****Description**

The TruncateSQL 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 set to TRUE. When DeleteAsTruncate is set to TRUE, any custom SQL in this property is used to truncate a table. When this property is not set, Campaign uses the TRUNCATE TABLE <TABLENAME> syntax.

This parameter is undefined by default.

Tokens available to TruncateSQL are described below.

*Table 75. Tokens available to TruncateSQL*

Token	Description
<TABLENAME>	This token is replaced with the database table name that Campaign is truncating.

**Default value**

No default value defined.

**Type****Description**

The partitions > partition[n] > dataSources > [data\_source\_name] > type property specifies the database type of this data source.

**Default value**

The default value depends on the database template used to create the data source configuration.

**Valid Values**



Valid values for system tables are:

- SQLServer
- DB2
- DB2ODBC
- ORACLE
- ORACLE8
- ORACLE9

Valid values for customer tables also include:

- TERADATA
- NETEZZA

## **UseExceptForMerge**

### **Description**

When 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 set to TRUE and we 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)
```

### **Default value**

FALSE

### **Valid Values**

TRUE | FALSE

## **UseMergeForTrack**

### **Description**

Implements SQL MERGE syntax to improve the performance of the Track process. The UseMergeForTrack 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

### 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

### Description

When 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 (value set to YES), 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

NO

### Valid Values

YES | NO

## UseSQLToProfile

### Description

The UseSQLToProfile property allows you to configure 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 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 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

## **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 | systemCodes**

Properties in this category specify, for 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**

#### **Description**

The offerCodeDelimiter property is used internally to concatenate multiple code parts (for example, to output the OfferCode field in Campaign Generated Fields) and for incoming offer codes in the Campaign Response process, to split the offer code into multiple parts. The value must be only a single character.

Note that in this version of Campaign, the NumberOfOfferCodesToUse parameter no longer exists. This value now comes from the offer template (every offer template can have a different number of offer codes).

#### **Default value**

-

### **allowVariableLengthCodes**

#### **Description**

The allowVariableLengthCodes property specifies whether variable length codes are allowed in 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**

#### **Description**

The displayOfferCodes property specifies whether to show offer codes beside their names in the 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**

### **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**

### **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 as follows:

- 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 Campaign substituted for the x with any ASCII character, and Campaign will use that character instead.

## **cellCodeGenProgFile**

### **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 Campaign)

## **campCodeGenProgFile**

### **Description**

The `campCodeGenProgFile` property specifies the name of the campaign code generator.

The properties that control the format of the code generated 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, and 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 Campaign)

### 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 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 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 *Campaign Administrator's Guide* for a list of supported encodings.

#### Default value

No default value defined.

## **forceDCTOneBytePerChar**

### **Description**

The `forceDCTOneBytePerChar` property specifies whether 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, 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)`, 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 Campaign flowchart waits, after the user has disconnected and all runs have completed, before exiting, and the Campaign server process waits for a response from external servers before reporting an error.

## **waitForGracefulDisconnect**

### **Description**

The `waitForGracefulDisconnect` property specifies whether the 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 Campaign server process waits for a response from external servers.

Currently, this applies to requests to IBM EMM servers and eMessage components that operate with Campaign.

If the 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 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 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 IBM Distributed Marketing.

**collaborateInactivityTimeout**

**Description**

The `collaborateInactivityTimeout` property specifies the number of seconds the `unica_acsvr` process waits after it finishes servicing a Distributed Marketing request before it exits. This waiting period allows the process to remain available in the typical scenario in which Distributed Marketing 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

**Campaign | partitions | partition[n] | server | permissions**

The properties in this category specify the permissions set on folders created by Campaign, and the UNIX group and permissions set on files contained in the **profile** directory.

**userFileGroup (UNIX only)**

**Description**

The `userFileGroup` property specifies a group associated with user-generated 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 Campaign on the 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 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 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 Campaign files (for example, log files, summary files, exported flat files).

### **Default value**

666 (everyone can read and write files created by 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 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 Campaign Snapshot process can have duplicate values.

If the value is `FALSE`, the Campaign server enforces unique cell codes.

If the value is `TRUE`, the 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 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 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 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 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`, 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 Campaign 7.2 and earlier.

When set to `FALSE`, Campaign replaces only distinct user variable and UCGF names with actual values (in both IBM EMM and raw SQL expressions). Set this property to `FALSE` for backwards compatibility with 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`. 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 Campaign versions should continue to work. However, if old flowcharts fail because of the change in the naming convention, you can revert Campaign behavior by setting this property to `TRUE`.

**Default value**

`FALSE`

**Valid Values**

`TRUE` | `FALSE`

**Campaign | partitions | partition[n] | server | flowchartSave**

The properties in this category specify the default settings for a new Campaign flowchart's auto-save and checkpoint properties.

**checkpointFrequency**

**Description**

The `checkpointFrequency` property specifies (in minutes) the default setting for a new 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 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 the this category specify how 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 Campaign macro language will treat numeric IDs longer than 15 digits as text.

Set the value to `TRUE` to specify that numeric IDs longer than 15 digits will be treated as text.

Set the value to `FALSE` to specify that numeric IDs longer than 15 digits are treated as numeric values (and thus might lose precision or uniqueness if truncated or rounded).

**Note:** 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 `stringConcatWithNullIsNull` property controls the behavior of the 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 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`, Campaign converts all data to upper case before comparing.

When the value is `LOWER`, 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 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 Campaign data comparisons are sensitive to alphabetic case (UPPER vs. lower).

When the value is `FALSE`, 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`, 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 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 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 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 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 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 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. 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. 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 Campaign server optimization for partitions.

**Note:** This category of parameters is not related to Contact Optimization.

### maxVirtualMemory

#### Description



This property specifies a default value for the maximum amount of system virtual memory to use when running a 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.

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 =  $(80\% \times 32) / 10$  = approximately 2.5 GB / 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, and change the **IBM Campaign Virtual Memory Usage** value on the Server Optimization tab.

### useInDbOptimization

#### Description

The useInDbOptimization property specifies whether Campaign tries to perform as many operations as possible in the database instead of in the Campaign server.

If the value is FALSE, Campaign maintains lists of IDs in the Campaign server at all times.

If the value is TRUE, Campaign avoids pulling the ID lists if possible.

#### Default value

FALSE

#### Valid Values

TRUE | FALSE

### maxReuseThreads

#### Description

The maxReuseThreads 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 as this property is set to 0.

Use the cache to reduce the overhead of thread allocation or if your operating system exhibits 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**

### **Description**

The threadStackSize determines the number of bytes allocated for each thread's stack. Do not change this property except under guidance from IBM. The minimum value is 128 K. The maximum value is 8 MB.

### **Default value**

1048576

## **tempTableDataSourcesForSegments**

### **Description**

The tempTableDataSourcesForSegments 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**

### **Description**

To improve performance for strategic segments, set this option to TRUE. When this option is set to TRUE, strategic segments do not create binary files on the Campaign server; instead, strategic segments create Segment temp tables in the data source. When the value is set to TRUE, at least one valid Temp Table data source must be specified in the CreateSeg process configuration.

### **Default value**

FALSE

### **Valid Values**

TRUE | FALSE

## **forceViewForPreOptDates**

### **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

## **Campaign | partitions | partition[n] | server | logging**

Properties in this category specify, for the Campaign server, whether standard and Windows event logging are enabled, logging levels and categories, and other logging behavior.

### **enableWindowsEventLogging**

#### **Description**

The enableWindowsEventLogging property enables or disables 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. If disabled, the windowsEventLoggingLevel and windowsEventLoggingCategory settings are ignored.

#### **Default value**

FALSE

#### **Valid Values**

TRUE | FALSE

### **logFileBufferSize**

#### **Description**

The logFileBufferSize property is used when the value of the keepFlowchartLogOpen property is yes. It sets an upper bound on the number of log messages after which the messages will be written to file.

If the value is 1, every log message is written immediately to file, effectively disabling buffering but causing somewhat worse performance.

This property is ignored if the value of keepFlowchartLogOpen is set to no.

#### **Default value**

5

### **keepFlowchartLogOpen**

#### **Description**

The keepFlowchartLogOpen property specifies whether Campaign opens and closes the flowchart log file each time a line is written to the log file.

If the value is FALSE, Campaign opens and closes the flowchart log file.

If the value is TRUE, Campaign opens the flowchart log file only once, and closes the flowchart log file only when the flowchart's server process exits. A value of yes may improve performance of real-time flowcharts. A side effect of using the yes setting is that recently-logged messages may not be immediately visible in the log file, as 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.

#### **Default value**

FALSE

#### **Valid Values**

TRUE | FALSE

## **logProcessId**

### **Description**

The logProcessId property controls whether the process ID (pid) of the Campaign Server process is logged 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**

### **Description**

The logMaxBackupIndex property specifies the number of backup 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**

### **Description**

The loggingCategories property specifies the category of messages written to the Campaign server log file. This works in conjunction with loggingLevels, which determines which messages are logged based on severity (for all selected categories). You can specify multiple categories in a comma-separated list. The special category all provides a shorthand for specifying all logging categories.

### **Default value**

ALL

### **Valid Values**

Supported categories are:

- 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

## loggingLevels

### Description

The loggingLevels property controls the amount of detail written to the Campaign server log file, based on severity.

### Default value

MEDIUM

### Valid Values

- LOW
- MEDIUM
- HIGH
- ALL

LOW represents the least detail (the most severe errors only), and ALL includes trace messages and is intended primarily for diagnostic purposes. You can adjust these settings from within a flowchart through the Tools >Logging Options menu.

**Note:** You may want to set the loggingLevels property to ALL during configuration and testing, to maximize the logging output from Campaign for diagnostic purposes. This setting generates a large amount of data and therefore may not be advisable for production operation.

## windowsEventLoggingCategories

### Description

The windowsEventLoggingCategories property specifies the category of messages written to the Campaign server windows event log. This 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 special 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**

### **Description**

The logFileMaxSize property specifies the maximum size, in bytes, that the Campaign server log file is allowed to reach before being rolled over to backup files.

### **Default value**

10485760 (10 MB)

## **windowsEventLoggingLevels**

### **Description**

The windowsEventLoggingLevels property controls the amount of detail written to the Campaign server windows event log based on severity.

### **Default value**

MEDIUM

### **Valid Values**

- LOW
- MEDIUM
- HIGH
- ALL

LOW represents the least detail (the most severe errors only), and ALL includes trace messages and is intended primarily for diagnostic purposes.

## **enableLogging**

### **Description**

The enableLogging property specifies whether 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****Description**

The AllowCustomLogPath property allows user to change the log path for each campaign flowchart that generates flowchart-specific logging information when it is run.

A setting of False prevents the user from changing the path to which the flowchart log file is written.

A setting of True enables the user to change the path through the user interface or when using unica\_svradm to run the flowchart.

**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 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.

**maxDataErrorsAllowed****Description**

The maxDataErrorsAllowed property specifies the maximum number of data conversion errors allowed in an Campaign Snapshot export.

**Default value**

0 (zero), which allows no errors

**saveRunResults****Description**

This property lets you save 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.

IBM 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 Campaign test run. A value of 0 (zero) removes the limitation on the number of IDs.

**Default value**

0 (zero)

**Campaign | partitions | partition[n] | server | profile**

Properties in this category specify the maximum number of categories created during profiling for numeric and text values in Campaign.

**profileMaxTextCategories**

**Description**

The profileMaxTextCategories and profileMaxNumberCategories properties specify the maximum number of categories created in 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 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 Campaign partition. If your Campaign installation has multiple partitions, set these properties for each partition that you want to affect.

### **internalIdLowerLimit**

#### **Description**

The `internalIdUpperLimit` and `internalIdLowerLimit` properties constrain the Campaign internal IDs to be within the specified range. Note that the values are inclusive: that is, Campaign may use both the lower and upper limit.

#### **Default value**

0 (zero)

### **internalIdUpperLimit**

#### **Description**

The `internalIdUpperLimit` and `internalIdLowerLimit` properties constrain the Campaign internal IDs to be within the specified range. The values are inclusive: that is, Campaign may use both the lower and upper limit. If Distributed Marketing is installed, set the value to 2147483647.

#### **Default value**

4294967295

### **eMessageInstalled**

#### **Description**

Indicates that eMessage is installed. When you select Yes, eMessage features are available in the Campaign interface.

The IBM installer sets this property to Yes for the default partition in your eMessage installation. For additional partitions where you installed eMessage, you must configure this property manually.

#### **Default value**

No

#### **Valid Values**

Yes | No

### **interactInstalled**

#### **Description**

After installing the Interact design environment, this configuration property should be set to Yes to enable the Interact design environment in Campaign.

If Interact is not installed, set to No. Setting this property to No does not remove Interact menus and options from the user interface. To remove menus and options, you must manually unregister Interact using the `configTool` utility.

#### **Default value**

No

**Valid Values**

Yes | No

**Availability**

This property is applicable only if you installed Interact.

**MO\_UC\_integration**

**Description**

Enables integration with Marketing Operations for this partition, if the integration is enabled in the **Platform** configuration settings. For more information, see the *IBM Marketing Operations and Campaign Integration Guide*.

**Default value**

No

**Valid Values**

Yes | No

**MO\_UC\_BottomUpTargetCells**

**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 *IBM Marketing Operations and Campaign Integration Guide*.

**Default value**

No

**Valid Values**

Yes | No

**Legacy\_campaigns**

**Description**

For this partition, enables access to campaigns created before Marketing Operations and Campaign were integrated. Applies only if **MO\_UC\_integration** is set to Yes. Legacy campaigns also include campaigns created in Campaign 7.x and linked to Plan 7.x projects. For more information, see the *IBM Marketing Operations and Campaign Integration Guide*.

**Default value**

No

**Valid Values**

Yes | No

**IBM Marketing Operations - Offer integration**

**Description**

Enables the ability to use Marketing Operations 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 *IBM Marketing Operations and Campaign Integration Guide*.

**Default value**

No

**Valid Values**

Yes | No

**UC\_CM\_integration**

**Description**

Enables Digital Analytics online segment integration for a 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 > Campaign | partitions | partition[n] | Coremetrics**.

**Default value**

No

**Valid Values**

Yes | No

**Campaign | partitions | partition[n] | server | fileDialog**

Properties in this category specify the default directories for Campaign input and output data files.

**defaultOutputDirectory**

**Description**

The `defaultOutputDirectory` property specifies the path used to initialize the Campaign File Selection dialog. The `defaultOutputDirectory` property is used when an output data file is mapped into 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 Campaign File Selection dialog. The `defaultInputDirectory` property is used when an input data file is mapped into 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 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 Campaign. By default, the `ExecutableCodeGenerator` (shipped with Campaign) uses this property to indicate the path (relative to 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 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), 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, Campaign assumes it is the name of a jar file that contains the Java class.

If the path is relative, Campaign assumes it is relative to the Campaign application home directory.

#### **Default value**

codeGenerator.jar (packaged in the Campaign.war file)

## **Campaign | partitions | partition[n] | Coremetrics**

Properties in this category specify integration settings for Digital Analytics and Campaign for the selected Campaign partition. If your 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 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**

Campaign uses the CoreMetricsKey to map IDs exported from Digital Analytics to the corresponding Audience ID in 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 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, 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 **IBM 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**

This property specifies which IBM EMM account is allowed to access the Digital Analytics integration service. See below for additional information.

If no value is specified, 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**

This property identifies the data source assigned to the Marketing Platform account specified in the **ASMUserForCredentials** setting. The default is UC\_CM\_ACCESS. This "data source for credentials" is the mechanism that Marketing 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 *IBM Marketing Platform Administrator's Guide*

**Default value**

UC\_CM\_ACCESS

## 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 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 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 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 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 Campaign JMX connector server for Interact. Campaign has no JMX security.

If set to FALSE, you cannot connect to the Campaign JMX connector server.

This JMX monitoring is for the Interact contact and response history module only.

### **Default value**

FALSE

**Valid Values**

TRUE | FALSE

**Availability**

This property is applicable only if you have installed Interact.

**protocol**

**Description**

Listening protocol for the Campaign JMX connector server, if `monitorEnabledForInteract` is set to yes.

This JMX monitoring is for the Interact contact and response history module only.

**Default value**

JMXMP

**Valid Values**

JMXMP | RMI

**Availability**

This property is applicable only if you have installed Interact.

**port**

**Description**

Listening port for the Campaign JMX connector server, if `monitorEnabledForInteract` is set to yes.

This JMX monitoring is for the 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 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 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 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 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 Campaign server is started.

**Default value**

3600 (60 hours)

## Campaign | unicaACLListener

The **Campaign | unicaACLListener** properties specify logging levels, some access privileges, language encodings, number of operating system threads, and the protocol, host, and port of the Campaign listener. These properties must be set only once per instance of Campaign; they do not need to be set for every partition.

**enableWindowsImpersonation**

**Description**

The `enableWindowsImpersonation` property specifies whether Windows impersonation is enabled in 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**

**Description**

The Campaign > unicaACListener > enableWindowsEventLogging property controls logging to the Windows event log. Set this property to TRUE to log to the Windows event log.

**Default value**

FALSE

**Valid Values**

TRUE | FALSE

**serverHost**

**Description**

The serverHost property specifies the name or IP address of the machine where the Campaign listener is installed. If the Campaign listener is not installed on the same machine where IBM EMM is installed, change the value to the machine name or IP address of the machine where the Campaign listener is installed.

**Default value**

localhost

**logMaxBackupIndex**

**Description**

The logMaxBackupIndex property specifies how many backup files can exist before the oldest one is deleted. If you set this property to 0 (zero), 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, 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**

**Description**

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 Campaign" in the *Campaign Administrator's Guide*.

## systemStringEncoding

### Description

The `systemStringEncoding` property indicates which encodings 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 Campaign* in the *Campaign Administrator's Guide*.

## loggingLevels

### Description

The `Campaign > unicaACLlistener > loggingLevels` property controls the amount of detail written to the log file.

### Default value

`MEDIUM`

### Valid Values

- `LOW`
- `MEDIUM`
- `HIGH`

## maxReuseThreads

### Description

The `Campaign > unicaACLlistener > maxReuseThreads` property sets the number of operating system threads cached by the 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.

### Default value

`0` (zero), which disables the cache

## logMaxFileSize

### Description

The `logMaxFileSize` property specifies the maximum size, in bytes, that the log file can reach before rolling into the backup file.

### Default value

`10485760` (10 MB)

## **windowsEventLoggingLevels**

### **Description**

The windowsEventLoggingLevels property controls the amount of detail written to the Windows event log file based on severity.

### **Default value**

MEDIUM

### **Valid Values**

- LOW
- MEDIUM
- HIGH
- ALL

The ALL level includes trace messages intended for diagnostic purposes.

## **serverPort**

### **Description**

The serverPort property specifies the port where the Campaign listener is installed.

### **Default value**

4664

## **useSSL**

### **Description**

The useSSL property specifies whether to use Secure Sockets Layer for communications between the Campaign listener and the Campaign web application.

Also see the description for the serverPort2 property, in this category.

### **Default value**

no

### **Valid Values**

yes | no

## **serverPort2**

### **Description**

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 Campaign listener and flowchart processes, separately from the communication between the Campaign web application and listener, which is specified by the serverPort and useSSL properties in this category.

All communication between 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 Campaign listener and the flowchart (server) processes and communication between the listener and the 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 Campaign web application always uses the communication mode specified by useSSL when communicating to the listener.

When SSL is enabled for communication between the Campaign listener and flowchart processes, set the value of this property (serverPort2) to an appropriate port.

#### **Default value**

0

### **useSSLForPort2**

#### **Description**

See the description for the serverPort2 property, in this category.

#### **Default value**

FALSE

#### **Valid Values**

TRUE, FALSE

### **keepalive**

#### **Description**

Use the keepalive property to specify, in seconds, the frequency with which the Campaign web application server sends keep alive messages on otherwise-inactive socket connections to the 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

## 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 Contact Optimization 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 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 the Marketing Platform is integrated with the IBM Tivoli web access control platform, you must set this property to the Campaign URL in Tivoli.

### **Default value**

No default value defined.

## **Campaign | logging**

The property in this category specifies the location of the Campaign log properties file.

## **log4jConfig**

### **Description**

The log4jConfig property specifies the location of the Campaign log properties file, campaign\_log4j.properties. Specify the path relative to the Campaignhome directory, including the file name. Use forward slashes (/) for UNIX and backslashes (\) for Windows.

### **Default value**

./conf/campaign\_log4j.properties





---

## Appendix B. Special characters in Campaign object names

Names of objects in Campaign can have specific requirements. Some special characters are not supported in any 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

Do not use any of the characters listed in the following table in the names of these objects:

- campaigns
- flowcharts
- folders
- offers
- offer lists
- segments
- sessions

*Table 76. 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
Tab	Tab

---

### Objects with no naming restrictions

The following objects in Campaign have no restrictions for characters used in their names:

- audience levels (audience level *field* names have naming restrictions)

- custom attribute *display* names (custom attribute *internal* names have naming restrictions)
- offer templates

---

## Objects with specific naming restrictions

The following objects in Campaign have specific restrictions on their names:

- Custom attribute *internal* names (custom attribute *display* names have no naming restrictions)
- Audience level *field* names (audience level names have no naming restrictions)
- 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, Campaign supports all the characters that are supported by the string encoding configured.

**Note:** Derived field names have additional restrictions. For details, see “Naming restrictions for derived fields.”

### 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 Appendix B, “Special characters in Campaign object names,” on page 329

---

## Appendix C. Supported data types for user tables

This topic lists the data types that Campaign supports for user tables created in each supported database. Any other data types not listed here are unsupported. Before mapping user tables in Campaign, ensure that your tables use only supported data types.

**Note:** A column of datatype DATE, DATETIME, or TIMESTAMP of a table, when mapped in an IBM 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.

### DB2

bigint

char

date

decimal

double

float

int

numeric

real

smallint

timestamp

varchar

### 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.

## Oracle

DATE

FLOAT (p)

NUMBER [ (p , s) ] [1]

TIMESTAMP

VARCHAR2(size BYTE)

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

## SQL Server

bigint

bit

char(n) [1]

datetime

decimal

float

int

nchar [2]

numeric

nvarchar [2]

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.

## **Teradata**

bigint

byteint

char

date

decimal

float

int

numeric

smallint

timestamp

varchar



---

## Appendix D. Internationalization and encodings

This section provides information on character encodings and language-sensitive database considerations, and lists the encodings supported by Campaign and PredictiveInsight.

---

### Character encodings in Campaign

This topic describes the encodings that Campaign supports.

On most operating systems, Campaign uses the GNU iconv library. Note that **IBM does not ship an iconv for AIX installations**. For an AIX system, you must obtain the appropriate character sets. For a list, see the National Language Support Guide and Reference:

- [http://moka.ccr.jussieu.fr/doc\\_link/en\\_US/a\\_doc\\_lib/aixprgdd/nlsgrdf/iconv.htm#d722e3a267mela](http://moka.ccr.jussieu.fr/doc_link/en_US/a_doc_lib/aixprgdd/nlsgrdf/iconv.htm#d722e3a267mela)
- <http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.nls/doc/nlsgrdf/nlsgrdf.htm>

This section lists the encodings that are supported by Campaign. The values in these lists are valid values for setting the Campaign internationalization parameters listed in “Set Campaign language and locale property values” on page 177. 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 Campaign configuration parameters to any of the values in a group, depending on how your system uses the encodings.
- When setting values for the 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 *Marketing Platform Administrator's Guide*.
- 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 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 CSIS087JISX0208
- ISO-IR-159 JIS\_X0212 JIS\_X0212-1990 JIS\_X0212.1990-0 X0212 CSIS0159JISX02121990
- 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-R0 JP CSIS014JISC6220R0
- CP932
- ISO-2022-JP CSIS02022JP
- ISO-2022-JP-1
- ISO-2022-JP-2 CSIS02022JP2

## 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 CSIS02022KR

## 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 VISCII1.1-1 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 Campaign for multiple locales, you can set the values for `DateFormat` and `DateTimeFormat` configuration parameters for any of the formats specified in the DATE macro, as shown in the following table.

However, **if you need to configure 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 174.

Table 77. 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
MMDDYYYY	2-digit month, 2-digit day, and 4-digit year	March 31, 1970 is 03311970
DELIM_M_D For <code>DateTimeFormat</code> , use DT_DELIM_M_D	Any delimited month followed by day	March 31, 3/31, or 03-31
DELIM_M_D_Y For <code>DateTimeFormat</code> , use DT_DELIM_M_D_Y	Any delimited month, day, and year	March 31, 1970 or 3/31/70
DELIM_Y_M For <code>DateTimeFormat</code> , use DT_DELIM_Y_M	Any delimited year followed by month	1970 March, 70-3, 1970/3
DELIM_Y_M_D For <code>DateTimeFormat</code> , 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

Table 77. Date formats (continued)

Format	Description	Example(s)
YYMM	2-digit year and 2-digit month	7003
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
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

Table 77. Date formats (continued)

Format	Description	Example(s)
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
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 Campaign for multiple locales, you can set the values for DateOutputFormat and DateTimeOutputFormatconfiguration parameters for any of the formats specified specified for format\_str in the DATE\_FORMAT macro, as shown in the following table.

However, **if you need to configure 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 174.

%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).





---

## Appendix E. Campaign error codes

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.

Campaign notifies users of error events when they occur with error messages that consist of a code number and error text.

If you see an error message that says you have invalid user access rights, you may not have the correct privileges assigned in Marketing Platform to perform that action. For more information, see the *Marketing Platform Administrator's Guide*.

If you encounter errors while using Campaign, try reading the descriptions and applying the solutions in this section before contacting IBM Technical Support. If an error does not appear here, or if a solution fails, contact your administrator or contact IBM Technical Support.

---

### Campaign error codes

The following table lists the error messages generated by Campaign.

Table 78. 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).
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).

Table 78. Campaign error codes (continued)

Code	Error Description
328	Matrix multiply dimension mismatch.
329	Matrix multiply dimension too big.
330	Singular matrix error.
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.
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.
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'

Table 78. Campaign error codes (continued)

Code	Error Description
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).
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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>[Campaign could not transcode a non-ASCII file name.]</p> <p>[Campaign could not locate the specified file.]</p> <p>[Campaign cannot open the file properly.]</p> <p>[Campaign could not transcode a non_ASCII file name.]</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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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> <p>[Your browser cannot connect to the 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 Campaign administrator to check that the Campaign server machine does not use the port assigned to the 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 Campaign server machine.]</p>
1713	Unable to send socket data.

Table 78. Campaign error codes (continued)

Code	Error Description
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>[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 Campaign administrator to check that the Campaign server machine does not use the port assigned to the Campaign listener process for another application. [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 Campaign server machine.]</p>
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 Campaign 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	<p>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 Campaign administrator to restart the process that ended abnormally. If the error occurs again, ask your system administrator for assistance.</p>
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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 Campaign listener cannot start the Campaign server process.</p> <p>Suggested solutions: Ask your system administrator for assistance.</p>
1787	Maximum number of flowchart instances is already in use.
1788	Requested flowchart is active for Distributed Marketing.
1789	Requested flowchart is in use by an 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 Marketing Platform.]</p> <p>[The user name or password field for accessing the database or other object is empty in the Marketing Platform]</p> <p>[The user name or password field for accessing the database or other object is empty in the Marketing Platform.]</p> <p>Suggested solutions:</p> <p>[Check that the user name and password you provided are correct.]</p> <p>[Ask your Campaign administrator to check that your user names and passwords stored in the Marketing 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
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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 Marketing Platform before re-attempting to log into Campaign.
2014	Web server authentication failure. You must provide a valid web server username and password in a data source named "webserver\" in the Marketing Platform before re-attempting to log into Campaign.
2015	HTTP request error after PAC file authentication failure.
2016	HTTP request error after PAC file scheme failure.
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.
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.



Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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.
10152	Internal Error: Invalid Segment (Data file name is blank).

Table 78. Campaign error codes (continued)

Code	Error Description
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 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.
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
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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 IBM EMM.
10401	INTERNAL ERROR: Client already connected.
10402	Client is not connected to a server.
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 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 Campaign administrator to check that the:]</p> <ul style="list-style-type: none"> <li>• [Campaign listener process is running.]</li> <li>• [same version of the Campaign web application, listener, and server are running on your system.]</li> <li>• port number is properly configured in the central configuration repository provided by Marketing 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.
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

Table 78. Campaign error codes (continued)

Code	Error Description
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 Campaign server is busy processing another request.</p> <p>Suggested solutions</p> <p>Ask your system administrator to check that 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.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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 eMessage Server
10582	eMessage Server: unknown error
10583	eMessage Server: Internal System Error
10584	eMessage server url is not set.
10585	Internal Error: parsing data returned from eMessage Server
10586	Error returned from eMessage 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.

Table 78. Campaign error codes (continued)

Code	Error Description
10613	Field not found.  Possible causes:  [The field no longer exists because of table mapping change.]  [The audience level changed.]  [The field was deleted.]  Suggested solutions: Re-configure the process box to reference a different field.
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 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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
10807	Illegal Custom Macro name.
10808	Name conflicts with an existing IBM 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.
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 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.
11122	Unable to read unica_tbmgr.bin file in conf directory.
11128	Operation disallowed in configuration.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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.



Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
11300	Invalid field name. Look at the end of message for invalid field name.  Possible causes:  [The field no longer exists because of table mapping change.]  [The audience level changed.]  [The field was deleted.]  Suggested solutions: Re-configure the process box to reference a different field.  Invalid Field Name=
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.
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	Unable to create lock file in conf directory.  Possible causes: The Campaign server cannot lock the dummy_lock.dat file  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 Campaign administrator to reboot the Campaign server to remove the lock.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
11327	ID of new version conflicts with existing versions.
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.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
11365	Raw SQL Custom Macro returning only ID list is not allowed in this context.
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.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
11403	Invalid or missing audience field name.
11404	Duplicate audience field name.
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.
11500	Internal Error: Not a valid table in the database.
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.
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).

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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?

Table 78. Campaign error codes (continued)

Code	Error Description
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 Campaign system tables.You will be switched to Summary tab now.
12031	Cannot proceed without connection to Campaign system tables.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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 iscomplete 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 iscomplete flag, but it was missing.
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.
13125	Invalid parameter value.
13131	Database authentication required.
13132	String conversion error.



Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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 78. 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.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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
14001	Model process internal error.
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 78. 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.
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.
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

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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
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 78. 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
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 Campaign Table index
16404	Internal Error
16405	Internal Error: New Table Unknown function
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

Table 78. Campaign error codes (continued)

Code	Error Description
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
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.
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?

Table 78. Campaign error codes (continued)

Code	Error Description
16709	Syntax check OK
16710	Discard changes to current expression?
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
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 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?
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?

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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.
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	Campaign Data Folder Identifier is empty.
17015	Campaign Data Folder Path is empty.

Table 78. Campaign error codes (continued)

Code	Error Description
17016	Duplicate Identifiers in 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.
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
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 Contact Optimization session.
17315	DB source is not defined in selected Contact Optimization session.
17316	Required field is missing from proposed contact table.

Table 78. Campaign error codes (continued)

Code	Error Description
17317	Selected Contact Optimization session is currently running.
17318	Database authentication required.
17319	No Contact Optimization 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 Contact Optimization session was running.
17332	Failed while trying to write to the proposed attribute table.
17333	One or more export field is unmapped.
17334	Tried to delete optimize process box while associated Contact Optimization 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
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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.
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
17540	Invalid Offer ID in eMessage Document
17541	Empty audience level.
17542	No offers are selected.
17544	Cell is missing offer assignment.
17549	Error returned from eMessage server during run.
17550	Internal error: unknown eMessage 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.

Table 78. Campaign error codes (continued)

Code	Error Description
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 eMessage server.
17560	Duplicate tracking codes not allowed.
17561	Cannot determine tracking audience level.
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	eMessage directory is invalid.
17574	Missing field assignment for content type.
17575	eMessage is still completing the last operation. Please try again later.
17576	No eMessage document selected.
17577	Unknown parameter.
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 eMessage document.



Table 78. Campaign error codes (continued)

Code	Error Description
17592	Missing or retired Offer or Offer List detected.
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.
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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?
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.
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.

Table 78. Campaign error codes (continued)

Code	Error Description
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
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.
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 Contact Optimization session not found.
20000	Internal Error: Unknown function tag.
20002	DOM exception
20003	Pipe open error
20004	Offer code not unique

Table 78. Campaign error codes (continued)

Code	Error Description
20005	End date specified is earlier than start date
20006	Invalid Report name
20007	Invalid Attribute name
20008	Offer is used. Cannot delete.
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.

---

## Contacting IBM technical support

If you encounter a problem that you cannot resolve by consulting the documentation, your company's designated support contact can log a call with IBM technical support. To ensure that your problem is resolved efficiently and successfully, you collect information before you log your call.

If you are not a designated support contact at your company, contact your IBM administrator for information.

### Information to gather

Before you contact IBM technical support, gather the following information:

- A brief description of the nature of your issue.
- Detailed error messages that you see when the issue occurs.
- Detailed steps to reproduce the issue.
- Related log files, session files, configuration files, and data files.
- Information about your product and system environment, which you can obtain as described in "System information."

### System information

When you call IBM technical support, you might be asked to provide information about your environment.

If your problem does not prevent you from logging in, much of this information is available on the About page, which provides information about your IBM applications.

You can access the About page by selecting **Help > About**. If the About page is not accessible, you can obtain the version number of any IBM application by viewing the `version.txt` file that is located under the installation directory for each application.

### Contact information for IBM technical support

For ways to contact IBM technical support, see the IBM Product Technical Support website: ([http://www.ibm.com/support/entry/portal/open\\_service\\_request](http://www.ibm.com/support/entry/portal/open_service_request)).

**Note:** To enter a support request, you must log in with an IBM account. If possible, this account must be linked to your IBM customer number. To learn more about associating your account with your IBM customer number, see **Support Resources > Entitled Software Support** on the Support Portal.



---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
170 Tracer Lane  
Waltham, MA 02451  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not



been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Privacy Policy and Terms of Use Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. A cookie is a piece of data that a web site can send to your browser, which may then be stored on your computer as a tag that identifies your computer. In many cases, no personal information is collected by these cookies. If a Software Offering you are using enables you to collect personal information through cookies and similar technologies, we inform you about the specifics below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name, and other personal information for purposes of session management, enhanced user usability, or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

Various jurisdictions regulate the collection of personal information through cookies and similar technologies. If the configurations deployed for this Software Offering provide you as customer the ability to collect personal information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for providing notice and consent where appropriate.

IBM requires that Clients (1) provide a clear and conspicuous link to Customer's website terms of use (e.g. privacy policy) which includes a link to IBM's and Client's data collection and use practices, (2) notify that cookies and clear gifs/web beacons are being placed on the visitor's computer by IBM on the Client's behalf along with an explanation of the purpose of such technology, and (3) to the extent required by law, obtain consent from website visitors prior to the placement of cookies and clear gifs/web beacons placed by Client or IBM on Client's behalf on website visitor's devices

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Online Privacy Statement at: <http://www.ibm.com/privacy/details/us/en> section entitled "Cookies, Web Beacons and Other Technologies."







Printed in USA