Unica Plan V12.1.3 Administrator's Guide

Contents

Chapter 1. Overview....1

Unica Plan in a clustered environment....1

Unica Plan with LDAP integration....1

Optional features....3

Unica Plan documentation and help....3

Users and user groups....6

Synchronizing users....7

Unica Plan administration process overview....8

Administrative Settings page....8

Setting your timezone....14

System-wide non-work dates....14

Adding non-work dates....15

Removing non-work dates....15

Changing the list of non-working business dates....16

Summary of Non-working Days Changes page....16

Triggers....17

Adding triggers....18

Adding and editing trigger bindings....18

Trigger binding dialog....19

Trigger events....21

Publishing advanced searches....26

Publishing or unpublishing saved searches....27

Viewing a list of locked objects....27

Unica Plan Performance....28
Unica Plan markup options....29
Markup tool availability....29
Enabling Adobe™ Acrobat markup....30
Enabling native Unica Plan markup....30
Disabling markup....31
Configuring system logs....31
Integrating Unica Content Integration....37
Chapter 2. Unica Plan interface customization options....39
Renaming marketing object types....39
The sysmodules.xml file....40
The sysmenu.xml file....43
Renaming a marketing object type....44
Customizing menus....45
Creating a menu....45
Reorganizing items on menus....46
Renaming a menu or an item on a menu....46
Adding a menu item that links to a URL....47
Synchronizing menus....47
Multiple locale support....48
Chapter 3. Unica Plan integrations....49
Unica Plan - Unica Campaign integration....49
Unica Plan - Unica Centralized Offer Management integration....49
Unica Plan - Unica Journey integration....50
Chapter 4. Marketing object types....51

Defining a custom marketing object type.... 53
States for marketing objects.... 53
The global states file.... 54
State transitions.... 55
Custom marketing object types.... 56
Adding a marketing object type.... 56
The Add marketing object Type page.... 57
Marketing object type definition files.... 59
Marketing object type properties files.... 60
Marketing object type changes.... 60
Marketing object templates.... 61
Creating a marketing object template.... 61
Associations between marketing object types.... 61
Chapter 5. Project and request deactivation.... 63
Deactivating projects and requests.... 64
Reactivating projects and requests.... 65
Deactivating projects and requests in bulk.... 65
Reactivating projects and requests in bulk.... 67
Chapter 6. Unica Plan reports.... 70
Unica Plan report and folder names in IBM Cognos®.... 70
Unica Plan reports and IBM Cognos®.... 71
Data model updates in IBM Cognos®.... 72
Updating the Unica Plan data model.... 72
Example Query Subject for a custom metric.... 73
Filters in IBM Cognos® reports.... 74

Hyperlinks in IBM Cognos® reports....75
Example custom report: Project Performance Summary (Custom)....76
Reports for Unica Plan administrators....80
Producing the Out of Office Users and Delegation Summary....81
Producing a Deny Reason Analysis Report....81
Chapter 7. Template Basics....83
When to create another template....83
Decide on a set of templates....84
Example templates....84
List of example templates....85
Template components....85
Template-building methodology....89
Unica Plan custom templates....90
What can you customize?....91
Sample Summary tab....92
Custom tab example....93
Unica Campaign project templates....94
Unica Campaign project template design....94
Offer templates....94
Chapter 8. Adding or editing templates....96
Building and managing templates....98
Template tabs reference....102
Effects of template changes....104
Template Properties tab for defining the template....104
Attributes tab....108

Exporting a single template....109
Template Tabs tab for customizing the user interface....110
Adding tabs to templates....112
Moving tabs and forms on templates....113
Deleting custom forms and tabs from templates....113
Building rules to show and hide forms....114
Rule Builder dialog....115
Template Attachments tab for adding folders and files....117
Template Custom Links tab for accessing other websites....118
Template Customize Alerts tab to set up event-triggered alerts....120
Icons page....120
Adding and editing icons....121
Template validation....122
Database validation....122
Attribute validation....123
Chapter 9. Program and project templates....124
Budget tab customization....124
Adding text columns to the Line Items Details table....125
Disabling or removing text columns....125
Vendor column for budgets....125
Template Budget Approval Rules tab for automating line-item approvals....126
Building budget approval rules....127
Editing budget approval rules....128
Deleting budget approval rules....129
Project template Project Roles tab for including participants....129

Project template Request tab for configuring the request process....130
Request tab fields....131
Example: building a template request rule....135
Project template Workflow tab....137
Workflow template tab....138
Adding workflow to templates....139
Configuring the workflow....140
Fields in workflow spreadsheets....142
Considerations for importing and exporting form tasks in workflow templates....145
Approval process and options....146
Customize milestone types....150
Creating and editing workflow templates....151
Workflow template branching....151
Branching the workflow template....152
Importing workflow templates into projects....153
Considerations for importing and exporting form tasks in workflow templates....154
Importing workflow templates....155
Exporting workflow templates....156
Project template Campaign tab for communicating with Unica Campaign....157
TCS® approval....159
Data Mapping Definitions....159
Mapping Unica Campaign contact and response counts to Unica Plan metrics....160
Metrics data mapping file....161

Adding data mapping files....162
Editing data mapping files....162
Deleting data mapping files....163
Chapter 10. Forms....164
Form Definitions page....164
Creating forms....166
Creating a form....166
Importing shared attributes....167
Form Editor interface....167
Attribute groups....169
Creating grids....171
Grid types....171
Creating an editable grid....172
Create New Grid dialog....173
Display an existing editable grid as a read-only grid....175
Display a grid as a list....176
Example: creating lists of marketing objects....179
Target cell spreadsheets....183
Creating a Target Cell Spreadsheet....184
Default cell attributes for Target Cell Spreadsheets....186
Publishing forms....188
Publishing a form....188
Exporting forms....188
Exporting a form....189
Importing forms....189

Importing a form..... 190
Troubleshooting when importing forms..... 190
Moving forms from one computer to another..... 191
Managing lookup values for a form..... 191
Disabling lookup values without changing the database table..... 192
Copying forms..... 192
Enabling data posting of list selections..... 193
Adding a form to existing objects..... 194
Chapter 11. Attributes..... 196
Standard attributes..... 198
Attributes for Plan-Campaign integration..... 199
Unica Campaign attributes..... 199
Cell attributes..... 200
Offer attributes..... 200
Creating, editing, and deleting attributes..... 200
Creating and enabling a shared attribute..... 201
Editing a shared attribute..... 201
Deleting a shared attribute..... 201
Shared Attributes list page..... 202
Creating a local attribute..... 203
Editing a local attribute..... 203
Deleting a local attribute..... 204
Attributes reference..... 204
Standard attribute fields..... 204
Database considerations for attribute database columns..... 208

Attribute types list....208
Campaign, cell, and offer attributes....211
Text attribute types....212
Rich text type....213
Single-Select attribute types....214
Single-Select - Database attribute types....215
Multiple-Select - Database attribute types....217
Yes or No attribute types....218
Decimal attribute types....219
Money attribute types....220
Calculated attribute types....220
URL Field attribute types....222
Object Reference attribute types....223
Image attribute types....225
Creative URL attribute types....226
Object Attribute Field Reference attribute types....226
Single List Object Reference attribute types....227
Dependent field attribute types....229
Visible attributes and attribute groups based on other attributes....231
Marking attributes and attribute groups visible based on other attributes....234
Chapter 12. Metrics....237
Types of metrics....237
Metric creation overview....239
Working with metrics, metrics dimensions, and metrics templates....240
Metrics properties....240

Metrics dimensions properties.... 242
Creating metrics templates and metric template groups.... 243
Creating or editing a metrics template.... 244
Metrics groups.... 245
Exporting and importing metrics templates.... 245
Chapter 13. Multiple locale support.... 247
Localized object types.... 247
Localized format and symbol settings.... 249
Multi-locale support for templates.... 249
Localizing templates through the properties file.... 251
Globalizing standard attributes.... 251
Localizing forms.... 253
Localizing forms.... 255
Localizing metrics.... 256
Importing a metrics properties file.... 257
Localizing lists.... 257
Chapter 14. Security.... 258
Global security policy.... 258
Roles.... 259
Default security policy roles.... 259
Custom security policy roles.... 260
Object access roles.... 260
Project roles.... 262
Security policies and permissions.... 262
Permissions.... 263

Planning security policies....265
Configure security policies....266
Editing the Global security policy....266
Creating a security policy....267
Configuring the user visibility option for roles....268
Assigning security policy roles....269
Controlling access to templates....270
Access controls for projects and requests....270
Example security configuration for project requests....271
Chapter 15. Setting up alerts....275
Event-triggered alerts....276
Reminders....276
Set default alert subscriptions....277
Setting default alert subscriptions....279
Default Alert Subscription page....280
Reminders and workflow task dates....280
Customizing notification messages for alerts....281
Customizing an alert notification message....282
Alerts Settings page....283
Example of a custom alert message....286
Changing the refresh interval for the alerts count....287
How Unica Plan determines the alert sender....287
Setting up the collaboration tool....288
Chapter 16. List definitions....289
Customizable lists....289

Adding options to a list.... 291
List properties.... 292
Enabling, disabling, or removing a list option.... 293
Chapter 17. Implementing project health rules.... 295
Key performance indicators.... 296
Default health rule.... 297
Configuring project health status rules.... 299
Assigning a rule to a project template.... 300
Customizing labels and colors.... 301
Chapter 18. Exporting and importing metadata.... 302
Exporting metadata.... 302
Exporting metadata in bulk.... 303
Importing metadata.... 306
Importing template metadata.... 307
Importing metadata.... 309
Chapter 19. Setting up libraries for digital assets.... 313
Creating a library.... 313
Enabled and disabled libraries.... 314
Chapter 20. Setting up accounts.... 315
Account administrators.... 315
Account owners.... 316
Creating an account.... 317
Account Properties page.... 318
Adding or remove account owners.... 319
Enabled and disabled accounts.... 320

Enabling or disabling an account....321
Chapter 21. Advanced topics....322
Populating fields programmatically....322
Examples of programmatically populating fields....323
Server-side ID generation and project attribute validation....325
Sample Java™ interface....326
Grid validation....337
Validator interface....338
Data validation rules....338
Validation rules....339
Chapter 22. Unica Plan REST API....345
Contents of the Unica Plan REST API....348
REST API interfaces....348
REST API exceptions....349
REST API handles....349
REST API AttributeMap....353
Configuring the Unica Plan REST API in Unica Platform....357
Changes after the Unica Platform SSO integration....359
Use Swagger User Interface as Rest API Client....360
Unica Plan REST API....
Index....

Chapter 1. Overview

With Unica Plan, you organize the people, tasks, and budget for marketing programs, while you reduce costs and time to market.

Unica Plan is a web-based application.

System requirements

Hardware, operating system, web server, and database requirements are described in the Recommended Software Environments and Minimum System Requirements document.

Client computers

Configure client computers to meet the following requirements.

- Internet Explorer must check for newer versions of stored pages every time users see the web page. For example, in Internet Explorer 9, select **Tools > Internet options** and in the **Browsing history** section click **Settings**. Select **Every time I visit the webpage**.
- If the client computer uses software to block pop-up ad windows, Unica Plan may not function properly. Allow pop-up ad windows for the Unica Plan URL (for example, `http://myMachine:7001/plan`).

Unica Plan in a clustered environment

If you are running Unica Plan in a clustered environment, do all system administration tasks on one instance.

To do system administration tasks in a clustered environment, shut down all but one instance of Unica Plan.

Unica Plan with LDAP integration

A set of files exist in the location `<plan_home>/conf/<locale>/ldap_conf.xml`. These files contain the LDAP server settings. These settings are used in any form attribute of type external data source.

In the following code, replace the bold-and-italicized variables with actual values:

```
<!-- url for the ldap server.  example: ldap://192.168.100.39:389
 -->
      <ldap_provider_url>SERVER-URL</ldap_provider_url>
      <!-- account to connect to the server -->
      <ldap_security_principal>ACCOUNT</ldap_security_principal>
      <!-- password -->
      <ldap_credentials>PASSWORD</ldap_credentials>
.
.
.
<!-- one or more ldap references for identifying the objects of interest
 -->
      <ldap_references>
         <ldap_reference>
            <!-- example:
 ou=Users,ou=US,dc=singlesignon1,dc=singlesignon1,dc=com -->
            <ldap_name>
            <![CDATA[\{*}NAME-STRING\{*}]]>
            </ldap_name>
             <ldap_filter>
            <![CDATA[(objectclass=user)]]>
            </ldap_filter>
         </ldap_reference>
       </ldap_references>
```

The account format must either of the following formats:

Format	Example
`<login>@<domain>`	`johndoe@unicaindia`
`<domain>\<login>`	`unicaindia\johndoe`

Format	Example
`cn=<login>,CN=Users,<base dn>`	`cn=johndoe,CN=Users,dc=unicaindi-a,dc=com`

The NAME-STRING must have the following format:

```
ou=Users,ou=US,dc=singlesignon1,dc=singlesignon1,dc=com
```

Optional features

Unica Plan offers several options that extend the core, out-of-box features. Your organization can integrate Unica Plan with other HCL products, or enable add-on modules.

To extend Unica Plan features, your organization can add the following products and features.

- Integrate Unica Campaign with Unica Plan.
- Enable optional offer integration when Unica Plan and Unica Campaign are integrated.
- Integrate IBM Digital Recommendations with Unica Plan.
- Enable the Financial Management module, which supports accounting activities for your plans, programs, and projects.
- Enable the Digital Asset Management module, which supports centralized storage and management of electronic files.
- Enable the Integration Services module, which includes the application programming interface (API) for extending Unica Plan.

These options are licensed separately. As a result, they are not operational at every organization.

Unica Plan documentation and help

Different people in your organization use Unica Plan to accomplish different tasks. Information about Unica Plan is available in a set of guides, each of which is intended for use by team members with specific objectives and skill sets.

The following table describes the information available in each guide.

Table 1. Guides in the Unica Plan documentation set

The following three-column table describes tasks in one column, guide names in the second column, and audience in the third column.

If you	See	Audience
• Plan and manage projects • Establish workflow tasks, milestones, and personnel • Track project expenses • Get reviews and approvals for content • Produce reports • Create to-dos and checklists	Unica Plan User's Guide	• Project managers • Creative designers • Direct mail marketing managers • Marketers
• Design templates, forms, attributes, and metrics • Customize the user interface • Define user access levels and security • Implement optional features • Configure and tune Unica Plan	Unica Plan Administrator's Guide	• Project managers • IT administrators • Implementation consultants

Table 1. Guides in the Unica Plan documentation set

The following three-column table describes tasks in one column, guide names in the second column, and audience in the third column.

(continued)

If you	See	Audience
• Create marketing campaigns • Plan offers • Implement integration between Unica Plan and Unica Campaign • Implement integration between Unica Plan and IBM Digital Recommendations	Unica Plan and Integration Guide	• Project managers • Marketing execution specialists • Direct marketing managers
• Learn about new system features • Research known issues and workarounds	Unica Plan Release Notes®	Everyone who uses Unica Plan
• Install Unica Plan • Configure Unica Plan • Upgrade to a new version of Unica Plan	Unica Plan Installation Guide	• Software implementation consultants • IT administrators • Database administrators
Create custom procedures to integrate Unica Plan with other applications	Unica Plan Integration Module and the API JavaDocs available when you click **Help > Product Documentation** in Unica Plan, and then download the `UnicaPlan<ver-`	• IT administrators • Database administrators • Implementation consultants

Table 1. Guides in the Unica Plan documentation set

The following three-column table describes tasks in one column, guide names in the second column, and audience in the third column.

(continued)

If you	See	Audience
	sion>PublicAPI.zip file for the SOAP API and UnicaPlan<*version*>PublicAPI-RestClient.zip for the REST API.	
Learn about the structure of the Unica Plan database	Unica Plan System Schema	Database administrators
Need more information while you work	• Get help and search or browse the *User's*, *Administrator's*, or *Installation* guides: Click **Help > Help for this page** • Access all of the Unica Plan guides: Click **Help > Product Documentation** • Access guides for all Unica products: Click **Help > All Unica Suite Documentation**	Everyone who uses Unica Plan

Users and user groups

You create and manage users and user groups with features installed with theUnica Platform

Before you work with other users, verify that the **Region setting** preference for your own Unica Plan user matches the default locale of your installation. The installation default locale is defined by the **defaultLocale** property under **Settings > Configuration > Unica Plan**. For details on creating users and user groups, setting user preferences, and assigning application access, see the Unica Platform Administrator's Guide.

After you create users, you assign them to a user group that has an appropriate level of access, such as PlanUserRole or PlanAdminRole. This process authorizes users to access Unica Plan.

After you create users and user groups, you must synchronize the Unica Plan user tables with the Unica Platform user tables. This step assigns the default Unica Plan security policy roles to the users.See Synchronizing users (on page 7).

If you do not see the users that you expect, or if users receive an error when they log in, make sure that the user group has application access rights to Unica Plan. Next, make sure that you synchronized the user tables.

Synchronizing users

Synchronizing users assures that system tables contain all of the user information that is entered through either Unica Platform or Unica Plan. You must synchronize users when you create new users.

Automated synchronization takes place on the schedule that is set by the **usermanagerSyncTime** property under **Settings > Configuration > Unica Plan > umoConfiguration**.

You can also manually synchronize the Unica Plan user tables with the Unica Platform user tables. This procedure enables new users to log in to Unica Plan and for changes to take effect before the next scheduled synchronization takes place.

Synchronizing users manually

You synchronize users between Unica Platform and Unica Plan. Do this task each time you add new users or to troubleshoot problems with users.

1. Log in to Unica Plan.
2. Click **Settings** > **Unica Plan Settings**.
3. Click **Synchronize Users**.

Unica Plan administration process overview

After you install Unica Plan, the administrator must configure the product and build objects, such as lists definitions and templates, before users can start working.

For best results, read the Unica Plan User's Guide to gain an understanding of the product before you set it up for users.

1. Install Unica Plan.
2. Configure Unica Plan.

   For more information about installing and configuring Unica Plan, see the Unica Plan Installation Guide.
3. Create templates. See Template Basics *(on page 83)*.
4. Set up a security policy and configure alerts. See Security *(on page 258)* and Setting up alerts *(on page 275)*.
5. Set up assets. See Setting up libraries for digital assets *(on page 313)*.
6. Set up list definitions. See List definitions *(on page 289)*.

Administrative Settings page

When you select **Settings > Unica Plan Settings**, the **Administrative Settings** page appears.

User Personalization section

Table 2. Administrative settings: User Personalization section

Section	Description
User Personal-ization	Contains links for users to customize Unica Plan to view and receive information that is important to them. See the *Unica Plan User's Guide* for details.

System Administration Settings section

Table 3. System Administration Settings section: Restricted Options

Link	Description
Security Policy Settings	Contains links to all security policies defined in your system. See Creating a security policy *(on page 267)* for details.
User Permissions	Lists all users authorized to use Unica Plan, organized by the groups to which they are assigned. See Assigning security policy roles *(on page 269)* for details.
Synchronize Users	Synchronizes the users in Unica Plan with the users in the Unica Platform. See Synchronizing users (on page 7) for details. When you synchronize users in a clustered environment, any changes propagate to the other servers when they next synchronize with Unica Platform.
Synchronize Menus	Synchronize the menus in the Unica Platform with the menus defined in Unica Plan.

Table 4. System Administration Settings section: Accessible Options

Link	Description
Default Alert Subscriptions	Opens a page for setting up and editing default alert subscriptions for Unica Plan objects. See Set default alert subscriptions *(on page 277)* for details.

Table 5. System Administration Settings section: Root-Level Object Definitions

Link	Description
Account Definitions	**Note:** This link is only available for systems that use the Financial Management module for Unica Plan.

Table 5. System Administration Settings section: Root-Level Object Definitions (continued)

Link	Description
	Contains links for managing Unica Plan accounts.
Budget Line Item Columns	Opens a page for adding text columns to the Line Items Details tables on Budget tabs.
Asset Library Definitions	Contains links for managing asset libraries. See Setting up libraries for digital assets *(on page 313)* for details.

Table 6. System Administration Settings section: Project Options

Link	Description
Health status	Click to customize the user interface labels and colors that are associated with the four possible project health statuses. See Customizing labels and colors *(on page 301)*.
Health rules	Click to view and manage the rules that are used to calculate the health status score for projects. You can add, edit, and delete rules, and assign the rule to project templates. See Configuring project health status rules *(on page 299)*.
Recalculate Project Health	Click to manually initiate recalculation of health status scores for all projects. See Implementing project health rules *(on page 295)*.
Project Health Daily Mails	Available to all users with the Plan User or Plan Administrator security policy role. Click to subscribe or unsubscribe yourself from project health status notifications. These messages are triggered by the automated daily process only, and are not sent if this process does not run for any reason. For more information, see the **Unica Plan > umoConfiguration > Scheduler > daily** property in the Unica Plan Installation Guide.

Table 6. System Administration Settings section: Project Options (continued)

Link	Description
IntraDay scheduler	Indicates the status of the batch process for periodically recalculating project health status scores. as ON or OFF. This setting can be either ON or OFF to indicate whether the **IntraDay scheduler** is running. For more information, see the **Unica Plan > umoConfiguration > Scheduler > intraDay** property in the Unica Plan Installation Guide.
Daily scheduler	Indicates the status of the daily batch process for calculating project health status scores. This setting can be either ON or OFF to indicate whether the **Daily scheduler** is running. For more information, see the **Unica Plan > umoConfiguration > Scheduler > daily** property in the Unica Plan Installation Guide.

Table 7. System Administration Settings section: Other Options

Link	Description
List Definitions	Contains links to the available lists where an administrator can populate or define list values. See Customizable lists *(on page 289)* for details.
Template Configuration	Contains links to features for working with templates and template components. See Building and managing templates (on page 98) for details. **Note:** In a clustered environment, you must shut down all but one server before you initiate any template configuration tasks.
Non-working Business Days	Opens a page for updating the set of system-wide non-work dates. See System-wide non-work dates *(on page 14)* for details.

Table 7. System Administration Settings section: Other Options (continued)

Link	Description
Manage Published Searches	Opens a page for publishing searches saved by Unica Plan users. See Publishing advanced searches *(on page 26)* for details.
Marketing Object Type Settings	Opens a page where you can view and manage custom marketing object types for your system. See Custom marketing object types (on page 56) for details. **Note:** In a clustered environment, you must shut down all but one server before you complete any marketing object configuration tasks.
Trigger Bindings	Opens a page listing triggers defined in the system and details about how they are bound to procedures. See Triggers *(on page 17)* for details.
Customize Alerts	Opens a page where you can change the text for alerts for specified events. See Setting up alerts *(on page 275)* for details.
Plan Upgrade	Opens a page where you can select Unica Plan components to upgrade. See the *Unica Plan Installation Guide* for details.
Data Migration	Offers options to export and import metadata. See Exporting and importing metadata *(on page 302)*.
Out of Office-Automatic Addition of Delegate	Provides access to a system-wide setting for the out of office feature. When a user is out of the office, a delegate can be specified to cover tasks, approvals, and requests. This setting defines whether a user can select only another team member as a delegate, or any Unica Plan user.

Table 7. System Administration Settings section: Other Options (continued)

Link	Description
	• Set **Enable Auto Addition of Delegate User** to yes to add the delegate as a project team member if necessary when a task, approval, or request is assigned. • Set **Enable Auto Addition of Delegate User** to no to require users to select only a user who is already a team member for all of the same projects as a delegate. You can override this setting for individual project templates. When the Out of Office settings are enabled, the current tasks that are ready to start are not automatically delegated to the user. The user who sets the Out of Office settings must manually delegate the ready to start tasks to the delegated user. For more information about the out of office feature, see the Unica Plan User's Guide.
Out of Office Users and Delegation Summary	Offers options to report information about team members who are out of the office. For more information, see Producing the Out of Office Users and Delegation Summary *(on page 81)*.
Associate Deny Reasons with Standalone Approvals	For stand-alone approvals, you can configure Unica Plan so that users who deny approval are required to select a reason from a predefined list. See Approval process and options *(on page 146)*.
Import Campaign Offers	Available only on systems that integrate Unica Plan with Unica Campaign and also enable optional offer integration. Enumerates the offers and offer templates, folders, and lists currently available in Unica Campaign. For more information about enabling offers, see Unica Plan and Unica Campaign Integration Guide.

Setting your timezone

You can set your timezone so you see Unica Plan dates and times in the locale time. You customize your timezone in the Unica Platform user settings.

Note: The default timezone is the server time.

1. Click **Settings > Users**.
2. Select your user name.

   Note: User names are listed in alphabetical order.

3. Click **Edit Preferences**.
4. Select Unica Platform.
5. Select you timezone from the list.
6. **Save** your changes.

You can decide whether to display users timezone information at the top of workflow pages and column headings in the **Calendar/Timeline** settings. You might choose to display timezone information in the workflow if many users from different timezones work on the same projects. You might choose not to display this information if project users work in the same timezone. For more information, see the Calender chapter in the *Unica Plan User's Guide.*

System-wide non-work dates

In Unica Plan, you can account for dates when, by default, no work is done on tasks. Unica Plan does not include non-work time in task duration calculations.

After you define one or more different types of non-working dates (such as "Holiday") you can enter specific non-work dates, singly or in date ranges. For more information about non-working date types, seeList definitions (on page 289).

Adding non-work dates

You can add non-work dates, such as holidays, on which no work is done.

When you add a non-work date, you must specify its type. To verify that the Non-working Date Types list contains the options that you need, click **Settings > Unica Plan Settings > List Definitions**. See List definitions *(on page 289)*.

1. Select **Settings > Unica Plan Settings**.
2. In the Other Options section, click **Non-working Business Days**.

   The **Non-working Business Days** page displays.
3. Enter a **Start Date** and an **End Date** for the new non-work time. The end date defaults to the start date, for easily entering single-date events.
4. Type a name for the non-work time in the **Name** field.
5. Select a type for the event from the **Type** list.
6. Click **Accept**.

   A message reminds you to click **Save changes** for your changes to take effect.

   Note: You cannot add past dates (that is, dates that already occurred).
7. Click **Save changes**.
   If your changes affect any projects, the system lists the affected projects with the names and email addresses of the project owners.
8. Click **Return to Previous Page** to add more non-working business days.

Removing non-work dates

You can remove dates on which no work is done, for example, if the holiday calendar changes.

1. Select **Settings > Unica Plan Settings**.
2. In the Other Options section, click **Non-working Business Days**.
3. Select a date or dates from the list.
4. Click **Remove**.

A message reminds you to click **Save changes** for your changes to take effect.

Note: You cannot delete past dates (that is, dates that already occurred).

5. Click **Save changes**.
   If your changes affect any projects, the system lists the affected projects with the names and email addresses of the project owners.
6. Click **Return to Previous Page** to remove more non-working business days.

Changing the list of non-working business dates

Typically, you define the list of holidays and other non-working dates at the beginning of the calendar year before marketing calendar details are set. However, you might need to change the set of non-working days after dates for tasks, projects, and programs are already set.

In this case, the process for changing the list of non-working business days is as follows.

1. Edit the list of non-working business days.
2. If your changes affect any projects or tasks, use the **Non-working Business Dates** summary page to notify affected project owners.

Summary of Non-working Days Changes page

If you change non-working dates, use the **Summary of Non-working Days Changes** page to send email notifications to project owners whose projects are affected.

The **Summary of Non-working Days Changes** page contains the following fields.

Table 8. Fields on the Summary of Non-working Days Changes page

Field	Description
Name	The name of the project owner of the affected project.
Email Address	The email address of the project owner of the affected project.

Table 8. Fields on the Summary of Non-working Days Changes page (continued)

Field	Description
Project list	A list of affected projects, and the actual non-working dates that were added or removed that occur during the duration of the project.

You can cut and paste the text in the **Project list** field into your email message, so the project owners can quickly assess the effect of your changes.

Triggers

Triggers can perform powerful custom validations and both pre- and post-event processing. Use triggers to integrate the out-of-box processing capabilities of Unica Plan with custom business processes.

You can set up triggers so that events related to certain objects cause execution of a procedure. When such event occurs, Unica Plan Procedure Manager runs the trigger.

For example, you want to insert data into a database whenever the state of a particular project changes from draft to active. To use a trigger to do insert data, you define the following.

- A procedure to insert a record into an external database table.
- A project template named DirectMail.
- A trigger binding for the DirectMail template, set to run when the state of a project changes, for example, from draft to active.

For projects that are based on the DirectMail template, the system calls the procedure when the specified state change occurs.

The following trigger rules apply:

- Triggers occur before or after an event.
- Triggers run on a subset of system events involving projects, requests, marketing objects, approvals, tasks, workflow spreadsheets, grid rows, users, invoices, budgets, accounts, and resources.

In general, define triggers at the most granular level possible. For example, rather than setting a trigger binding for any object, configure a trigger binding for a particular event for a particular project template.

You can view the list of trigger bindings for your installation by selecting **Settings > Unica Plan Settings** and clicking **Trigger Bindings**.

Adding triggers

You must complete several tasks before a trigger automatically runs a procedure when a certain condition is met for an object.

Note: You complete some of the tasks outside of Unica Plan.

1. Create a procedure in Java™ that implements the IProcedure interface.
   For more information, see the *Unica Plan Integration Module* guide.
2. Put the procedure into the procedure folder. You specify this directory with the **integrationProcedureDefinitionPath** property under **Settings > Configuration > Unica Plan > umoConfiguration > integrationServices**.
3. Build the procedure.
4. Add the procedure to the `procedure-plugins.xml` definition file, also in the directory that is defined by **integrationProcedureDefinitionPath**.
5. Restart the web server.
6. Log in to Unica Plan to add the trigger binding.

Adding and editing trigger bindings

You can modify triggers and create new ones to automatically respond to preset events.

1. Select **Settings > Unica Plan Settings > Trigger Bindings** to show the current trigger bindings in the system.
2. Click **Add new trigger binding**, or **Edit** for an existing binding.
   A dialog opens.
3. Provide an identifying **Name** for the trigger binding, then select the object type, template, context, and event.
   See Trigger binding dialog *(on page 19)*.
4. Click **Save** to save the binding and return to the **Trigger Bindings** list page.

Trigger binding dialog

When you create or edit a trigger binding, you specify information to define your trigger, such as template to filter. This information displays on the **Trigger Bindings** list page.

Table 9. Data entry controls on the Trigger binding dialog

Column	Description
Name	Text name for the binding. Names must be unique.
Plan Object	The type of object the trigger is defined for, such as project, request, marketing object, approvals, accounts, and invoices. The default is **Any Plan Object**.
Template	The template to use as a filter. After an object is selected, related templates are populated to allow users to specify trigger for a template. Your selection acts as a filter that applies to all instances of objects for that template. The default is **Any Template**, meaning either that all templates for the object are included, or that the object you chose has no templates.
Context	The context for the trigger. For example, the context can be a task or workflow. A default of **Any** means that any context is considered for the filter criterion.

Table 9. Data entry controls on the Trigger binding dialog (continued)

Column	Description
Event	An event for the filter operation. You only see events that correspond to the selected object, template, and context, unless **Any Event** was selected for the object, in which case all events show. Typical events are Created, Updated, Status Changed, Member Assigned/Unassigned, and Objects Linked/Unlinked.
Deferred	The procedure runs sometime after the transaction is committed. If checked, the trigger runs after the configured action is completed; the trigger does not affect the operation on which the trigger is configured. This type of trigger is typically used for post-processing activities.
Validating	A validating trigger starts a procedure to validate data before the current transaction is committed. The trigger sets up the procedure call with an enclosing context, which contains the database transaction. If checked, the trigger runs just before completion of the configured action. Then, if the trigger returns failure, the action on which the trigger is configured rolls back and user receives errors that are reported by the trigger. This type of trigger is typically used for custom validations or any pre-processing activities.
Exclusive	If checked, the trigger procedure runs exclusively even if multiple trigger binding match the action that you selected. For example, if the trigger is not exclusive, whenever the user updates any workflow task, any trigger binding that is configured on the event "Project Updated" also runs. To prevent running multiple trigger bindings, select this check box.

Table 9. Data entry controls on the Trigger binding dialog (continued)

Column	Description
	An exclusive trigger binding does not run with other procedures, even if multiple bindings match. If multiple exclusive bindings match, all the exclusive bindings run. If this box is cleared, the binding is inclusive. Inclusive bindings denote procedures that run with other procedures when the selection criteria for more than one trigger matches. If any matching procedure is exclusive, then only the exclusive matching procedure runs. Only the most specific exclusive bindings match. For example, if you have three exclusive triggers, a global one, one on all projects, and one on a particular project template. Then, if the event fired matches all three, only the one for the particular project template runs.
Procedure	The procedure that is bound to the trigger; that is, the procedure to run when the trigger fires.

Trigger events

The following table includes the trigger events and the context where they are used.

Object	Context	Event	Trigger Description
Account	Account	Created	Approval created
Account	Account	Updated	Approval updated
Approval	Approval	Approved	Approval approved
Approval	Approval	Approved_with_-Changes	Approval approved with changes
Approval	Approval	Cancelled	Approval cancelled

Object	Context	Event	Trigger Description
Approval	Approval	Changed	Approval changed
Approval	Approval	Completed	Approval completed
Approval	Approval	Created	Approval created
Approval	Approval	Denied	Approval denied
Approval	Approval	Response_Request-ed	Approval response requested
Approval	Approval	Restarted	Approval restarted
Approval	Approval	Started	Approval started
Approval	Approval	Stopped	Approval stopped
Asset	Asset	Archived	Asset transitioned to the ARCHIVED state.
Attachment	Attachment	Deleted	Invoice attachment deleted
Account	Attachment	Added	Account attach-ment added
Account	Attachment	Deleted	Account attach-ment deleted
Invoice	Attachment	Added	Invoice attachment added
Project	Attachment	Added	Attachment added to a project
Project	Attachment	Deleted	Project attachment deleted

Object	Context	Event	Trigger Description
Project	Budget	Amount_Changed	Project budget saved
Project	Budget	Finalized	Project budget finalized
Project	Budget	Saved	Project budget allocation saved
Project	Project	Created	Project campaign created
Project	Child_Project	Linked	Child project linked to the project
Project	Child_Project	Added	Child project added to the project
Project	Child_Project	Removed	Child project removed from the project
Project	Task	Completed	Dependent of a workflow task completed
Invoice	Invoice	Cancelled	Invoice cancelled
Invoice	Invoice	Paid	Invoice paid
Invoice	Invoice	Payable	Invoice marked as payable
Invoice	Invoice	Redrafted	Invoice redrafted
Invoice	Invoice	Updated	Invoice updated
Program	Member	Added	Member added to a program

Object	Context	Event	Trigger Description
Program	Member	Removed	Member removed from a program
Project	Member	Added	Member added to the project
Project	Member	Removed	Member removed from the project
Member Team	Member	Added	Team member added
Member Team	Member	Removed	Member removed from the project
Project	Parent_Project Project	Added	Sub project added
Project	Parent_Project Project	Linked	Sub project linked
Project	Parent_Project Project	Removed	Sub project removed from a project
Plan	Plan	On_Hold	Plan completed
Program	Program	Updated	Program's attributes updated
Project	Project	Cancelled	Project cancelled
Project	Project	Completed	Project completed
Project	Project	Created	New project created
Project	Project	Deleted	Project deleted

Object	Context	Event	Trigger Description
Project	Project	Not_Started	project put in "Not Started" state
Project	Project	On_Hold	Project put on hold
Project	Project	Restarted	Project restarted
Project	Project	Started	Project started
Project	Project	Updated	Project's attribute updated
Project	Project	Linked	Project linked to a program
Project	Project	Removed	Project removed from the program
Request	Request	Accepted	Project request accepted
Request	Request	Cancelled	Project request cancelled
Request	Request	Returned	Project request returned
Request	Request	Submitted	Project request submitted
Task	Task	Added	Workflow task added
Task	Task	Assigned	Workflow task assigned to the user
Task	Task	Changed	Workflow task definition changed

Object	Context	Event	Trigger Description
Task	Task	Completed	Workflow task completed
Task	Task	Milestone	Workflow milestone reached
Task	Task	Removed	Workflow task removed
Task	Task	Started	Task started
Task	Task	Stopped	Workflow task stopped
Task	Task	Assigned_To_-Team_Member	Task assigned to team member
Task	Task	Unassigned_From_-Team_Member	Task unassigned from team member
Task	Task	Assigned_To_Team	Task assigned to team
Team	Team	Created	Team created
Team	Team	Deleted	Team deleted
Project	Workflow	Saved	Workflow spreadsheet added

Publishing advanced searches

To help users find objects efficiently, Unica Plan provides an advanced search feature. After users enter criteria for a search, they can provide an identifying name and save the search for future use. Administrators manage saved searches.

When administrators use the advanced search feature and save a search, the system provides an option to publish the search. Publishing a search makes it available to all of

the Unica Plan users. Administrators can also review all of the saved searches, and choose which ones to publish or unpublish.

For more information about searches, see the Unica Plan User's Guide.

Publishing or unpublishing saved searches

After a user saves the criteria for an advanced search, an administrator can publish it to make it available to all Unica Plan users. Alternatively, if a previously published search is no longer widely needed, you can change it back to a private search.

1. Select **Settings > Unica Plan Settings > Manage Published Searches**.
2. By default, the **Manage Published Searches** page lists only your own searches. To work with a search created by another user, select that user in the **Search by User Name** field.
3. To publish a search, select it in the **Available Saved Search** list, then click **>>**.
   The search moves to the **Publish Saved Search** list.
4. To make a search private, select it in the **Publish Saved Search** list, then click **<<**.
   The search moves to the **Available Saved Search** list.
5. Click **Save Changes.**

Viewing a list of locked objects

Unica Plan includes a tool for viewing a list of the objects that are currently locked in the application.

1. To use the tool, enter the following URL into a browser window:

   ```
   http://<HCL_Unica_hostname>:<port>/unica/jsp/main.jsp?
   redirectURL=http://<Plan_hostname>:<port>/
   plan/affiniumplan.jsp?cat=adminobjectlocklist
   ```

   A login screen displays.
2.

Enter the user name and password for a Unica Plan administrator-level account.

A page with the list of object locks displays.

The object lock page lists the current locks that are grouped by objects, grids, and grid rows. The page displays information about each lock, including the ID and the user who has the object that is locked.

Unica Plan Performance

Unica Plan provides configuration properties that you can configure to increase system performance. To access these configuration properties, click **Settings > Configuration > Unica Plan**.

Maximum result size for lists

To help overcome some database query limitations, you can limit the resultset for all system list pages to a specified value. Click **umoConfiguration > database** to configure the **commonDataAccessLayerMaxResultSetSize** property.

Batch size for queries

For certain performance-sensitive queries, you can define a batch size for the resultset. Only the specified number of records are returned at one time. Click **umoConfiguration > database** to configure the **commonDataAccessLayerFetchSize** property. The default setting is 0, or unlimited. A typical setting is 500.

Maximum item count for calendar pages

To manage performance issues when users review or export items in calendar view, you can limit the number of objects (plans, programs, projects, and tasks) that appear. Click **umoConfiguration > listingPages** to set the **maximumItemsToBeDisplayedInCalendar** property. The default setting is 0, or unlimited. The maximum value is 500.

If you limit calendar display items, users can use an advanced search to include wanted items.

Unica Plan markup options

Unica Plan provides markup tools for making comments on attachments. When Unica Plan users send approvals for review, approvers can place their comments directly in the electronic file, where they can be viewed by other users.

Unica Plan provides two types of markup tools.

- Native Unica Plan markup: The native markup option provides various markup functions that can be applied to files in PDF, HTML, JPG, PNG, GIF, and BMP format. Users can mark up whole websites if a URL is supplied. The comments can then be saved in Unica Plan. Native markup is the default option. It does not require Acrobat to be installed on client computers.
- Adobe™ Acrobat markup: This markup tool requires the installation of Adobe™ Acrobat on each client computer. Users can apply all Acrobat commenting features, and then save the edited PDF document in Unica Plan.

The markup option is a global setting. You cannot enable different markup options for different groups of users.

Markup tool availability

By default, the native Unica Plan markup tool is enabled. You can change the type of markup tool available to users by changing the markup configuration properties under **Settings > Configuration > Unica Plan > umoConfiguration > markup**.

Changing the markup tool after users start viewing and editing markups has serious consequences.

- Switching from Acrobat markup to native markup means that users cannot view or edit markups that were made in Acrobat.
- Switching from native markup to Acrobat markup means that users cannot view or edit markups that were made in native markup tool.

Note: For best results, avoid changing markup configuration after users start to use the markup tool. Carefully consider the implications for users before you change markup tool availability.

Enabling Adobe™ Acrobat markup

Users can choose Adobe™ markup for review. Enabling Adobe™ Acrobat markup disables the native Unica Plan markup for all users.

1. Click **Settings > Configuration > Unica Plan > umoConfiguration > markup**.
2. Click **Edit Settings**.
3. Set the **markupServerType** property to `SOAP`.
4. Set the **markupServerURL** property to the URL for the Unica Plan host server. Include the fully qualified host name and the port where the web application server listens.

   Use this path format, with your values substituted for *<server>* and *<port>*.

   `http://<server>:<port>/plan/services/collabService?wsdl`
5. Set the **useCustomMarkup** property to `True` to enable the Unica Plan custom **Send Comments** button for your PDF markups.
6. Click **Save**.
7. Restart Unica Plan for the change to take effect.

Installing and configuring Adobe™ on client computers

For users to use Adobe™ markup effectively, you install Adobe™ Acrobat on each client computer that is used to access Unica Plan.

In addition, users who use the Internet Explorer browser to access Unica Plan must set Internet Explorer preferences to display PDFs in the browser.

Enabling native Unica Plan markup

You can enable native Unica Plan markup for review. Enabling native Unica Plan markup disables Adobe™ Acrobat markup.

1. Click **Settings > Configuration > Unica Plan > umoConfiguration > markup**.
2. Click **Edit Settings**.
3. Set the **markupServerType** property to `MCM`.
4. Click **Save**.
5. Restart Unica Plan for the change to take effect.

Disabling markup

Instead of enabling native Unica Plan markup or Adobe™ Acrobat markup for review, you can disable markup. If you disable markup, users cannot add comments to .pdf files.

1. Click **Settings > Configuration >Plan > umoConfiguration > markup**.
2. Click **Edit Settings**.
3. Clear the **markupServerType** property value.
4. Click **Save**.
5. Restart Unica Plan for the change to take effect.

Configuring system logs

To log configuration, debugging, and error information, Unica Plan uses the Apache `log4j` utility. To change the types of messages that this utility includes in the system log, you must change the value of the level property in the `plan_log4j.xml` file placed in the `<Plan_Home>/conf/` location. The `plan_log4j.xml` file is referenced in the JVM system properites by the `-Dlog4j2.configurationFile` and `-Dlog4.configurationFile`. If you are using an XML file other than `plan_log4j.xml` in the properties `-Dlog4j2.configurationFile` and `-Dlog4.configurationFile`, then you must use the same file, instead of using the `plan_log4j.xml` file, for the earlier mentioned purpose

Setting the logging level

You can set the logging level to FATAL, ERROR, WARN, INFO, or DEBUG to capture an increasing number of messages. For example, to record the maximum number of messages to help troubleshoot a problem, you scroll to the end of the `plan_log4j.xml` file and change the level to DEBUG:

```
<Root level="DEBUG" includeLocation="true">
      <AppenderRef ref="SYS_LOG"/>
</Root>
```

When your investigation is complete, you change the level value back to the default, WARN:

```
<Root level="WARN" includeLocation="true">
   <AppenderRef ref="SYS_LOG"/>
</Root>
```

Tip: The `plan_log4j.xml` file is reloaded 60 seconds after it is updated, so a server restart is not necessary after you edit this file.

Creating a back up of a log file

You can create a back up of a log file after it reaches a certain size by using the size based trigger policy. A back up file is created when the log file reaches the size mentioned in `SizeBasedTriggeringPolicy`. The value of max attribute in `DefaultRolloverStrategy` is the maximum number of back up files that can be created for a log file.

Consider the following example for creating a back up for the `system.log` file.

```
<RollingFile  name="SYS_LOG" fileName="${sys:plan.home}/logs/system.log"
 filePattern="${sys:plan.home}/logs/system.log.%d{yyyy-MM-dd}.%i.log.gz"
 immediateFlush="false" append="true" >
  <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} [%X\{user}] %-5p
 %F.%M:%L: %m%n" />
  <Policies>
   <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
   <SizeBasedTriggeringPolicy size="10 MB" />
  </Policies>
  <DefaultRolloverStrategy max="10"/>
```

```
 <UALevelMatchFilter levelToMatch="DEBUG, WARN,INFO, ERROR, FATAL"
onMatch="ACCEPT" onMismatch="DENY"/>
</RollingFile>
```

Note: If you copy the above example and use it as is, you must comment the default `System Log Appender` in the `plan_log4j.xml` file.

Enabling debugging for specific functionalities

Use the preset modules in `plan_log4j.xml`, such as PROJECT, TASK, APPROVAL, and so on, to enable debugging for specific functionalities.

To debug specific modules, add the following loggers in the `plan_log4j.xml` file and uncomment the relevant sections.

```
<!-- ================================================================ -->
<!-- To debug specific module(s), uncomment the following logger(s) -->
<!-- ================================================================ -->

<!--  Functionality: PROJECT -->
<!--
<Logger name="com.unicacorp.uap.project" level="DEBUG"/>
-->
<!--  Functionality: PROGRAM -->
<!--
<Logger name="com.unicacorp.uap.program" level="DEBUG"/>
-->
<!--  Functionality: PLAN -->
<!--
<Logger name="com.unicacorp.uap.plan" level="DEBUG"/>
-->
<!--  Functionality: TASK -->
<!--
```

```
<Logger name="com.unicacorp.uap.task" level="DEBUG"/>
-->
<!--  Functionality: CALENDAR -->
<!--
<Logger name="com.unicacorp.uap.calendar" level="DEBUG"/>
-->
<!--  Functionality: TEAM -->
<!--
<Logger name="com.unicacorp.uap.team" level="DEBUG"/>
-->
<!--  Functionality: ASSET -->
<!--
<Logger name="com.unicacorp.uap.assets" level="DEBUG"/>
-->
<!--  Functionality: APPROVAL -->
<!--
<Logger name="com.unicacorp.uap.approval" level="DEBUG"/>
-->
<!--  Functionality: ACCOUNT -->
<!--
<Logger name="com.unicacorp.uap.account" level="DEBUG"/>
-->
<!--  Functionality: INVOICE -->
<!--
<Logger name="com.unicacorp.uap.invoice" level="DEBUG"/>
-->
<!--  Functionality: FORM EDITOR -->
<!--
<Logger name="com.unicacorp.uap.formeditor" level="DEBUG"/>
-->
<!--  Functionality: GRID -->
<!--
```

```
<Logger name="com.unica.grid" level="DEBUG"/>
<Logger name="com.unicacorp.uap.grid" level="DEBUG"/>
<Logger name="com.unicacorp.uap.tvc.grid" level="DEBUG"/>
-->
<!--  Functionality: TEMPLATE -->
<!--
<Logger name="com.unicacorp.uap.common.template" level="DEBUG"/>
-->
<!--  Functionality: NOTIFICATION -->
<!--
<Logger name="com.unicacorp.uap.notification" level="DEBUG"/>
-->
<!--  Functionality: OFFER -->
<!--
<Logger name="com.unicacorp.uap.offer" level="DEBUG"/>
-->
<!--  Functionality: MARKETING OBJECT -->

<!-- <Logger name="com.unica.uap.component" level="DEBUG"/> -->
```

Generating SQL logs

You can generate SQL logs when the root logging level is set to DEBUG.

Important: When you upgrade to Fix Pack 10.0.0.2, the `plan_log4j.xml` file is not overwritten. After you install Fix Pack 10.0.0.2, if you want to disable the generation of SQL logs, you must comment the `SQL_LOG` section. Otherwise, SQL logs will be generated after the upgrade.

```


<!-- ================ -->
<!-- SQL Log Appender -->
```

```
<!-- ================ -->
<!-- uncomment the section below for obtaining sql logs -->
<!--
<RollingFile  name="SQL_LOG" fileName="${sys:plan.home}/logs/sql.log"
     filePattern="${sys:plan.home}/logs/sql.log.%d{yyyy-MM-dd}"
     immediateFlush="false" append="true" >

     <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} [%X\{user}] %-5p
%F.%M:%L: %m%n" />
     <Policies>
          <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
     </Policies>
     <UALevelMatchFilter levelToMatch="SQL" onMatch="ACCEPT"
onMismatch="DENY"/>
</RollingFile>
-->
```

Obtaining logs for a specific user in a specific log file

You can obtain logs for a specific user in a specific log file by using the class `com.unica.afc.logger.UserMatchFilter`.

To generate logs for a specific user, you must add and uncomment the following section in the `plan_log4j.xml` file and add the required information for the specific user.

The following example creates a log file `asm_admin.log`, for the user `asm_admin`.

```
<!-- ================================================================ -->
<!-- To obtain log file(s) by user, uncomment the following logger(s) -->
<!-- Below is an example to obtain a log file (asm_admin.log) for the -->
<!-- 'asm_admin' user.


                                    -->
```

```
<!-- ============================================================== -->
<!--
 <RollingFile  name="UserLogAppender"
 fileName="${sys:plan.home}/logs/asm_admin.log"
  filePattern="${sys:plan.home}/logs/asm_admin.log.%d{yyyy-MM-dd}"
  immediateFlush="false" append="true" >
  <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} [%X\{user}] %-5p
 %F.%M:%L: %m%n" />
  <Policies>
   <SizeBasedTriggeringPolicy size="10 MB" />
  </Policies>
  <DefaultRolloverStrategy max="10"/>
  <UserMatchFilter user="asm_admin" onMatch="ACCEPT" onMismatch="NEUTRAL"/>
 </RollingFile>
-->

<!--
 <Logger name="com.unicacorp" level="DEBUG">
  <AppenderRef ref="UserLogAppender"/>
 </Logger>

 <Logger name="com.unica" level="DEBUG">
  <AppenderRef ref="UserLogAppender"/>
 </Logger>
-->
```

Integrating Unica Content Integration

Unica Content Integration facilitates easy integration with Content Management Systems and enables searching content from the them.

The fetched content can be used by the client of Unica Content Integration for various content-oriented business use cases. A Unica Content Integration client is any product from Unica Suite which integrates with it to consume the content from the target systems.

To integrate Unica Content Integration with Unica Plan, complete the following steps:

1. Ensure that you have installed the Unica Content Integration component when installing Unica Platform. For more details, see the topic **Unica Platform components** in the *Unica Platform Installation Guide*.
2. Assign permissions related to Unica Content Integration to the concerned user role in Unica Plan.
    - For information related to Unica Plan users, see Users and user groups *(on page 6)*.
    - For information related to Unica Content Integration permissions, see *Unica Platform Administrator's Guide*.
3. On the Unica Platform home page, select **Settings > Configuration**.
   The **Configuration** page opens.
4. In the **Configuration categories** panel, expand **Plan** and select **umoConfiguration**.
   The **Settings for 'umoConfiguration'** appears.
5. Select **Edit settings**.
   The **(umoConfiguration) - Edit settings** page opens.
6. For the **enablePlanAssetPickerIntegration** field, select `True` and click **Save changes**.
   You can now use Unica Content Integration to configure third-party CMS for use. To configure a third-party CMS, see *Unica Content Integration Administrator's Guide*.

Note: When integrating a third-party CMS with Unica Content Integration for use with Unica Plan, in the **Configuration categories** panel, access the **Content Integration** node inside the **Plan** node.

Chapter 2. Unica Plan interface customization options

Unica Plan offers options for customizing the user interface. You can customize or rebrand the Unica Plan interface to match your organization's needs.

You can customize the following menus and labels.

- Rename standard marketing object types to change the labels on pages and menus.
- Create menus and rename them.
- Reorganize and rename menu items, and add menu items that link to URLs.

You also define the options on several lists in the user interface. For more information, see List definitions (on page 289).

Renaming marketing object types

You can change the labels and other text strings that display in the user interface for marketing object types.

The standard Unica Plan marketing object types include plans, programs, projects, approvals, and assets. You can also add custom marketing object types to meet the needs of your organization. See Marketing object types (on page 51).

To rename an object type, you edit one or more of the following files:

- `<Plan_Home>/messages/com/ibm/umo/ext/UMOConfigurationMessages_<defaultLocale>.properties` (for a standard marketing object type) or `UMOMktObjectConfigurationMessages_<defaultLocale>.properties` (for a custom marketing object type)
  This file contains `list` parameters and `ui` parameters.

    - The `list` parameters define labels and text strings for the list page that displays when users select the menu item for the object type. For example, `projectlist.columnList.PROJECT_STATUS.header=Status`.
    - The `ui` parameters define labels and text strings for the pages that display data for a single instance of that object type. For example, `projectsui.tabset.tab_edit_workflow.item_addTask.display=Add Task Row`.
- *<Plan_Home>*`/conf/`*<defaultLocale>*`/sysmodules.xml`

  See The sysmodules.xml file (on page 40).
- *<Plan_Home>*`/conf/`*<defaultLocale>*`/sysmenu.xml`

  See The sysmenu.xml file (on page 43).

For your changes to take effect in the Unica Plan user interface, you must restart the application server and synchronize menus with Unica Platform.

If your organization supports multiple locales, you can localize the labels that display for the standard marketing object types. See Localized object types (on page 247).

The `sysmodules.xml` file

The *<Plan_Home>*`/conf/`*<locale>*`/sysmodules.xml` file defines the labels that display in the user interface for standard marketing object types.

This file contains sections for `module` elements and `item` elements.

- The `module` elements define how the marketing object type displays on menus in Unica Plan.
- The `item` elements define singular and plural versions of the object type name for other uses in Unica Plan.

For each of the standard marketing object types, the `sysmodules.xml` file includes one `module` element and two `item` elements. For example, the `en_US/sysmodules.xml` file includes the following values for projects:

```
<module id="projects">
     <display>Projects</display>
     <description>Projects Module</description>
     <helptip>Projects</helptip>
     <link>uapprojectservlet?cat=projectlist</link>
     <helpfile>plan.htm</helpfile>
</module>
<item id="project">Project</item>
<item id="projects">Projects</item>
```

To change the labels that display on menus and pages, one of the files that you edit is the `sysmodules.xml` file for your default locale. For example, to change "projects" to "promotions" throughout the user interface, you edit the value of the `<display>` element for the module, and the values for both of the `<item>` elements for projects.

```
<module id="projects">
     <display>Promotions</display>
     <description>Projects Module</description>
     <helptip>Projects</helptip>
     <link>uapprojectservlet?cat=projectlist</link>
     <helpfile>plan.htm</helpfile>
</module>
<item id="project">Promotion</item>
<item id="projects">Promotions</item>
```

You also edit the `sysmodules.xml` file if you want to add a menu item that links to a URL. See Adding a menu item that links to a URL (on page 47).

To apply changes you make in the `sysmodules.xml` file to the Unica Plan user interface, you must restart the server and synchronize menus with Unica Platform.

Note: The default locale for your organization is set during installation. To verify this setting, click **Settings > Configuration > Unica Plan** and review the **defaultLocale** configuration property.

When you create a custom marketing object type, Unica Plan automatically adds a `module` element and the singular and plural `item` elements for it to the `sysmodules.xml` file. For more information, see Marketing object types (on page 51).

Elements in the `sysmodules.xml` file

To define modules in the `sysmodules.xml` file, you use the `<module>`, `<display>`, `<description>`, and `<link>` elements.

module

The `<module>` element is the container element for the elements that define a module. This element has the following attributes:

Attribute	Description
id	Unique name of the module.

The `<module>` element has no value. It can contain the following child elements: `<display>`, `<description>`, and `<link>`.

display

The `<display>` element defines the name that Unica Plan uses for this module in the interface. This element has no attributes and no child elements. The element value is the name that you want to use.

description

The `<description>` element defines the description for this module. This element has no attributes and no child elements. The element value is the description that you want to use.

link

The `<link>` element defines the page that displays when users click the menu item for this module. This element has no attributes and no child elements. The element value is the link.

The sysmenu.xml file

The *<Plan_Home>*/conf/*<locale>*/sysmenu.xml file defines the labels of menus and menu items, including the sequence of the menu items and any separators.

User interface menus and menu items are defined by the elements in sysmenu.xml. Initially, the menu items correspond to the values defined in the sysmodules.xml file for module elements. Every item in the sysmenu.xml file must have a corresponding module element in the sysmodules.xml file.

To reorganize menu items in their menus, rename a menu that you created, or add a menu item, one of the files that you edit is the sysmenu.xml file for your default locale. To apply changes you make in the sysmenu.xml file to the Unica Plan user interface, you must restart the application server and synchronize menus with Unica Platform.

Elements in the sysmenu.xml file

You use the <menugroup>, <display>, and <menuitem> elements to define menus and menu items in the sysmenu.xml file.

menugroup

The <menugroup> element identifies a menu in the user interface, and contains elements that define the label for the menu and the items that it offers. This element has the following attribute:

Attribute	Description
id	Unique identifier for the menu.

The <menugroup> element has no value. It can contain the following child elements: <display> and <menuitem>.

display

The <display> element defines the label that Unica Plan displays in the user interface for this menu. This element has no attributes and no child elements. The attribute value is the name that you want to use.

menuitem

The `<menuitem>` element defines an item on a menu. This element has the following attributes:

Attribute	Description
id	Unique identifier for the module that corresponds to this menu item. A corresponding module id value must be present in the `sysmodules.xml` file.
type	The type of the module that corresponds to this menu item. Optional.

The `<menuitem>` element has no value or child elements.

To include a horizontal line between two menu items, you add a `<separator/>` element to `sysmenu.xml` between the `<menuitem>` elements.

Renaming a marketing object type

You can rename a marketing object type in the `sysmodules.xml` file.

1. Open the `sysmodules.xml` file.
2. Locate the <syscatalogitems> section of this file. For the <item> elements that define the singular and plural names of the marketing object type you are renaming, replace the values with your preferred terminology.
3. Locate the <module> section for the marketing object type you are renaming. Change the value of the <display> element in that section to your preferred terminology.
4. Save and close the `sysmodules.xml` file.
5. Open the `UMOConfigurationMessages_<defaultLocale>.properties` file (for a standard marketing object type) or the `UMOMktObjectConfigurationMessages_<defaultLocale>.properties` file (for a custom marketing object type).

   For the marketing object type you want to rename, locate all list and ui property values (after the = sign) that include the object type name.

6. Change the name values to your preferred terminology. Save and close the file.
7. Open the `sysmenu.xml` file. The menu item names that this file defines override the display names that are defined by `sysmodules.xml`.
8. Change the menu item name values to your preferred terminology. Save and close the file.
9. Restart the Unica Plan application server.
10. Click **Settings > Unica Plan Settings > Synchronize Menus**.

Note: If menus do not display as expected after you complete this procedure, use the `configTool` utility to import menu items manually. For information about using this tool, see the Unica Plan Installation Guide. See the section on configuring Unica Plan before deployment, and the step for registering Unica Plan manually.

Customizing menus

You can configure the menus and menu options for Unica Plan based on the needs of your organization.

You can make the following customizations in Unica Plan:

- Create menus
- Reorganize items on menus
- Rename items on menus
- Rename a menu you created earlier
- Add menu items that link to URLs

You cannot rename the default menus (Operations, Analytics, etc.) although you can rename items within those menus.

Creating a menu

You can create a menu in Unica Plan by modifying the `sysmodules.xml` file.

Before you can create a menu, the `sysmodules.xml` file must contain a module for the menu and a module for each item you want to include on the menu.

1. Open the `sysmodules.xml` file.
2. Add a <module> element with a unique id value for the new menu.
3. Ensure that a <module> exists for each menu item you want to include on the menu.

   Make a note of the id value for each one.
4. Save and close the `sysmodules.xml` file.
5. Open the `sysmenu.xml` file.
6. To create the menu, add a <menugroup> element.

   Enter the same id value for this element that you entered for the new menu <module> element in the `sysmodules.xml` file.
7. To create menu items for the menu, add <menuitem> elements to that <menugroup>.

   Make sure that each <menuitem> element has an id value that is the same as one of the `sysmodules.xml` file <module> elements that you identified in step 3.
8. Save and close the `sysmenu.xml` file.
9. Restart the Unica Plan server.
10. Click **Settings > Plan Settings > Synchronize Menus**.

Reorganizing items on menus

You can reorganize items on menus by moving items from one menu to another or changing the order of items on a menu.

You reorganize items on menus by changing the location of the <menuitem> elements in the `sysmenu.xml` file. When you are through, click **Settings > Plan Settings > Synchronize Menus.**

Renaming a menu or an item on a menu

You can rename a menu or an item on a menu by modifying the `sysmenu.xml` file.

1. Open the `sysmenu.xml` file.
2. Find the <menugroup> element for the menu you want to rename or the <menuitem> element for the menu item you want to rename.
3. Do one of the following options:
    a. If the element has a <display> child element, change the value of the <display> element to the text you want to display.
    b. If the element does not have a <display> child element, create a <display> child element whose value is the text you want to display.
4. Save and close the `sysmenu.xml` file.
5. Restart the Unica Plan application server.
6. Click **Settings > Plan Settings > Synchronize Menus**.

Adding a menu item that links to a URL

You can add a menu item that links to a URL by modifying the `sysmodules.xml` file.

1. Open the `sysmodules.xml` file.
2. Create a module.

    The value of the <link> element must be the URL to which you want to link.
3. Make a note of the values for the id and type attributes.

    You must enter them in the `sysmodules.xml` file in a later step.
4. Save and close the `sysmodules.xml` file.
5. Open the `sysmenu.xml` file.
6. Find the <menugroup> for the menu to which you want to add the link.
7. Add a <menuitem> element that references the module you created earlier.
8. Save and close the `sysmenu.xml` file.
9. Restart the Unica Plan server.
10. Click **Settings > Plan Settings > Synchronize Menus**.

Synchronizing menus

Anytime you make a change to the menus in Unica Plan, you must synchronize the menus in order for your changes to display.

Note: Creating a marketing object type counts as a change to the menus. Unica Plan modifies the `sysmodules.xml` file and the `sysmenu.xml` file automatically, but you must manually synchronize the menus.

To synchronize the menus, click **Settings > Plan Settings > Synchronize Menus**.

Multiple locale support

For organizations with Unica Plan users in multiple locales, you can translate the labels and text strings in the user interface for each locale.

The **supportedLocales** and **defaultLocale** configuration properties, found under **Settings > Configuration > Unica Plan**, identify the locales for your organization. These properties are set during installation.

You can localize the following objects in Unica Plan:

- Standard marketing object types and custom marketing object types. See Localized object types *(on page 247)*.
- Template properties. See Multi-locale support for templates *(on page 249)*.
- Form attributes. See Localizing forms *(on page 253)*.
- Metrics. See Localizing metrics *(on page 256)*.
- Alerts. See Customizing an alert notification message *(on page 282)*.
- Lists, including user roles for project templates. See Localizing lists *(on page 257)*.

Note: Changes to the `<Plan_Home>/messages/com/hcl/umo/core/UMOMessages_<locale>.properties` files to localize system warning and error messages are not supported. System upgrades and other processes overwrite these files.

Chapter 3. Unica Plan integrations

Unica Plan offers integration with Unica Campaign, Unica Centralized Offer Management, and Unica Journey. Integration with Unica Centralized Offer Management is only used for migrating offer management capability from Unical Plan to Unica Centralized Offer Management.

- Unica Plan - Unica Campaign integration *(on page 49)*
- Unica Plan - Unica Journey integration *(on page 50)*
- Unica Plan - Unica Centralized Offer Management integration *(on page 49)*

Unica Plan - Unica Campaign integration

Unica Campaign can be integrated with Unica Plan to use its marketing resource management features to create, plan, and approve campaigns

For a detailed information for Unica Plan - Unica Campaign integration, see Unica Plan and Unica Campaign Integration Guide.

Unica Plan - Unica Centralized Offer Management integration

From Unica 12.1.03 onwards, you can migrate offer management capabilities of Unica Plan to Unica Centralized Offer Management.

Unica recommends that Unica Plan customers evaluate the capabilities of Unica Centralized Offer Management before switching to Unica Centralized Offer Management. Once the users move to Unica Centralized Offer Management, and start managing the migrated offers in Unica Centralized Offer Management, support for switching the offer management capability back to Unica Plan will not be available.

For a detailed information for Unica Plan - Unica Centralized Offer Management integration, see Unica Plan to Unica Centralized Offer Management Migration Guide.

Unica Plan - Unica Journey integration

Unica Plan supports projects to link Journeys created on Unica Journey.

To enable Unica Journey integration, complete the following steps:

- Install Unica Journey. For details, see *Unica Journey Installation Guide*.
- In the Marketing Platform homepage, select **Settings > Configuration**.
- On the **Configuration categories** panel, select **Unica Platform**.
- On the **Settings for 'Unica Platform'** page, select **Edit settings**.
- Set **HCL Plan - Journey integration** to TRUE.
- Click **Save changes**.

Chapter 4. Marketing object types

Marketing objects are the work products that a team develops and reuses in the course of marketing activities. In addition to a set of standard marketing object types that is supplied with Unica Plan, you can create custom marketing object types.

Standard marketing object types

Unica Plan provides a standard set of marketing object types that represent work products that most organizations use to plan and manage marketing goals:

- Plans
- Programs
- Projects
- Assets

These standard marketing object types are added when optional features are enabled for Unica Plan:

- Invoices (Financial Management module)
- Offers (systems with Unica Plan and Unica Campaign are integrated and offer integration enabled)

Custom marketing object types

At your organization, users may need to track other types of marketing objects. Marketing objects can include physical items, such as a letter, credit card, or banner ad. Marketing objects can also represent business components, such as a credit card offer, target segment definition, or rewards program definition. You can define a custom marketing object type for each physical item or business component you want to manage and track in Unica Plan. Given the examples that are listed above, you create these custom marketing object types.

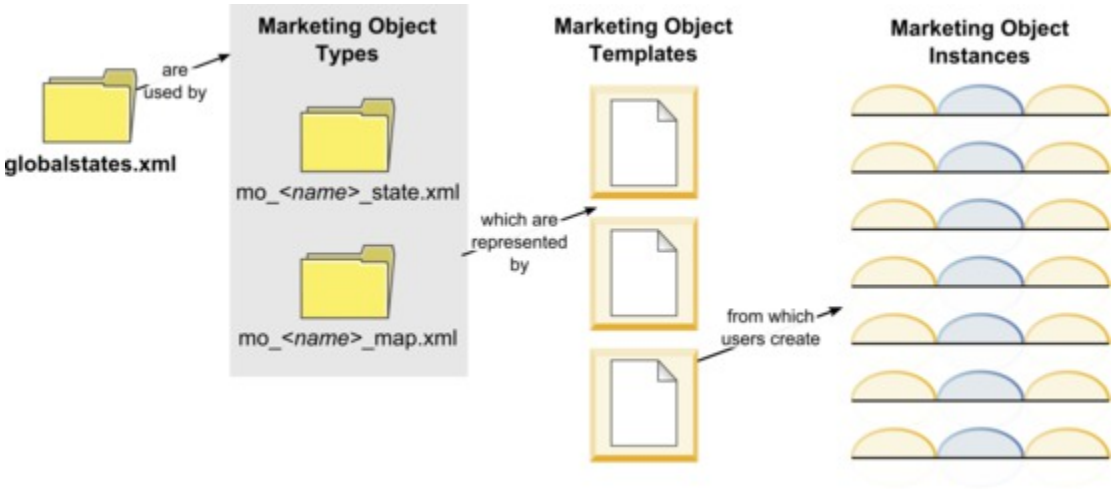
- Letter
- Credit card

- Banner ad
- Credit card offer
- Target segment definition
- Rewards program definition

After you add a custom marketing object type, you create one or more templates for it. For example, you can create separate letter templates for different kinds of form letters. Each template can then be used multiple times to create different instances of that type of letter.

Components of marketing object types

The following diagram shows the internal representation of marketing object types in Unica Plan, and how the components relate to one another.



Typically, object instances move through a series of states or workflow statuses, such as Not Started, In Progress, and Completed. A single XML file defines the possible states for all marketing object types, both standard and custom. To specify the states that users can select for a marketing object type, and the transitions that are allowed between states, you update that marketing object type. Refer to the notes at Marketing object type definition files (on page 59) for further details like 11.0 version onwards generate only a `*_state.xml` file and a `*_map.xml` file is not generated.

Defining a custom marketing object type

To define a new marketing object type for your organization, you complete several tasks.

1. If the new marketing object type requires new states, define the new states in the global states file. See The global states file *(on page 54)* for details.
2. If you defined new states, restart the web server to make the new states available.
3. Add the custom marketing object type. You specify the states that are valid, and the transitions between states, when you add the marketing object type.
   See Custom marketing object types *(on page 56)* for details.
4. Restart the web application server to make the new marketing object type available.
5. Create a template for the marketing object type. See Marketing object templates *(on page 61)* for details.
6. Users create marketing object instances from the marketing object template. See the Unica Plan User's Guide for details.

States for marketing objects

Every marketing object has a state that indicates its status. The state changes as the marketing object moves through its lifecycle.

The following states are available by default:

- Not Started
- In Progress
- On Hold
- Cancelled
- Completed

You can create additional states by editing the global states file.

When you create a marketing object type, you specify the possible states for that type from the global list of states.

The global states file

The global states file lists all possible states for marketing objects in your installation. The `globalstates.xml` file is in the *`Plan_Home`*`\conf\`*`locale`* folder.

You define states in this file with the following XML tags:

Table 10. XML tags for global states

Tag	Description
`id`	A unique identifier for the state. The id value cannot contain spaces.
`displayName`	The label to display for objects when they are in this state. For example, "In Progress".
`icon`	The visual indicator that displays next to the `displayName` to represent this state. Referenced image files must be placed in the `webapp/images` directory of the WAR file. Images that are 20x20 pixels in size and in GIF format provide the best results. The system automatically resizes images of other sizes to 20x20 pixels.
`frozen`	A flag that indicates whether the object can be edited in this state: • `false`: users can edit the object when it is in this state. • `true`: users cannot edit the object when it is in this state.

An example entry for the `IN_PROGRESS` state follows:

```
<state id="IN_PROGRESS">
  <displayName>In Progress</displayName>
  <icon>status_onschedule.gif</icon>
  <frozen>false</frozen>
</state>
```

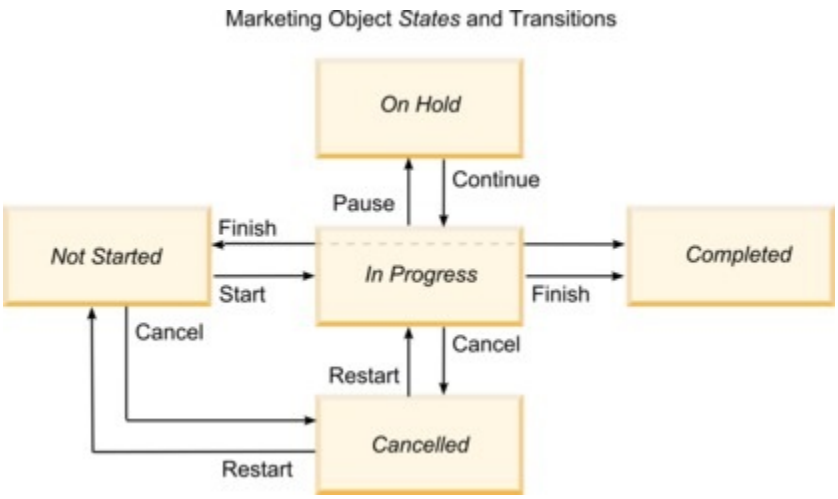
Important: To update the global states file used by your Unica Plan instance, you must restart your web server.

State transitions

When users work with marketing object instances, they can change the status of an individual instance while they edit it. They can also change the status of multiple selected instances on the list page for that object type. Users choose from a list of possible transitions: the values that are available depend on the status of the instance and on the state transitions defined for the marketing object type.

By default, the following transitions are defined.

Transition	From Status	To Status
Start	Not Started	In Progress
Cancel	Not Started	Cancelled
Continue	On Hold	In Progress
Cancel	In Progress	Cancelled
Pause	In Progress	On Hold
Finish	Not Started or In Progress	Completed
Restart	Cancelled	Not Started or In Progress



For example, if a marketing object instance is In Progress, a user can make these choices.

- Pause it: change its status to On Hold
- Finish it: change its status to Completed
- Cancel it: change its status to Cancelled

You add or remove transitions when you create a marketing object type.

Custom marketing object types

You add a marketing object type to Unica Plan before you create marketing object templates, and before users create instances of that type.

Note: Unica Plan does not offer a user interface for updating marketing object types after you add them. Before you begin, determine the state transitions that the marketing object type must offer and collect the information that you need.

Adding a marketing object type

You add a custom marketing object type in Unica Plan.

1. Select **Settings > Plan Settings**.
2. In the **Other Options** section, click **Marketing Object Type Settings**.
   The **Marketing Object Type Settings** page opens.
3. Click **Add marketing object Type**.
   The **Add marketing object Type** page opens.
4. Specify the details for the object type. For descriptions of the fields on this page, see The Add marketing object Type page *(on page 57)*.
5. Click **Save changes** to add the new marketing object type
   When you save the marketing object type, Unica Plan creates XML definition files and properties files with its specifications.
6. If your organization supports multiple locales, translate the labels and text strings that display in the user interface into the language of each locale. For information, see Localized object types *(on page 247)*.

Important: To make the new marketing object type available, synchronize the menu by going to Settings>Plan Settings>Synchronize Menus. Then, restart the web application server and log back in.

The **Add marketing object Type** page

On this page, you define the internal and display names for a marketing object type, how users navigate to instances, and valid states and state transitions.

Table 11. Fields on the Add marketing object Type page

Field	Description
Marketing Object Type Module Name	The internal name of this marketing object type. The name can include only English-language alphanumeric characters and underscores. An all-lowercase version of this name is used in the names of the XML definition files, and in the parameters in the properties files, for this marketing object type.
Marketing Object Type Display Name	The label to use for the marketing object type in menus and listings.
Marketing Object Type Module Description	A brief description of this marketing object type.
Marketing Object Type Module Help Tip	Reserved for future use.
Marketing Object Name (Singular)	The label to use in links and titles in the Unica Plan user interface when a singular name is needed (for example, "Add Creative").

Table 11. Fields on the Add marketing object Type page (continued)

Field	Description
Marketing Object Name (Plural)	The label to use in links and titles in the Unica Plan interface when a plural name is needed (for example, "All Creatives").
Marketing Object Type Menu Settings	To add a navigational menu to access a list page for this marketing object type, select **Create new menu group with display name** and supply the menu name. To add a navigational option to an existing menu to access the list page for this marketing object type, select **Add in existing menu group** and the menu.
Initial State	Specify the state to assign to all newly created instances of this marketing object type.
Transitions Names	The label for a transition from one marketing object state to another. **Note:** This name is only visible on this screen; it does not display to users when they work with marketing objects of this type.
From	The first state in this transition; the state the marketing object instance is moving from. Select a state that is defined in the global states file.
To	The second state in this transition; the state the marketing object instance is moving to. Select a state that is defined in the global states file.

Marketing object type definition files

When you create a marketing object type, Unica Plan updates database tables and creates the following XML files to store definitions for that marketing object type.

- `mo_name_map.xml`, which defines the standard attributes to show on the **Summary** tab for the marketing object. You can also change the labels of these attributes.
- `mo_name_state.xml`, which contains the metadata for the transitions that are defined between states for the object. The metadata for every marketing object state that is defined on your system is stored in the `globalstates.xml` file. If you add any new states, you must define them in that file. If you add a custom state in a marketing object, access the `MarketingOperations/messages/com/hcl/umo/ext/` location and make the following modifications to the `UMOMktObjectConfigurationMessages_<locale>.properties` file:
  - Add a line for each new status key and label. The key must match the state ID as specified in `globalstates.xml` file.
  - Repeat the same for each locale.
  - After saving the changes, restart the web application.

Where *name* is a lowercase version of the name you specified in the **Marketing Object Type Module Name** field when you created the marketing object type.

The marketing object definition files are stored in the *Plan_Home*`\conf\`*locale* directory.

Note:

- In versions before version 8.6.0, the system created two extra files for marketing object types: `mo_name_list.xml` and `mo_name_ui.xml`. These files are no longer created; however, the *Plan_Home*`\conf\backupUiListConfig` directory retains files that were created before the upgrade to version 8.6 for reference.

- From the 11.0 version onwards, only a `*_state.xml` is generated and the `*_map.xml` file is not generated.
- The standard attributes shown in the Summary tab cannot be changed.

Important: Do not remove any of the marketing object definition files. If you do so, you cannot start your web server or use Unica Plan.

Marketing object type properties files

A marketing object type properties file defines the user interface labels and strings for the object type.

For the first custom marketing object type that you create, Unica Plan creates a properties file for each locale. These properties files are the *Plan_Home*`/messages/com/ibm/umo/ext/UMOMktObjectConfigurationMessages_`*locale*`.properties` files.

Each time that you add another custom marketing object type, the system adds a set of parameters for it to these properties files. The new parameters begin with *name*`_ui` and *name*`_list`, wherename is a lowercase version of the name you specified in the **Marketing Object Type Module Name** field when you created the marketing object type.

To change user interface labels and strings for a marketing object type, you edit the properties file for your default locale. For more information, see Renaming marketing object types (on page 39).

To localize user interface labels and strings for a marketing object type, you edit the properties file for each supported locale. For more information, see Localized object types (on page 247).

Marketing object type changes

If you want to change a marketing object type after you create it, you must modify the marketing object type definition files and properties files.

If you want to change how the marketing object type displays in menus, you must modify the corresponding entries in the `\conf\`*locale*`\sysmenu.xml` file under your Unica Plan

installation directory. For more information, see Unica Plan interface customization options (on page 39) and Multiple locale support (on page 48).

Marketing object templates

After you define a marketing object type, you create at least one marketing object template for that type. Users cannot create instances of that marketing object type until you create a template.

Creating a marketing object template

You create marketing object templates in Unica Plan.

1. Select **Settings > Plan Settings**.
2. In the **Other Options** section, click **Template Configuration**.
3. Click **Templates**.

   The **Templates** page includes a section for each of the standard and custom marketing object types in your system. For example, if you configure a custom marketing object type named "Creatives", the page includes a section that is called **Creatives Templates**.
4. To create a template for a marketing object type, click **Add Template** in that section of the page.
5. On the **Add a template** page, supply the properties for the new template.
   For more information, see Building and managing templates *(on page 98)*.
6. Click **Save changes** to add the new template.

Associations between marketing object types

You can associate one marketing object type with a project or with another marketing object.

For example, your organization has a type of project that always includes a brochure. You set up a custom marketing object type for the brochures, and then configure the project template to prompt users for one or more brochures when they create projects. As a result, when a user works with the project and clicks a **Select** control, a list of brochures displays.

The list can contain all brochure instances in the system, or only brochures that are created from a specific brochure template.

You can also configure templates to:

- Include standard or custom attributes that are defined in one template when users create a project or other object instance from another template. You can include an attribute only if you also associate the marketing object types with each other.
- Create an instance for a specified marketing object automatically when users create another type of marketing object. For example, create a brochure automatically when users create a certain type of project. This feature is not available if you are associating the marketing object type with another marketing object.

To configure a template to include data for another marketing object type, you create a form and add attributes to it.

- To associate one type of marketing object with another, you add an attribute with an attribute type of Single Select Object Reference or Multi Select Object Reference to a form. You then add the form to a template.
- To include an attribute that is defined in one template in another type of object instance, you add an Object Attribute Field Reference attribute to the same form. To include an attribute, you must use a Single Select Object Reference attribute to associate the marketing objects.

For more information about forms, see Forms (on page 164). For more information about attributes, see Attributes (on page 196).

Chapter 5. Project and request deactivation

After your organization uses Unica Plan for some time, the user interface can become cluttered with out-dated or unwanted projects and requests. You can deactivate projects and requests to reduce clutter and simplify searches.

Most Unica Plan users can deactivate and reactivate projects and requests from the object Summary tab or project and request list pages. You can deactivate or reactivate up to 1000 projects and requests with this method, which depends on the number of objects that are displayed on each page. Your administrator specifies the maximum number of objects that display on each page in the administrative configuration settings. Administrators can also deactivate or reactivate multiple projects and requests that fit selected criteria with bulk deactivation and reactivation features.

Note: You must have appropriate security permissions to deactivate and reactivate projects. Administrators set permissions at the template level. When a template is created, the Project Owner and Plan Administrator can deactivate and reactivate projects and requests by default.

After you deactivate projects and requests, they no longer appear in the user interface or search results unless you specifically search for it. The following standard searches show deactivated projects and requests.

- Deactivated Projects
- Deactivated Requests
- Deactivated Projects and Requests

You can also locate deactivated projects and requests in searches that are built by filtering these default searches.

The inactive projects and requests remain in your database. Inactive projects can always be reactivated.

To deactivate a project or request, it must fulfill the following criteria.

- Projects must be in the state of Complete, Canceled, or On Hold.
- Project requests must have a status of Canceled.
- If the project has subprojects, the state of the subprojects must also be Complete, Canceled, or On Hold. You must also deactivate the subprojects before you can deactivate the parent project.
- In an integrated Unica Campaign-Unica Plan environment, the campaign end date must pass before you can deactivate a project or subproject with a linked campaign. You cannot deactivate a project with a future end date.

Deactivating projects and requests

You can deactivate projects and requests to remove them from the Unica Plan user interface directly from the list pages.

You can deactivate up to 1000 projects and requests with this method. If you need to deactivate more than 1000 items and have the appropriate permissions, use the bulk deactivation feature.

1. Go to the projects or requests you want to deactivate.
   You can open the project or request **Summary** tab. You can also select multiple projects and requests from a list view such as **All Projects and Requests**.

   Note: Projects must be in the state of Complete, Canceled, or On Hold state. Project requests must have the status of Accepted or Canceled status. Select a list view that shows projects and requests in these statuses.

2. Click
3. **Optional:** Type comments to explain why you are deactivating the project or request.
4. Click **Deactivate**.

After you deactivate projects and requests, you see them in the user interface only if you use predefined searches for deactivated projects and requests. To see the deactivated projects, use the **Deactivated Projects**, **Deactivated Requests**, and **Deactivated Projects and Requests** searches. You can refine these predefined searches for more specific results.

Reactivating projects and requests

You can reactivate projects and requests on an individual basis.

Find deactivated projects and requests with the default searches **Deactivated Projects**, **Deactivated Requests**, and **Deactivated Projects and Requests**. When projects and requests are deactivated, you can view them only in these searches.

You can reactivate up to 1000 projects and requests with this method. If you need to reactivate more than 1000 items and have the appropriate permissions, use the bulk reactivation feature.

1. Go to the projects or requests you want to reactivate.
   You can open the project or request Summary tab. You can also select multiple projects and requests from a list view.
2. Click
3. **Optional:** Type comments to explain why you are reactivating the project or request.
4. Click **Reactivate**.

After reactivation, projects and requests appear in the **All projects and requests** view again.

Deactivating projects and requests in bulk

Deactivate projects and requests in bulk to quickly remove unnecessary objects from the user interface. This procedure reduces clutter and churn when users search for projects and requests. You can always reactivate projects if you change your mind.

Note: You can also deactivate up to 1000 projects at one time from the project list pages.

1. Go to **Settings**>**Administration Settings**>**Deactivation Administration**>**Deactivate Projects and Requests**.
2. Set up a query to find the subset of projects and requests you want to deactivate.

3. After you set up the query, click **Deactivate Projects and Requests** to schedule the deactivation.
4. **Required:** Enter a unique, descriptive name to reuse this query.

   The query name identifies the search so you can reuse it. The log file also draws part of its name from the query name.

   The window lists the number of projects that currently meet the search criteria. Unica Plan runs the query again when deactivation is scheduled and deactivates the projects that fit the criteria then.
5. Schedule the deactivation.
   This feature uses the Unica Platform scheduler. For more information about the scheduler, see the Unica Platform documentation.

   a. Enter or verify the name of the schedule so you can use it again.

   b. Select when to start running the deactivation from the list.

   Note: A bulk deactivation process might require many minutes and use many system resources. Consider scheduling bulk operations for an off-peak time of day.

   - **Now**. If you choose this option, the deactivation goes into the queue immediately.
   - **On a date and time**. Specify when you want the deactivation to start.
   - **On a trigger**
   - **On a trigger after a date**
   - **On completiton of other tasks**

   c. Select when to stop running the deactivation from the following options.
   - **Only run once**
   - **Stop after X occurences**. Set the deactivation to run a specified number of times.
   - **Stop by date and time**. Set the deactivation to run until a specified date and time.

6. **Optional:** You can also set up triggers to run on successful completion of the deactivation or on error.
7. Click **Run with this schedule**.
   After you set up the deactivation, you view the schedule definitions so you can see your deactivation in the schedule.

When the deactivation completes, you receive an alert. If an error occurs, you also receive an alert. If you configured email notifications, you receive an email in addition to an alert. The alert contains the file path to the log for further details or troubleshooting.

The path to the log file is **<Plan_home>\logs\bulkdeactivationreactivation**. The log name is formatted **bulkDeactivationReactivation-<search_name>.log**, where "search name" is the name you entered in step 4. The log file lists projects and requests that were deactivated. For troubleshooting, the log also lists the names of any projects or requests that could not be deactivated categorized by the reason for the error.

After you deactivate projects and requests, you see them in the user interface only if you use predefined searches for **Deactivated** projects and requests. To see the deactivated projects, use the **Deactivated Projects**, **Deactivated Requests**, and **Deactivated Projects and Requests** searches. You can refine these predefined searches for more specific results.

Reactivating projects and requests in bulk

If you deactivated multiple projects and requests and change your mind, you can reactivate projects and requests in bulk.

Note: You can also reactivate up to 1000 projects at one time from the project list pages.

1. Go to **Settings**>**Administration Settings**>**Deactivation Administration**>**Reactivate Projects and Requests**.
2. Set up a query to find the subset of projects and requests you want to reactivate.

Note: Projects and requests appear in your search results only if they fit the criteria and are in a deactivated state.

3. After you set up the query, click **Reactivate Projects and Requests** to schedule the reactivation.
4. **Required:** Enter a unique, descriptive name to reuse this query.

   The query name identifies the search so you can reuse it. The log file also draws part of its name from the query name.

   The window lists the number of projects that currently meet the search criteria. Unica Plan runs the query again when reactivation is scheduled and reactivates the projects that fit the criteria then.
5. Schedule the deactivation.
   This feature uses the Unica Platform scheduler. For more information about the scheduler, see the Unica Platform documentation.

   a. Enter or verify the name of the schedule so you can use it again.

   b. Select when to start running the reactivation from the list.

   Note: A bulk deactivation process might require many minutes and use many system resources. Consider scheduling bulk operations for an off-peak time of day.

   - **Now**. If you choose this option, the reactivation goes into the queue immediately.
   - **On a date and time**. Specify when you want the reactivation to start.
   - **On a trigger**
   - **On a trigger after a date**
   - **On completiton of other tasks**

   c. Select when to stop running the reactivation from the following options.

- **Only run once**
- **Stop after X occurences**. Set the reactivation to run a specified number of times.
- **Stop by date and time**. Set the reactivation to run until a specified date and time.

6. **Optional:** You can also send set up triggers to run on successful completion of the reactivation or on error.
7. Click **Run with this schedule**.
   After you set up the reactivation, you view the schedule definitions so you can see your reactivation in the schedule.

After reactivation, projects and requests appear in the **All projects and requests** view again.

When the reactivation completes, you receive an alert. If an error occurs, you also receive an alert. If you configured email notifications, you receive an email in addition to an alert. The alert contains the file path to the log for further details or troubleshooting.

The path to the log file is **<Plan_home>\logs\bulkdeactivationreactivation**. The log name is formatted **bulkDeactivationReactivation-<search_name>.log**, where "search name" is the name you entered in step 4. The log file lists projects and requests that were reactivated. For troubleshooting, the log also lists the names of any projects or requests that could not be reactivated categorized by the error code.

Chapter 6. Unica Plan reports

Unica Plan provides a set of default reports and dashboard report components. The Unica Plan reports package provides extra reports and dashboard report components, which are created in IBM® Cognos®, a separate business intelligence application.

- For Unica Plan users, there are two ways to access reports.
  - To report information for a single object instance, such as an individual project or marketing object, click the **Analysis** tab for that item.
  - To produce a Cognos® report that includes data for more than one object, select **Analytics > Analytics**.

    Note: Only users with the View Analysis permission can access analytics. For example, to grant access to users with the PlanUserRole level of access, ensure that Plan User is checked under the View Analysis access permission in security policy settings.

    The reports administrator can modify these reports, create new ones, add custom attributes, set up filters, and so on.
- For Unica Plan administrators, reports that help you monitor different activities are available. For more information, see Reports for Unica Plan administrators *(on page 80)*.

For information about installing the Unica Plan reports package, see the Unica Reports Installation and Configuration Guide. For information about creating and managing dashboards, see the Unica Unica Platform Administrator's Guide.

Unica Plan report and folder names in IBM Cognos®

IBM Cognos® Connection presents reports in a directory structure, the top level of which is named **Public Folders**.

When the Unica Plan reports package is installed in Cognos®, **Public Folders** contains the following subfolders for Unica Plan.

- **Affinium Plan**, which contains the multiple-object reports that are listed on the **Analytics Home** page in Unica Plan. If you create new multiple-object reports in Report Studio, save them to this folder. If necessary, you can create subfolders within this folder to organize the reports into a hierarchy.
- **Affinium Plan - Object Specific Reports**, which contains the single-object reports that are listed on the Analysis tabs of individual Unica Plan object instances. This folder contains subfolders for plans, programs, projects, and teams. If you create new single-object reports, you save them to the appropriate subfolder.

Do not rename the folders. If you do change the folder names, note the following requirements.

- Edit the **reportsAnalysisSectionHome** and **reportsAnalysisTabHome** properties under **Settings > Configuration >Plan > umoConfiguration > reports** so that they match the names of the folders.
- Do not use special characters (such as quotation marks or '<') in the folder names. Use only alphanumeric characters, and the space and hyphen ('-') characters if you rename the default report folders.
- If you rename the **Affinium Plan - Object Specific Reports** folder, you must edit the **Project Budget Summary by Quarter** report in Report Studio. This report contains a URL that links to the **Detailed Expense Breakout** report, and it is hardcoded to the report folder name. If the folder name changes, you must edit the links to refer to the new folder name.
- Do **not** rename the subfolders in the **Affinium Plan - Object Specific Reports** folder.

Unica Plan reports and IBM Cognos®

In Cognos®, you can create reports that are based on the Unica Plan data model and you can edit the reports in the Unica Plan reporting pack.

Common customization tasks include the following.

- Adding custom attributes and metrics to reports
- Creating filters for reports
- Adding hyperlinks from a report column to the related Unica Plan object

Before you create or customize reports, update the Unica Plan data model in Cognos® to include any new attributes or metrics you want to use in reports.

Save new reports in the appropriate folder in Cognos®.

Data model updates in IBM Cognos®

Whenever there are changes to the Unica Plan system or custom tables, ensure that the Unica Plan data model in Cognos® is updated to reflect those changes.

For example, if you add custom attributes or metrics, you must update the data model. Otherwise, you cannot use the new attributes or metrics in Cognos® reports.

Updating the Unica Plan data model

You use Cognos® Framework Manager to update the Unica Plan data model.

1. Identify the custom attributes that you want to include in reports and identify the tables (including lookup tables) that are needed for those attributes.
2. Using **Import View** in Cognos® Framework Manager, import the metadata for the attributes.
3. Using **Model View** in Cognos® Framework Manager, define the appropriate relationship between the custom attributes and objects to which they belong. (For example, relate "Project Custom Attributes" to "Project".) Define appropriate relationships to lookup tables.
4. Using **Business View** in Cognos® Framework Manager, define query items and aggregate them into Query Subjects.
5. Republish the data model.

The Query Subjects for the custom attributes and metrics are now available to report writers.

Example Query Subject for a custom metric

You can define a single Query Subject for all of the metrics that are associated with an object type.

This example Query Subject is for metrics that are associated with projects.

```
Select
     UAP_PROJECTS.PROJECT_ID,
     a.METRIC_VALUE1 as TotalRevenue,
     b.METRIC_VALUE1 as ResponseRateActual,
     b.METRIC_VALUE2 as ResponseRateTarget,
     c.METRIC_VALUE1 as TotalLeadsGeneratedActual,
     c.METRIC_VALUE2 as TotalLeadsGeneratedTarget,
     d.METRIC_VALUE1 as TotalCostPassed
From
     UAP_PROJECTS
LEFT JOIN
(select PROJECT_ID, METRIC_VALUE1 from UAP_PROJ_METRICS
 where UAP_PROJ_METRICS.METRIC_ID = 'TotalRevenue') as a
ON a.PROJECT_ID = UAP_PROJECTS.PROJECT_ID
LEFT JOIN
(select PROJECT_ID, METRIC_VALUE1, METRIC_VALUE2 from UAP_PROJ_METRICS
 where UAP_PROJ_METRICS.METRIC_ID = 'ProjectResponseRate') as b
ON b.PROJECT_ID = UAP_PROJECTS.PROJECT_ID
LEFT JOIN
(select PROJECT_ID, METRIC_VALUE1, METRIC_VALUE2 from UAP_PROJ_METRICS
 where UAP_PROJ_METRICS.METRIC_ID = 'NumberOfLeadsGeneratedPassed') as c
ON c.PROJECT_ID = UAP_PROJECTS.PROJECT_ID
LEFT JOIN
(select PROJECT_ID, METRIC_VALUE1 from UAP_PROJ_METRICS
 where UAP_PROJ_METRICS.METRIC_ID = 'TotalCostPassed') as d
ON d.PROJECT_ID = UAP_PROJECTS.PROJECT_ID
```

Filters in IBM Cognos® reports

As part of your design for a IBM Cognos® report, you can provide options to filter report results, rather than selecting all the data in the application. Using IBM Cognos® Report Studio, you can create various filters.

Unica Plan users frequently want filters that do the following.

- Filter by the name or code of an object
- Filter by when an object is active
- Filter by the status, type, or both for an object

Make filter prompts optional, rather than required. Optional filters are simpler to use when the report is run.

Filters for object names or codes

You can include a Select & Search prompt to enable users to filter a report that is based on an object name or object code. The Unica Plan data items use the naming scheme [PlanBV].[Object].[item]. (For example, the data item for project IDs is [PlanBV].[Project].[ProjectID].)

When you create the Select & Search prompt, you can specify one type of value to display to the user and another to use to search the database. For example, the following prompt control configuration prompts users for the project name or code, but searches by using project IDs. This configuration generally results in a faster search.

- Values to use: [PlanBV].[Project].[Project ID]
- Values to display: [PlanBV].[Project].[Project Name (Code)]

Filters for dates

You can create a date filter in Cognos® that returns any object that is active during a certain date range, exactly like Advanced Search in Unica Plan. To do so, use a date prompt with the range option enabled and create a filter that includes both the start date and the end date. This filter returns objects that meet any of the following criteria.

- Start within the active date range
- End within the active date range
- Start before the active date range AND end after the active date range

The following filter searches for projects active during the date range that is entered in the date prompt named Target_Date_Prompt.

```
[PlanBV].[Project].[Project Start Date] in_range ?Target_Date_Prompt? OR
[PlanBV].[Project].[Project End Date] in_range ?Target_Date_Prompt? OR
([PlanBV].[Project].[Project Start Date] <= ?Target_Date_Prompt? AND
[PlanBV].[Project].[Project End Date] >= ?Target_Date_Prompt?)
```

Filters for object status and type

Because there is a small, fixed set of statuses and types, use a simple multi-select control for filtering on status or type.

To prompt users for the status or type of an object (or both), do the following.

- To prompt for status, use a multi-select control that uses the *OBJECT* Status Query Subject.
- To prompt for type, use a multi-select control that uses the *OBJECT* Template Query Subject.

Hyperlinks in IBM Cognos® reports

You can create hyperlinks in Cognos® reports so that users can open a referenced object in Unica Plan from within the report.

For example, if the report includes a list of projects that you hyperlinked, users who click a project name in the report open the Summary tab for that project. Hyperlinks also work in reports that are emailed to users. Users who click the links may be asked to log in to Unica Plan.

You can create hyperlinks for the following objects.

- plans
- programs
- projects
- project requests
- standalone approvals
- work and approval tasks
- invoices

The Unica Plan reports package includes a URL query item for each object for which you can create a hyperlink. For example, the URL query item for plans is named Plan URL. The URL query item for an object is listed in the query subject for the object.

In Cognos® Report Studio, use the appropriate URL query item to define the URL source for the hyperlinks.

Example custom report: Project Performance Summary (Custom)

The Unica Plan reports package provides two versions of the Project Performance Summary. The Project Performance Summary uses only default attributes. The Project Performance Summary (Custom) contains custom attributes and metrics.

A list of the changes that were made to the Unica Plan data model and to the report in Cognos® to produce the Project Performance Summary (Custom) follows.

Custom attributes and metrics identified

The following custom attributes and metrics were needed to produce the Project Performance Summary (Custom).

Attribute	Column	Lookup table
Initiative	dyn_projectatts.init_type_id	lkup_initiative
Business unit	dyn_projectatts.business_unit_id	lkup_business_unit
Product Family	dyn_projectatts.prod_family_id	lkup_prod_family

Attribute	Column	Lookup table
Segment	dyn_projectatts.segment_id	lkup_segments

The following are the custom metrics that were needed for the report.

- Total Revenue: metricid = 'TotalRevenue' (actual)
- Response Rate: metricid = 'ResponseRate' (actual)
- Total Leads Generated: metricid = 'NumberOfLeadsGeneratedPassed' (actual, target)
- ROI: metricid = 'ROI' (actual)

Metadata associated with the custom attributes

The following columns in the dyn_projectatts table were imported to support the custom attributes.

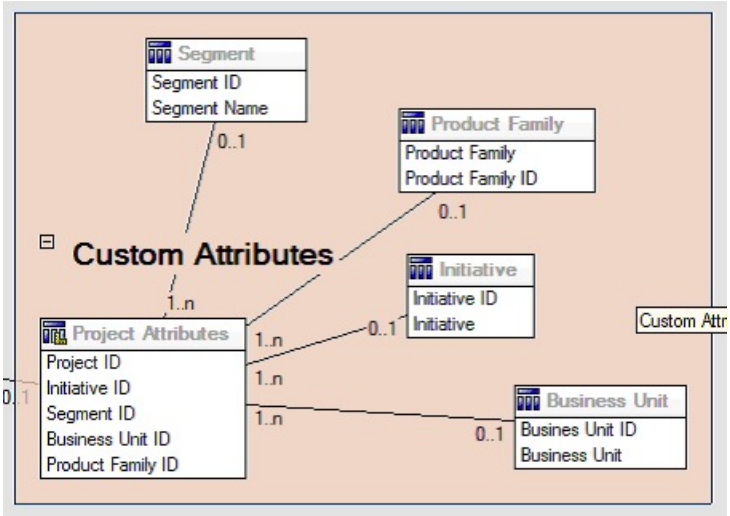
- init_type_id
- segment_id
- business_unit_id
- prod_family_id

The following lookup tables were imported to support the custom attributes.

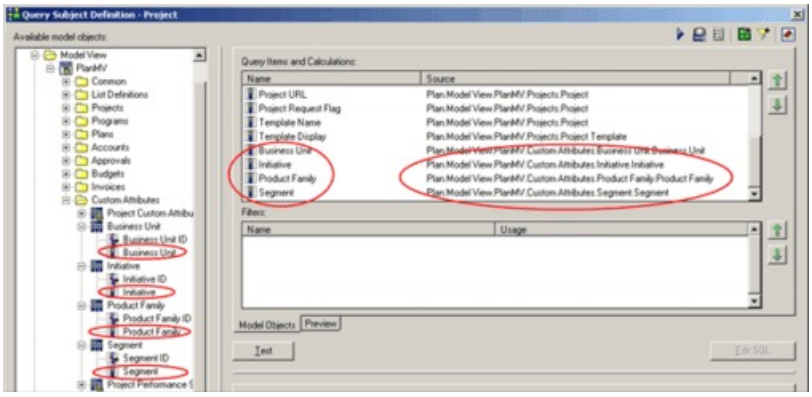
- lkp_initiative
- lkup_segments
- lkup_business_unit
- lkup_prod_family

Relationships and Queries defined in Model View

In Model View in Cognos® Framework Manager, relationships were defined as shown here.



The Query Subject definition for Project was updated with query items for the custom attributes, as shown here.



Query Items added in Business View

The following query items were added in the Business View in Cognos® Framework Manager.

Column	Type/extra info	Query item
Initiative	String; group-by column	Project Custom Attributes.Initiative

Column	Type/extra info	Query item
Business Unit	String	Project Custom Attributes.Business Unit
Segment	String	Project Custom Attributes.Segment
Product Family	String	Project Custom Attributes.Product Family
Total Revenue	Currency	Project Performance Summary Metrics.Total Revenue
Response Rate (Actual)	Percent	Project Performance Summary Metrics.Response Rate (Actual)
Response Rate (Target)	Percent	Project Performance Summary Metrics.Response Rate (Target)
Response Rate Variance	Percent, Calculation	ResponseRate Actual - ResponseRate Target
Total Leads Generated (Actual)	Number	Project Performance Summary Metrics. Total Leads Generated (Actual)
Total Leads Generated (Target)	Number	Project Performance Summary Metrics. Total Leads Generated (Target)
Total Leads Generated Variance	Number, Calculation	Total Leads Generated (Actual) - Total Leads Generated (Target)
ROI	Percent, Sort Column, Calculation	Project Performance Summary Custom Metrics].[Total Revenue]-[Project Budget].[Actual Total])/[Project Budget].[Actual Total]

Columns added and deleted from the report

In Cognos® Report Studio, all the columns were removed from the report except for Project Name (Code), Project Start Date, and Project End Date.

The following columns were added to the report.

- Initiative
- Business Unit
- Segment
- Product Family
- Total Revenue
- Response Rate (Actual)
- Response Rate (Target)
- Response Rate Variance
- Total Leads Generated (Actual)
- Total Leads Generated (Target)
- Total Leads Generated Variance
- ROI

Prompts created

The following two prompts were created.

Prompt	Prompt type	Query Subject
Initiative	Search & Select	Project Custom Attributes.Initiative
Business unit	Search & Select	Project Custom Attributes.Business Unit

Reports for Unica Plan administrators

Reports are available to help Unica Plan administrators monitor different user activities.

- Individual project and request owners can view out-of-office team members on the People tab for their projects. As an administrator, you can track all users with the out-of-office parameter set. See Producing the Out of Office Users and Delegation Summary *(on page 81)*.

- Your organization can require users to provide a reason when they respond to an approval with Deny. You can analyze the reasons that are selected for denials over time, or for a specific project or stand-alone approval. See Producing a Deny Reason Analysis Report *(on page 81)*.

Producing the Out of Office Users and Delegation Summary

Administrators can produce reports to track who is out of the office, and who the delegates are.

1. Select **Settings > Plan Settings > Out of Office Users and Delegation Summary**.
2. In the list of **Projects/Requests**, click to select None, All, or a listed project or request. To select several projects and requests, use Shift+click or Ctrl+click.
3. In the list of **Approvals**, click to select None, All, or one or more of the listed approvals.
4. In the list of **Users**, click to select All or one or more of the listed user names.
5. Click **Search**.
   For each selected project, request, and approval, a list of the project team members, request recipients, or approvers who are out of the office displays. The specified delegate and delegation start date also display.
6. To save the report in a spreadsheet, click **Export**.

For more information about how project owners and participants use out-of-office settings, see the Unica Plan User's Guide.

Producing a Deny Reason Analysis Report

Administrators can review a list of approvals, with the total number of denials and the percentage for each selected reason for denying approval.

Different reports are available for workflow approvals and stand-alone approvals. You use similar procedures to produce these reports.

1. Click **Analytics > analytics**.
2. To produce a report for workflow approvals, click **Deny Reason Analysis Report for Workflow Approvals**.

To produce a report for stand-alone approvals, click **Deny Reason Analysis Report for Standalone Approvals**.

3. For a workflow approval, select the project template or All Templates.
4. You can enter the name of a specific project for a workflow approval, or enter the name of a specific approval for a stand-alone approval.
5. You can supply start and end dates.
   The report includes projects with an end date within this date range, or stand-alone approvals with an approval due date in this range.

   To run the report for all dates, delete the default start and end date values.

Chapter 7. Template Basics

A template for a marketing object type, such as a plan, program, or project, defines the information that your organization wants to capture about the object. Then, you use templates to create object instances. If you have the Financial Management module installed, an invoice template defines the information that your organization wants to capture in invoices.

After you install Unica Plan you can use the example marketing object type and invoice templates that are provided by HCL to get started. The example templates give you an opportunity to see what templates are and how Unica Plan uses them. After you understand templates, you can customize the example templates or create your own templates.

When to create another template

You can add templates at any time. Design new templates when you must capture new information about a marketing object type.

You design and implement templates to provide an appropriate framework for each type of marketing object. Generally, you create a new template only when the information that you want to capture about a marketing object type differs from the information the existing templates collect. For example, in one type of project you want to capture specific metrics, provide a specific workflow, and supply specific reference materials as attachments. For another type of project, one or all of these requirements are different. In this case, use a different template for each project.

You can also choose to use different project templates for variations of a marketing program. For example, you might have the following templates:

- A project for a routine monthly direct mailing.
- A project template for a targeted direct marketing program around a new product launch.
- A project to organize your organization's booth at a tradeshow.

Each of these project types can have its own project template.

Note: The exceptions to this method are plans, invoices, and assets: your organization can set up only one template each of these marketing object types.

Decide on a set of templates

Design one template for each marketing object type.

Examples of how an organization can create a set of templates to meet business needs follow.

Your organization needs to collect information about projects that create or modify marketing collateral, so you create a project template called Marketing Collateral. Whenever someone in your organization creates a project that involves marketing collateral, they select the Marketing Collateral template and create the project from that template. For more information about creating a project, see the Unica Plan User's Guide.

Your organization also develops and runs new direct marketing campaigns. To collect the information that is needed for those projects, you create another template that is called Database Marketing Campaign. Whenever someone in your organization creates a project for a particular database marketing campaign, they select the Database Marketing Campaign template. Then, they create the project from that template.

You can have as many templates as you have types of marketing projects or types of marketing programs.

Example templates

Unica Plan provides a few example templates. You can use the example templates to start creating marketing objects. Then, you can modify the example templates and create custom templates for your organization.

Unica Plan provides a few program and project templates. Unica Plan also provides one default template for a plan, an invoice, and an asset; you can edit these templates to suit your needs, but you cannot create new templates for these object types. The examples are in the following file under your Unica Plan installation.

```
\tools\admin\sample_templates\sampleTemplates<database>.zip
```

where *<database>* is your database. For example, if you are using an Oracle database, you should import `sampleTemplatesOracle.zip`.

List of example templates

Unica Plan comes with several example templates.

The following example templates are for programs.

- Database Marketing contains basic information about programs that develop and implement direct marketing campaigns.
- Product Launch contains basic information about programs that develop and carry out new product launch campaigns.

The following example templates are for projects.

- Database Marketing Campaign contains basic information about projects that develop and complete direct marketing campaigns.
- Tradeshow contains basic information about projects that plan tradeshows.
- Marketing Collateral contains basic information about projects that develop marketing collateral.
- Unica Campaign, which if Unica Plan-Unica Campaign integration is enabled, contains information about projects that are linked with campaigns in Unica Campaign.

For details on importing the example templates, see Importing template metadata (on page 307).

Template components

Templates include default data, and different components that you create and manage. Template components are modular, reusable structures that help you meet the needs of the users who create instances of different types of marketing objects.

To work with templates and template components, click **Settings > Plan Settings > Template Configuration** to open the **Template Configuration** page.

Fields

A field is a data entry for a piece of data. For example, fields can record the phone number of the marketing manager, or the data type of an attachment. In Unica Plan, you use attributes to define the fields (or other controls) that display in the user interface.

Fields can be standard or custom. For more information, see Attributes (on page 196).

Shared attributes

Shared attributes are specialized fields. Each attribute has a specific format for collecting data, such as Yes or No, Single-Select from a predefined list or from a database lookup table. For more information, see Attribute types list (on page 208). After you define shared attributes, you can import them into different forms.

To create an attribute, on the **Template Configuration** page click **Shared Attributes**. For more information, see Attributes (on page 196).

Forms

A form organizes groups of fields for collecting data. After you define the standard and custom fields that you want in a template, you use a form to layout the fields.

You can then add the form to the Summary tab of a template. For some types of templates, you can add the form to the template as a custom tab. You add custom forms to templates on the template Tabs tab.

In marketing object templates, each tab contains one or more forms. You can use a form across marketing object templates; for example, in both project and program templates.

To create a form, you create the attributes that collect data and combine them into a form definition: on the **Template Configuration** page click **Forms**. For more information, see Forms (on page 164).

Rules

A set of verification functions that can be applied to a form automatically to assure that entered data is valid. For example, you can add rules to automatically approve budget line items or send them for approval.

You use an XML editor to create a rules file, then add it to Unica Plan: on the **Template Configuration** page click **Rules**. For more information, see Grid validation (on page 337).

Metrics

Metrics measure the performance of the object. Metrics are user-entered or computed numbers. Typical financial metrics include cost and revenue, while performance metrics can track the number of contacts and the number of responses in a particular marketing campaign.

Metrics are assigned to metrics templates, which you can associate with a plan, program, or project template. If you associate a metrics template with an object template, instances of that object include the Tracking tab.

You can configure metrics in the following ways.

- Set up metrics that calculate based on other metric values. For example, metrics can calculate that the Profit of a campaign is the Revenue minus the Cost.
- Group metrics.
- Define both the metrics and their groups.
- Roll up metrics from projects to programs, and from programs to plans.

To create and edit metrics and metrics templates, on the **Template Configuration** page click **Metrics**. For more information about metrics, see Metrics (on page 237).

Workflow

Stages, tasks, milestones, personnel, dependencies, and other data that organize and schedule the work that is needed to complete a project. Workflow is used in project templates only.

To create or edit a workflow template, you define a workflow on the Workflow tab of a project template or in any project instance. You can then save that work as a separate workflow template. A workflow template can then be imported into the Workflow tab of any project template or project instance to replace any previously supplied values.

To disable, enable, or delete workflow templates, or to export them to or import them from another Unica Plan instance, on the **Template Configuration** page click **Workflow**. For more information, see Project template Workflow tab (on page 137).

Data Mapping

If Unica Campaign and Unica Plan integration is enabled, a data mapping file establishes how the metrics for tracking and roll-up established in each system correspond.

You use an XML editor to create a data mapping file, then add it to Unica Plan by clicking **Data Mapping** on the **Template Configuration** page. For more information, see Data Mapping Definitions (on page 159).

Icons

Images that represent marketing objects in the user interface. You use image-editing software to create icon images, then add them to Unica Plan by clicking**Icons** on the **Template Configuration** page. For more information, see Icons page (on page 120).

Attachment folders

You add folders to the Attachments tab so that users can organize their attachment files into meaningful categories. When users add an attachment, they can add it to a folder. For example, a user might add a brochure as an attachment and file it in the Creative Ideas folder that is provided by the template. Users cannot add their own folders to the Attachments tab.

Tabs

Tabs organize information. Different kinds of tabs can contain forms, rules, metrics, workflows, data mappings, icons, and attachment folders. Templates have some standard forms; in addition, some types of templates can have custom tabs. For those templates, you can compile several forms into a custom tab.Adding tabs to templates (on page 112).

After these template components are defined and available, you assemble them into templates.

Template-building methodology

Building a custom template is a bottom-up process. You build each component that you need, and then assemble them into a complete template. You use the complete template to create instances of the object.

Task 1: Planning

Before you begin building templates within Unica Plan, analyze the needs of your organization, and plan out the types of templates you need. For more information, see Unica Plan custom templates (on page 90).

Task 2: Define attributes and forms

After you determine the types of fields you need and how you want to organize them, create the attributes and forms. For more information, see Creating, editing, and deleting attributes (on page 200) and Creating forms (on page 166).

Task 3: Define metrics

After you determine the types of metrics you need, create and edit the appropriate metrics. For more information, see Metric creation overview (on page 239).

Task 4: Define other template components

Using the appropriate software, create any icons and data mapping files your template needs.

Task 5: Define the template

Assemble the components into the template. You can create custom tabs, and specify the icons, forms, metrics, and other components, to use in the template. For more information, see Adding or editing templates (on page 96).

Task 6: Test the template

Use your new template to create a marketing object. Building a template is an iterative process. You usually must go back and tweak the individual components, maybe swap components in and out, and then retest the template by creating new objects. For more information about building objects from templates, see the Unica Plan User's Guide.

Unica Plan custom templates

Most of the work that is involved in creating custom templates is determining the fields that you want in your templates and how to organize those fields. Capturing this information on paper before you start creating attributes and forms streamlines the creation process.

Before you begin creating custom templates, decide on the types of templates that your organization needs. Then, create a storyboard or a spreadsheet that shows the custom template's tabs and each field that you want on each tab.

For example, you want a project to list the business unit that requested the project. Therefore, you record the following information about the business unit field.

Table 12. Example of recording template field information

Attribute information	Value
Shared or Local	Shared
Attribute Category	Form
Attribute Type	Single-Select
Internal Name	BusinessUnit
Display Name	Business Unit
Tab/grouping	In the Summary tab under the Collateral Request information section.
Field type	Drop-down list

Table 12. Example of recording template field information (continued)

Attribute information	Value
Possible values or database table to retrieve values from	Retail banking, investment services, insurance, credit card brochure, postcard, data sheet, white paper, print ad, or a table name or column name where Unica Plan looks up these values.
Required?	Yes
Help Tip	Enter the business unit that requested this collateral piece.

After you do this planning step for every field in a form, you can create the attributes and the form. Then, compile the tabs and forms into a custom template.

What can you customize?

The items that you can customize for your templates depend on the marketing object type. You can change the names of standard attributes; add custom attributes, metrics, and workflow; and translate templates.

The items that you can customize for your templates depend on the marketing object type. An overview of the different ways you can customize templates follows.

- You can rename the standard attributes on the Summary tab for all template types. You can also mark those attributes as required, standard, or hidden to streamline the template. See Attributes tab *(on page 108)*.
- You can add custom fields to the Summary tab for all template types. See Sample Summary tab *(on page 92)* for an example.
- You can add a tab with custom fields to templates for programs, projects, offers, and your custom marketing object types. See Custom tab example *(on page 93)*.
- You can create metrics and add them to a metrics template. Then, associate the metrics template with a plan, program, or project template. Users see the metrics

on the Tracking tab when they work with instances of one of these marketing object types. See Metrics *(on page 237)*.

- On the project Workflow tab, you can customize nearly all of the characteristics, including the stages, tasks, dependencies, and durations. See Project template Workflow tab *(on page 137)*.
- For the Attachments tab, you define the folders that help users organize attachment files. You can also add default attachments to all objects that are created from a template. See Template Attachments tab for adding folders and files *(on page 117)*.
- For projects, requests, and custom marketing objects, you can customize security permissions for each tab included in the templates (both custom and standard tabs). See Controlling access to templates *(on page 270)*.
- You can customize the Budget tab, which is available for programs and projects when the Financial Management module is enabled. See Budget tab customization *(on page 124)*.

You can also configure templates to translate values for different locales. See Multi-locale support for templates (on page 249).

Sample Summary tab

When you build a template, you specify information for users to enter on the Properties tab. When users create an instance from the template, the information that is entered on the Properties tab becomes the Summary tab in the marketing object.

Note:

- The fields in the top portion of the tab (the Database Marketing Campaign section) are standard attributes. When creating the object, users enter this information on the template's Properties tab.

  For most of these attributes, you can change the label and determine whether the field is Standard, Required, or Hidden. You cannot remove the project

name or project code. For more information about customizing standard attributes, see Attributes tab (on page 108).

- The fields in the bottom portion of the tab (the Campaign Info section) are custom fields. You create custom fields by adding attributes to a form, which groups related fields together.

The custom fields allow you to capture the information that is needed for the project. You can later use the values that are entered in these fields for reporting and analysis purposes. Some examples of custom fields are:

- A text box where the user enters free text.
- A drop-down list or multi-select list box, where the user selects a value or multiple values from a list. You can specify a static list that the user selects from or you can specify a database table from which the list retrieves values.
- A radio button, where the user selects exactly one option from several predefined options.

Custom tab example

You can add custom tabs to the marketing object to capture customized information. Add forms to the template Tabs tab to create custom tabs in the marketing object.

For example, a custom tab named "Creative Development" tab could have fields that capture information about the objectives, background, and audience profile for the marketing campaign.

Custom tabs can capture information about a project that you currently capture on paper forms and give to other departments or vendors. By including this information in your project, you can ensure that users complete the information and minimize delays if the information is incomplete. You can also ensure that all team members see it by keeping all information in one location.

Unica Campaign project templates

If Unica Plan is integrated with Unica Campaign, campaign project templates guide users in creating a campaign project. Campaign projects integrate the planning and project management features of Unica Plan with the campaign development features of Unica Campaign.

You indicate that a template is a campaign project template when you create the project template. In integrated systems, all new project templates include a Campaign tab on which you complete the options on the Campaign tab for the template. Then, you must specify a Target Cell Spreadsheet. If you want to import Unica Campaign contact and response metrics, you must specify a metric map file.

Unica Campaign project template design

You can create as many campaign project templates as you want. For example, you might want to create a separate campaign project template for each type of campaign that you run.

You create a separate template for each unique combination of forms you need. For example, if you need to gather different information to define your target cells for some campaigns, you create different Target Cell Spreadsheets (TCS®). Then, associate each TCS® with a different template. Similarly, if some custom campaign attributes are relevant only for certain types of campaigns, you can create different campaign project templates. You can create separate templates to control custom campaign attributes, the attribute display order, and their organization on tabs.

Offer templates

If Unica Plan is integrated with Unica Campaign, and optional offer integration is also enabled, you create offer templates in Unica Plan. Offer templates guide users in creating offers.

To work with offer templates, you select **Settings > Plan Settings > Template Configuration > Templates** and use the options in the Offer Templates section.

For more information about the features of offer templates, and about administering and using offers, see the Unica Campaign guides for administrators and users.

When you enable offer integration, you can import any pre-existing offer templates and their custom offer attributes, along with offers, offer lists, and offer folders, from Unica Campaign. For more information about enabling offer integration, see Unica Plan and Unica Campaign Integration Guide.

Chapter 8. Adding or editing templates

After you identify the template components that you need, you can create the template. You need to build components before assembling them in the template.

Before you create a template, review your existing template components to see whether you can reuse any or if you need new components. Template components include: metrics or metrics templates for tracking performance and financial data, attributes and forms for collecting data, and workflow templates for identifying and scheduling project management tasks.

The steps to create a template are generally the same for each object type, with the following exceptions.

- Unica Plan offers only one template for plans, one template for invoices, and one template for assets. You can edit these templates as needed, but you cannot create more plan, invoice, or asset templates.
- For project, program, and plan templates, you can select a metrics template on the Properties tab.
- In each project template, you can specify a workflow within that template, or you can import a previously defined and reusable workflow template.
- If Unica Plan-Unica Campaign integration is enabled, you identify a project template as a campaign project template on its Campaign tab.

To add or edit a template:

1. Select **Settings > Unica Plan Settings**.
2. In the Other Options section, click **Template Configuration** then click **Templates**.
3. On the **Templates** page, scroll to the section for the type of marketing object you want to work on.
4. To create a template, click **Add a template** in that section. To edit an existing template, click its name.

5. Supply data on the template Properties tab. This tab corresponds to the Summary tab in the instances that users create from this template. You must supply a template display **Name** and an internal **Template ID**.
   The **Template ID** can include lowercase alphanumeric values only. Do not use accented or non-Roman characters.

   For project, program, and plan templates, you also select a metrics template and set the security policy on this tab. For more information, see Template Properties tab for defining the template (on page 104).
6. Click **Save changes** on the Properties tab.
7. Supply data on other tabs to complete the template. The tabs that are available depend on the type of template you are creating or editing.

Important: Click **Save Changes** when you finish editing each tab, and before you click another tab in the template. Otherwise, your changes are not saved.

Table 13. Tabs available for each type of template

Tab Name	Plan/ Custom	Pro- gram	Project	Invoice	Asset	Offer
Properties	X	X	X	X	X	X
Attributes	X	X	X			
Tabs	X	X	X	X	X	X
Attachments	X	X	X			X
Custom Links	X	X	X			X
Customize Alerts	X	X	X	X	X	X
Budget Approval Rules		X	X	X		
Project Roles			X			
Request			X			

Tab Name	Plan/ Custom	Pro-gram	Project	Invoice	Asset	Offer
Workflow			X			
Campaign			X			

Building and managing templates

To create and manage templates and template components, you use the **Template Configuration** page. Select **Settings > Unica Plan Settings**. Then, click **Template Configuration**.

The items and functions on the **Template Configuration** page are organized into two sections, **Template Configuration** and **Templates Components**. There is also an option to validate all templates.

For information about the entire process of building and managing templates, see Template-building methodology (on page 89).

Template Configuration section

The template configuration section of the **Template Configuration** page contains the **Templates** link. This link opens a page that lists all the existing templates and template folders that are organized by marketing object type. You use the links on that page to create, delete, and organize templates, and to edit or export individual templates.

Validating templates

To run a utility that validates templates and forms and shows any validation errors, click **Validate Templates** in the template configuration section.

Templates Components section

The templates components section of the page contains the following links.

Table 14. Links in the Templates Components section

Links in the Templates Components section

Link	Description
Forms	Opens the **Form Definitions** page, which lists the form definitions and provides options for working with forms. The following information shows for each form definition: • Name of the form • Database table name that stores the values users enter in the form fields • List of templates that use the form Use the links and icons on the **Form Definitions** page to create, import, enable, disable, delete, export, copy, publish, and manage forms.
Metrics	Opens a page with sections for Metrics Templates, Metrics, and Metrics Dimension. Unica Plan displays the name and a short description for each listed item. For Metrics Templates, Unica Plan displays the following information and options. • ID; used when you add a metrics template to an object template. • List of templates that use the metrics template. • Links to edit or delete individual metrics templates. • An **Export Properties File** link to export properties files for metrics. • An **Import Metrics Template** link to import an xml file for a metrics template or a properties file. • An **Add Metrics Template** link to add a metrics template.

Table 14. Links in the Templates Components section

Links in the Templates Components section

(continued)

Link	Description
	Metrics also have the following extra information and options. • ID; used when you add a metric to a metrics template. • List of projects that use the metric. • Links to edit or delete individual metrics. • An **Add Metrics** link to add a metric. Metrics Dimension also have the following extra information and options. • Type (Actual, Target, Other) of each metrics dimension. • Links to edit or delete individual metrics dimensions. • An **Add Metrics Dimension** link to add a metrics dimension. Use the **Legacy Metrics Specification Files** link to retrieve legacy metrics specification files if they were uploaded before your upgrade to Unica Plan version 8.5.0. From Unica Plan 8.5.0 onwards, users cannot add any further files.
Workflow	Opens a list of separately saved workflow templates and shows the following information. • Name • The number of stages and tasks in the workflow template • When it was first created and last modified • Whether it is enabled or disabled

Table 14. Links in the Templates Components section

Links in the Templates Components section

(continued)

Link	Description
	You create workflow templates by saving the work that is done on the Workflow tab of a project template or instance. You can use the links on this list page to delete, enable/disable, import, or export a workflow template.
Data Mapping	Opens a list of data maps and shows the following information. • Data mapping file names • Type: Campaign Metrics Import (if you have data maps from previous versions, you might see other values) • List of templates that use the mapping. • You can use the links on the **Data Mapping** page to add and delete data mapping files. For more information, see Data Mapping Definitions (on page 159).
Icons	Opens a list of icons and shows the following information. • Icon images; large and small • Icon name • List of templates that use the icon • Delete link for deleting the icon (does not delete the file from its location on disk) Click **Add icon** to add an icon. For more information, see Icons page (on page 120).
Rules	Opens the **Rules Definitions** page. Click **Add Rules Definition** to add rules.

Table 14. Links in the Templates Components section

Links in the Templates Components section

(continued)

Link	Description
Shared Attributes	Opens a list of the shared attributes in the system, which is organized by attribute category, and shows the following information. • Display name • Type of attribute • List of templates that use the attributes

You can also transfer templates from one computer system to another using the export and import features.

Template tabs reference

When you create a template, the data entered on the template tabs corresponds to different tabs when you create an instance from that template. Some tabs determine behavior in the instance, but do not correlate to specific tabs.

Table 15. Template and instance tab reference

This table describes which template tabs correspond to which object instance tabs.

Template	Instance	Notes®
Properties tab	Summary tab	
Attributes	Summary tab	On this tab, you can customize labels, add help-tips, and determine if fields are Standard, Required, or Hidden on the instance's Summary tab.

Table 15. Template and instance tab reference

This table describes which template tabs correspond to which object instance tabs.

(continued)

Template	Instance	Notes®
		You can also localize Summary tab fields on this tab.
Tabs	Summary tab or custom tabs	Add forms on this tab. You can add forms to the bottom of the Summary tab or add custom tabs containing forms to the instance.
Attachments	Attachments	This tab is not available on Invoice or Asset templates.
Custom Links	Summary tab or previously created custom tabs	This tab is not available on Invoice or Asset templates.
Customize Alerts	Not visible	This template tab determines behavior in the instance. It does not directly correspond to a tab in the instance.
Budget Approval Rules	Not visible	This template tab determines behavior in the instance. It does not directly correspond to a tab in the instance. This template tab is only applicable to Project, Program, and Invoice templates.
Project Roles	Not visible	This template tab determines behavior in the instance. It does not directly correspond to a tab in the instance. This template tab is available only on Project templates.

Table 15. Template and instance tab reference

This table describes which template tabs correspond to which object instance tabs.

(continued)

Template	Instance	Notes®
Request	Not visible	This template tab determines behavior in the instance. It does not directly correspond to a tab in the instance. This template tab is available only on Project templates.
Workflow	Workflow tab	This template tab is available only on Project templates.
Campaign	Adds Implementation button ()	This template tab is available only on Campaign Projects whenUnica Plan and Unica Campaign are integrated.

Effects of template changes

When you edit a template, be aware you are changing all instances of objects that were previously created from the template.

The exceptions are workflow, metrics, and attachment folders. When you change the workflow or metrics template for an object template, or add or remove an attachment folder, your changes apply only to objects you create after the changes are made. Existing workflows are not changed, nor are the attachment folders nor metrics for any existing projects, programs or plans.

Template Properties tab for defining the template

For all objects, the template Properties tab contains properties for you to set, such as name, description, and security policies.

The properties that you set on this tab display on the Summary tab when you create a plan, program, or project. More properties display on this tab for project templates.

Table 16. Properties for all templates

Property	Description
Name	The display name for the template, which displays on the **Templates** list page.
Description	Short description of the template. Displays on the template selection page when users add a marketing object instance.
Icon	Large and small icon images for the template. The large icon displays when users create an instance that is based on this template. The small icon displays next to the template name in the **Templates** list page. Click **Change Icon** to import different image files.
Security Policies	List of security policies that determine which users have access to the template. **Note:** Different fields display for project templates.
Template ID	Internal identifier for the template. Use lowercase alphanumeric values only. Do not include accented or non-roman characters or spaces. **Note:** • For plan and invoice templates, this field is display-only. There is only one template for plans and one for invoices; their IDs cannot be changed. • Template IDs must be unique across Unica Plan object type. For example, you cannot have two project templates, both having an identical ID, such as *tradeshow*. If you have two

Table 16. Properties for all templates (continued)

Property	Description
	trade show project templates, use a different ID for each one, such as *tradeshow01* and *tradeshow02*. Additionally, after you use a template ID, you cannot use it again, even if you delete it. • You can edit this field only until a user creates the first object instance that is based on this template.
Default Name	The default name to give to an object instance (such as a program, asset, or marketing object) created from this template. For auto-created marketing objects, this name is part of the unique name that is generated when the system auto-creates the marketing object. You can leave this field blank.
ID Prefix	Prefix for the external ID of the object. Each plan, program, project, or marketing object in Unica Plan has an external ID assigned to it. For example, the ID for the first project might be 1001. You can set the ID prefix by template, to easily determine the template on which an object is based. For example, you choose an ID Prefix of **TRS** for the Tradeshow project template. The first tradeshow project that you create would then have an ID of TRS1001.
ID Generation Class	Java™ class to specify a numbering algorithm for objects. By default, Unica Plan assigns a sequential number to each object (plan, program, or project). However, you can configure Unica Plan to use an algorithm that you define to set the external ID. If you choose this configuration option, the ID Generation Class specifies the Java™ class that is used to generate the code. You must edit this attribute only if you want to generate IDs according to an algorithm other than the default.

Table 16. Properties for all templates (continued)

Property	Description
Metrics	For processes (projects, programs, and plans), the metrics template that is used for the object. You can select any metrics template available from the list.

To export metadata for an individual template, click **Export Template** at the top of this tab. See Exporting a single template (on page 109).

In addition to the properties that apply to all templates, templates for projects contain the following properties.

Table 17. Properties for project templates

Property	Description
Security Policy Use Model	Specifies how a "use" security policy is determined when a project request becomes a project. When the value of this field is **User Security Policy**, the **Use Security Policy** field on this tab is disabled. The person who creates a project or request from this template specifies the "use" security policy when the item is created. When the value of this field is **Template Security Policy**, the **Use Security Policy** field on this tab is enabled and the template developer selects the "use" policy.
View Security Policies	Specifies the security policy that determines which users can select this template when they create a project or request.
Use Security Policy	Specifies the security policy that determines which users can access the projects or requests after they are created.
Project Health Status Rule	Selects a rule for calculating project health. For more information about rules, see Implementing project health rules *(on page 295)*.
Export Tab	Select the project tab to export when you export the Calendar. You can choose the Summary tab or any custom tab.

Table 17. Properties for project templates (continued)

Property	Description
	When users export a Calendar, a link to the specified tab and the data for the tab are included with the exported calendar data. Users can click the link to view the data that is exported for the project.
Enable Automatic Addition of Delegate to the People tab	When a user is out of the office, a delegate can be specified to cover tasks, approvals, and requests. Used to override the system-wide setting at the project template level. • If set to yes, the system automatically adds the delegate as a project team member (if necessary) when a task, approval, or request is assigned to the delegate. • If set to no, users can select only a delegate who is a team member for all of the same projects. For information about the system-wide setting, see Administrative Settings page (on page 8). For information about the out-of-office feature, see the Unica Plan User's Guide.

Attributes tab

You can customize the standard attributes on plan, program, and project templates for your organization. You can determine whether the attributes on the Properties tab are required, standard, or hidden by editing the Attributes tab. You can also add a custom help-tip, which displays on hover. You can also localize the attributes and the corresponding help-tip. This customization determines how the standard attributes display on the Summary tab when you create a program or project.

1. Edit a plan template. Create or edit a program or project template. Open the attributes tab.
2. To change the label of the attribute in the default locale, click the edit icon in the **Label** column. Type the new label in the text box.

3. To add a help-tip, click the edit icon in the **Help-tip** column. Type the help-tip in the text box.
4. To set the attribute to Required, Standard, or Hidden, click the edit icon in the **Display type** column. Select Required, Standard, or Hidden from the drop-down menu.

   Note: On the plan template, the following attributes are always set to Required: Name, Plan Code, and Program Areas. On the program template, the Name and Program Code are always set to Required. On the project template, the Name and Project Code are always set to Required.

5. To localize the attribute label and help-tip, click the edit icon in the **Language** column.

   Note:

   The **Language** column appears only if your organization supports multiple locales.

   Each locale your organization supports displays in the **Modify details for name** window. If you do not see a specific language, that locale is not supported.

   a. Enter translated text for each locale.
   b. Click **Save changes**.
6. Click **Save changes**.

Note: For more information about localizing templates, see Multi-locale support for templates *(on page 249)*.

Exporting a single template

You can export a single template by clicking on the template you want to export and clicking **Export Template**.

1. From the **Settings** menu, select **Unica Plan Settings**.
2. Click **Template Configuration**.
3. Click **Templates**.
4. Click the name of the template to export.

   The Properties tab displays.
5. Click **Export Template**.
6. Specify the **Database Type** of the system that receives the template metadata through an import operation. The selected database type determines the format of the SQL script files that are generated by the export process.
7. Click **Export** to export the template, or **Close** to cancel the export; skip the remainder of the instructions.
8. On the File Download dialog, click **Open** or **Save**.

   The system creates a compressed archive that contains the XML and SQL script files for the selected template. Open or extract the archive file to view these files.

Template Tabs tab for customizing the user interface

Use this tab to add forms to the **Summary** tab, or to create custom tabs for the marketing object types that support this feature.

For example, you want to collect information about the outside vendor that users plan to contract with to get collateral printed. To do so, you create a form with an attribute for users to select a printing company from a list of several vendors. You also include a text box for users to enter the quoted price of each page of the collateral. You then use the Tabs tab in a project template to add a custom tab and name it **Printing**.

Tip: You cannot rename the standard tabs that display for object instances.

When you configure security policies for your organization, you can configure custom security permissions for these tabs.

Table 18. Fields for defining a new tab

Field	Description
Display Name	The display name for the new section on the **Summary** tab or the new custom tab.
Page Style	• Select **summary** to add a form to the bottom of the **Summary** tab. Use this option for forms that contain relatively few attributes and forms that users need to see frequently. • Select **tab** to create a custom tab specifically for the form. Use this option for forms, or groups of forms, that require their own page, as in the **Printing** tab example. This option is not available for all template types.
Form	Select the form to add.
Click icon to add rule	Each section on a tab can be expanded to show all fields, or collapsed to hide fields until users want to work with them. By default, all sections are expanded. You can build a rule that uses an if-then statement to specify conditions in which the section is expanded. If you specify a rule, the form is expanded only when the conditions of the rule are satisfied; otherwise, the form is collapsed. For more information, see Building rules to show and hide forms (on page 114).
Data Validation Class	System-supplied when you select a **Data Validation Rules** file.
Data Validation Rules	Forms that use a grid to collect data can apply an XML file with validation functions to edit check user entries. If the form that you specified in the **Form** field has one or more associated rules files, you select one from this list.

Table 18. Fields for defining a new tab (continued)

Field	Description
Visibility	Select **Show in Wizard** to display the new Summary section or custom tab in the series of pages that display when users create a marketing object. If you clear this option, the section or tab displays only after users save the object. For project templates only, select **Show in Request** if this new **Summary** tab section or custom tab applies to both projects and requests. If you clear this option, the section or custom tab displays for projects only.

Adding tabs to templates

Add a tab or form to templates to collect information about the marketing object. You can add more forms to the Summary tab. For some types of templates, you can add custom tabs, on which you add forms.

1. Add or edit the template then click its Tabs tab.
2. Click **Add Tab**.
3. Enter a descriptive name for the tab in the **Display Name** text box.

   The name that you choose becomes the name of the tab that users see when they create instances from this template.
4. Choose whether to show the form on the **Summary** tab or its own custom tab.

   This option is not available for all template types.
5. You can create rules to show and hide the forms. See Building rules to show and hide forms *(on page 114)*.
6. Select a form from the **Form** list.

   This list contains all the forms available in Unica Plan, except for the TCS® (Target Cell Spreadsheet) forms used by campaign projects.
7. If you are adding a grid, you can select a data validation rule from the **Data Validation Rules** list.

For more information, see Advanced topics *(on page 322)*.

8. Select the visibility options for the tab.

   This option is not available for all template types.

9. Click **Save changes** to save the tab, or **Add Tab** to add another tab.

Moving tabs and forms on templates

You can change the position on forms on tabs. You can also change the order of custom tabs as you change and refine templates.

1. Edit the template then click its Tabs tab.
2. Click one of the following buttons after **Move**.
   - **Down** to move down the tab or form. Moving a form on the Summary tab down places it lower on the object's **Summary** tab. Moving a custom tab down places it further to the right in the tab list. For example, if the tab was fourth in the list, moving it down once makes it fifth.
   - **Up** to move up a tab. Moves it up or forward one position.

Note: On the Summary tab, the standard form for instance properties must come before any custom forms.

Deleting custom forms and tabs from templates

You might want to remove forms from tabs or custom tabs from templates. Consider whether you want to change the template by deleting the form or create a new template. If you delete a form or tab, data in existing object instances is lost.

Important: Deleting a custom form or tab from a template also deletes it from all existing object instances that are created from the template. Do not delete a custom form or tab from a published template if users already created object instances from it. If you do, data is lost.

1. Edit the template then click its Tabs tab.
2. Scroll to the section that defines the form or custom tab you want to remove and click **Delete** (on the right side of the page).

   The form or custom tab is removed from the object template.
3. Click **OK**.
4. Click **Save changes**.

Building rules to show and hide forms

When you add custom forms or tabs, you can show (fully expand) or hide (collapse) the forms when users create and edit object instances. Hiding a form that has user controls to collect optional information can streamline the process of creating an instance from the template. Users can expand a hidden form when needed.

Rules determine whether forms are shown.

- If you do not create a rule, the form is shown by default.
- If you do create a rule, you supply conditions for the "if" part of an if-then-else statement. If the conditions are met, then the form is shown; otherwise, the form is hidden.

1. Edit the template then click its Tabs tab.
2. Select the form on the Tabs tab then click **Add Rule** ( ).
   The **Rule Builder** dialog opens.
3. You create the rule conditions (the "if" part of the statement) at the bottom of the dialog. Select a form attribute, an operator, and a resource for each condition.
   For more information, see:Rule Builder dialog *(on page 115)*
4. Click **Add** to include each condition.
5. Use the **AND** and **OR** operators to build compound conditions. Click **Add** to include each one.
6. When the condition is complete, click **Save Compound Condition** to move it up into the **Compound Conditions** section of the dialog.

7. To see your rule as a complete if-then-else statement, and to verify that the logic is correct, click **Preview**. If you want, you can print the rule.
8. Click **Save and finish** to apply the rule.

Rule Builder dialog

Rules are if-then-else statements. In Unica Plan, you use the Rule Builder to assemble different types of rules for your templates: to expand or collapse a custom form by default, to assign the recipients for new project requests, or to require approval of a budget line item.

The following table describes the fields in the Rule Builder dialog.

Table 19. Controls for building rules

Field	Description
Compound Conditions	Lists the conditions that are built in the **IF** and **THEN** sections at the bottom of the Rule Builder dialog. To edit an existing condition, click **Update**.
IF the following compound condition is true	Lists each condition in the "if" portion of the rule. You use the controls below this field to build and add each condition.
Select an Attribute	Lists the standard and custom attributes that you can use for the rule, depending on where the rule applies. • For template rules this list does not include grid attributes. • For form attribute rules, this list only includes single-select database and single-select custom attributes from the form. It does not include grid attributes.
Operator	After you select an attribute, you choose an operator. The system populates this list based on the data type of the attribute.

Table 19. Controls for building rules (continued)

Field	Description
	For example, you select the Description attribute. The list of operators is **Starts With**, **=**, **contains**, and **Ends With**. For an Amount attribute, the operators are **<=**, **>=**, **=**, **>**, **<**, and **!=**.
Value	After you select an attribute and operator, you specify the value. If your "if" statement is complete, click **Add** to save it. To include another condition, continue to the next field.
And/or	For a rule that contains more than one condition, select **And** or **or** to connect the current condition with the next condition. Then, click **Add** to save this condition and click **New** to specify the next condition.
THEN	Specifies the result when the condition (or conditions) that you entered is met ("then"). • For a rule to show or hide a form, the result is Show. • For a rule to require approval of a budget line item, you assign the approver. • For a rule to assign the recipient for a project request, you select a user. After the **IF** list displays the complete condition and you specify the result, click **Save Compound Condition**. The complete if-then statement displays in the **Compound Conditions** section at the top of the dialog.
Default Action	Specifies the result when the condition (or conditions) is not met ("else").

Table 19. Controls for building rules (continued)

Field	Description
	• For a rule to show or hide a form, the default action is "Hide". • For a rule to require approval of a budget line item, the default action is "No approval required". Line items are automatically approved. • For a rule to assign the recipient for a project request, you select a **Default Resource** to assign if none of the conditions are met.

To review or print the complete if-then-else rule, click **Preview**.

For more information about adding different types of rules, see:

- Building rules to show and hide forms *(on page 114)*.
- Example: building a template request rule *(on page 135)*.
- Building budget approval rules *(on page 127)*.

Template Attachments tab for adding folders and files

Use this tab to manage attachments in your templates. You can add attachments and folders for future attachments.

On this tab, you:

- Add one or more attachment folders so that users can add and organize attachments: Click **Add Folder**.
- Change the relative position of the folders on the tab: Click **Up** and **Down** to reorder attachment folders.
- Delete folders: Click **Delete** next to the folder you want to remove. All attachments in the folder are also deleted.

- Attach one or more files to the template, so that whenever an object is created from the template, certain images and documents are attached by default. Click **Add Attachment**.
- Delete default attachments: Click **Delete** next to the file you want to remove.

To add folders and default attachment files:

1. Add or edit the template then click its Attachments tab.
2. Add at least one folder to the tab: Click **Add Folder** and provide a **Name**.
3. To add an attachment, click **Add Attachment** next to a folder.
   The **Upload Attachment** dialog opens.
4. Enter the file path and name, or click **Browse** to locate the attachment.
5. Click **Save** to attach the file.

   The attachment file displays in a list under its folder.
6. On the Attachments tab, click **Save changes** to save the new folder and its attachment.

   Repeat these steps to add as many folders and attachments as needed.

Template Custom Links tab for accessing other websites

Use this tab to create custom links that display on one or more tabs for the object instances that are created from this template. For example, you might want to link to an application that your organization uses to generate ID codes for collateral pieces or direct marketing offers.

To add a custom link, click **Add a Custom Link**. For each link, you can add a parameter, such as the query portion for a dynamic URL. To form the complete URL for the link, Unica Plan appends a question mark (?) to the end of the URL you supply and then adds the parameter. To include a parameter, click **Add a Parameter**. More fields display for the parameter.

This tab contains the following properties. When your work is complete, click **Save changes**.

Table 20. Properties for custom links

Property	Description
Display Name	Enter a name for the link. This value becomes the name of the link.
ID	Enter a unique internal ID for the custom link.
Description	Enter some descriptive text for the link. This text is shown as a tip when a user moves the cursor over the link.
URL	Enter the fully qualified URL (including http://) of the website to open when users click the link. The website opens in a new browser window.
Tab Visibility	To define where this link displays to users, select one or more tabs. The link displays at the bottom of every selected tab. You can select the Summary tab and the custom tabs previously added to any template of the same type.
Options	For project templates, you can make the link display for both requests and projects that are created from the template, or only for projects. Select **Show in Request** to display the link for both.

Custom link URLs can include a parameter. For example, the link can open one page for an object that is Not Started, and another page for an object that is In Progress.

Table 21. Fields for custom link parameters

Property	Description
Name	Enter a name for the parameter.
Value	Select a value from the list to define the type of object property to use. Depending on your selection, another **Value** list can display to collect more specification. The choice that you make determines the options available from the subsequent list.

Table 21. Fields for custom link parameters (continued)

Property	Description
	For example, if you select a value of **<object type> Properties**, a list displays values including Target Start, Actual End, Security Policy, and other relevant properties for the object.

You can add more than one custom link: Click **Add a Custom Link** to display a new set of property fields. If you add more than one custom link, you can change their relative positions: Click **Up** and **Down** to reorder the links.

Template Customize Alerts tab to set up event-triggered alerts

Use this tab to specify events to trigger alert messages.

When a specified event occurs in an object instance that is created from this template, the system automatically sends a notification. For each event you specify, you can customize the locale, subject, message text, header, and footer of the notification message.

For more information, see Customizing an alert notification message (on page 282).

Icons page

On the **Icons** page, you review and add icon files. These icons display in various parts of Unica Plan user interface to identify a type of template or object instance.

Click the **Icons** link on the **Template Configuration** page (or **Change Icons** on the template Summary tab) to manage the icons that identify templates and the objects that are created from them.

The **Icons** page contains the following columns:

Column	Description
Images	A large and small image for each icon. Click the images to change the icon name or the image files.

Column	Description
Name	The name of icon.
Used By	A list of object templates that use this icon. You specify the icons that are used by a template on the template Properties tab. See Template Properties tab for defining the template *(on page 104)*.
Delete	A link to delete the icon image file. This link is only available for icons that are not used in any templates.

Unica Plan is installed with a set of default icons. You can choose from these icons, or add icons that are customized for your organization. When you add your own custom icons, you upload two file sizes for each icon:

- File image: the large (46x54 pixels) image that displays when users create an object instance.
- List icon image: the small (20x24 pixels) image that displays on the list page next to an object instance. For example, the Project list page contains the list icons for all of the projects on the page.

The image files must be in JPEG, PNG, or GIF format.

Adding and editing icons

You can add or edit icons to use on templates.

1. Click **Settings > Unica Plan Settings > Template Configuration**.
2. Click **Icons**.
3. To add an icon, click **Add icon**.

   To edit an icon, click the icon image (large or small).
4. Enter a **Name** for the type of template and object the icons identify, such as the name of a type of project.
5. Upload the icon image files:

- To upload a main icon, under **File Image** enter a path and file name or click **Browse**.
- To upload a list icon, under **List Icon Image** enter a path and file name or click **Browse**.

To replace an existing file, you must select the check box next to the type of icon you want to change.

6. Click **Continue** to load the files into Unica Plan.
7. Click **Save changes** to confirm the upload.

The new or edited icon displays in the list.

Template validation

Validate your templates to check for errors.

Unica Plan offers two types of predefined template validation checks:

- Database validation
- Attributes validation

You can perform these validation checks on all of your templates at any time: on the **Template Configuration** page, click **Validate Templates**. Information about these validation checks follows.

The system can also include more validation procedures if defined by your installation. For more information, see Grid validation (on page 337).

Database validation

Database validation checks the validity of the database schema, and whether form attributes match their data type in the database.

The system validates the database when you import, upgrade, and export templates. When you export templates, only forms that are not linked to any template are validated.

When you import and upgrade, you can save templates even if they are invalid. You receive a warning, but you can still save. However, when you add a form, you cannot save the form if validation finds any errors.

Attribute validation

Unica Plan has two types of validation to check for inconsistencies in forms.

When you save templates, the system validates the attributes. Unica Plan has two types of attribute validation: template attributes validation and template attribute type validation.

- Template attributes validation: Checks whether two or more form attributes point to the same table column in both Summary tab and other, non-Summary tabs. If two or more form attributes on different tabs point to the same table column, the system generates an error message that describes the duplicated references.
- Templates attributes type validation: Checks whether two form columns, of different types, from two templates point to same table column but with different type. If two or more form attributes with different types point to the same table column, the system generates an error that describes the inconsistency.

Chapter 9. Program and project templates

Program and project templates have additional tabs that customize the object instances that are made from the template.

If you have the Financial Module installed, program and project templates can contain budget information. You can set budget approval rules to manage expenses on the Budget Approval Rules tab.

You can set participant roles on the Project template's Roles tab.

Your organization might restrict your users from creating project instances. In that case, users must create project requests. Manage default rules for project requests on the Request tab.

Project templates contain a workflow tab. You can build a default workflow for the project instances built from this template.

If Unica Plan is integrated withUnica Campaign, you can create Unica Campaign projects. Unica Campaign projects use features from both products.

Budget tab customization

To customize the Line Items Details tables on the Budget tab for programs and projects, you can add up to three text columns. This change is global; the text columns you specify appear for all Line Items Details tables in all programs and projects in Unica Plan. Changes to the Budget tab apply to both new and existing programs and projects.

You can also choose to include a Vendor column in the Line Items Details tables on the Budget tab. You can enable this column for programs, for projects, both, or neither.

Note: Budget tabs are a feature of the Financial Management module. If you do not have the module, these options and controls are not visible.

Adding text columns to the Line Items Details table

You can add text columns to the Line Items Details table when you must capture more information about your budget.

1. Select **Settings > Plan Settings** and click **Budget Line Item Columns**.
2. To add a column, enter the label that you want as the column heading.
   Entering label text automatically enables the column.
3. Click **Save changes**.

The new columns appear on the Budget tab in the Line Items Details table for all programs and projects. To edit the column labels, repeat these steps at any time.

Disabling or removing text columns

You can temporarily disable a text column or remove it permanently if the information you want to capture about your budget changes. Click **Settings > Unica Plan Settings > Budget Line Item Columns**.

To disable a text column from the Line Items Details tables without deleting the column label completely, click **Disable**. After you disable a text column, the **Disable** option is replaced with an **Enable** option.

To delete a text column from the Line Items Details tables, click **Remove**. Click **Save changes**. The text column fields are renumbered if necessary and new fields are added at the bottom to keep the total number of available text columns at three.

Vendor column for budgets

The vendor column can track suppliers and the amounts your organization owes them.

If you want the Vendor column to be visible when users are editing budget line items, you set the **FMPrgmVendorEnabled** and **FMProjVendorEnabled** properties to `True` under **Settings > Configuration > Plan > umoConfiguration**.

Template Budget Approval Rules tab for automating line-item approvals

You can write approval rules on the Budget Approval Rules tab of program, project, and invoice templates to streamline the approval process. Using the rule builder, you can define rules that automatically approve budget and invoice line items.

Each time a line item is added or edited, Unica Plan checks it against the approval conditions that are set in the template. If the line item meets the conditions, it triggers an approval process. If the line item change does not meet the conditions, it is approved automatically.

Note: If you do not write any rules, Unica Plan does not require any approvals.

Each line item requires a separate approval. Depending on the conditions that are set, one line item can trigger multiple, parallel approvals from multiple approvers.

To build rule conditions, you can use any attribute in any form in the template. For projects and programs, you can also build conditions that are based on the following budget attributes.

- Expenditure date
- Source account
- Cost category
- Committed amount
- Forecast amount
- Vendor name

For invoices, you can also build conditions that are based on the following line item attributes.

- Source account
- Cost category
- Cost per unit

- Quantity
- Total cost

For more information about the approval process, see the Unica Plan User's Guide.

You can build rules for Unica Plan placeholder users.

- Program templates: program owner and account owner
- Project templates: project owner and account owner
- Invoice templates: invoice owner and account owner

These placeholder users can then be mapped to real users in your organization.

Building budget approval rules

To define rules and require approval for items that meet them, you write if-then-else statements.

You write rules to identify situations that require an approver to review line item changes.

- If you do not write any rules, no approvals are required.
- If you do write rules, only line items that meet the conditions that you specify require approval.

1. Edit the template then click its Budget Approval Rules tab.
2. Click **Add Approval Rule**.
   The Rule Builder dialog opens. See Rule Builder dialog *(on page 115)*.
3. You create the rule conditions (the "if" part of the statement) at the bottom of the dialog. Select a form attribute, an operator, and a resource for each condition.
4. Click **Add** to include the condition.
5. Use the **AND** and **OR** operators to build compound conditions. Click **Add** to include each one.
6. From the **Assign Approver** list, select an approver to receive the request when the condition is met ("then").

To assign the approval to the owner of the object instance or the account, select one of the Object Owners. The system assigns the approval to the user who is the actual owner when you select one of these options.

7. When the condition is complete, click **Save Compound Condition** to move it up into the **Compound Conditions** section of the dialog.
   If the condition is met, the approval process is triggered. If the condition is not met ("else"), no approval is required.
8.
9. To see your rule as a complete if-then-else statement, and to verify that the logic is correct, click **Preview**. You can also print the rule.
10. Click **Save and finish**.
    The Rule Builder closes.
11. When you are finished building rules, click **Save Changes** on the Budget Approval Rules tab.

    You can build multiple rules that result in multiple, parallel approval processes.

For each rule, if the condition is met, the line item approval request is sent to the assigned approver. If the line item does not meet the conditions for any rules, it is automatically approved.

Editing budget approval rules

You can edit budget approval rules as the needs of your organization change.

1. Add or edit the program, project, or invoice template, then click its Budget Approval Rules tab.
2. Click **Build Rule** ( ) in the **Edit Rule** column of the rule you want to change.
   The Rule Builder dialog opens. See Rule Builder dialog *(on page 115)*.
3. To change the sequence of the conditions, select one of the conditions then click **Up** or **Down**.
4. To delete a condition, select it then click **Delete**.
5. To add a condition, click **Add Compound Condition** and follow the steps in Building budget approval rules *(on page 127)*.

6. To modify a condition, select it then click **Update**.
   The condition displays in the work area of the Rule Builder. Click the condition to edit the attribute, operator, or value for the condition, or to select a different user or role from the **Assign Approver** list.
7. After you add or modify a condition, click **Save Compound Condition**.
8. To see your rule as a complete if-then-else statement, and to verify that the logic is correct, click **Preview**. You can also print the rule.
9. Click **Save and finish**.
10. Click **Save changes** on the Budget Approval Rules tab.

Deleting budget approval rules

You can remove budget approval rules as you refine your templates.

1. Edit the program, project, or invoice template then click its Budget Approval Rules tab.
2. Select the check box next to the rule you want to delete.
3. Click **Remove Selected Rule(s)**.
4. Click **OK** to confirm that you want to delete the rules.
5. Click **Save changes** on the Budget Approval Rules tab.

Project template Project Roles tab for including participants

Use this tab to specify the project roles of the people who participate in the projects and project requests that are created from this template.

Before you can add a project role on the Project Roles tab, it must exist in the system. To create a project role, select **Settings > Plan Settings > List Definitions > Roles**. For more information, see List definitions (on page 289).

The security policies for your organization can be configured to only allow team members with a specific role to access projects and requests. For more information, see Project roles (on page 262).

This tab contains the following sections.

Section	Description
Project Request Recipients	The project roles of the team members who receive the requests that are created from this template. The values that you specify in this field on the Project Roles tab display in the **Recipient Role** field on the Request tab. You use the template Request tab to configure how requests are handled.
Team Members	The project roles of the people who participate in projects that are created from this template. Users assign these project roles to tasks on the Workflow tab.
Reviewers	The project roles of the people who participate as reviewers. Users assign these roles as reviewers in the projects that are created from this template.

To add a role, click the **Name** list in the appropriate section and select a role. The list presents all of the roles that are defined in the Roles list. When roles are displayed to users, they are displayed in alphabetical order and not in the order that you add the roles.

Tip: If you import a workflow template on the Workflow tab, any roles that are defined in the workflow template are automatically added to this tab.

To delete a role, click **Remove** next to that role. You cannot delete a role that is specified in an object instance, either in a workflow task or as a recipient on the Request tab.

Project template Request tab for configuring the request process

Users who do not have permission to create instances from the template can submit a request to create an instance. Then, the request must be approved. Use this tab to determine who receives requests and how requests are processed.

Use this tab to set up the following conditions for requests that are created from this template.

- Request recipients or how request recipients are designated.
- The order in which recipients receive request notifications and in which they must respond to requests.
- The length of time that recipients have to respond.
- The way reapprovals are handled.

You use the Rule Builder on the Request tab of a project template to set up conditions that determine who receives a project request. Note the following behavior.

- A project request that is created from a template that has recipient rules uses all of the rules you set up with the Rule Builder. You can set up multiple rules.
- If you change the recipient rules in the template, it affects the behavior of all existing requests that are created from the template.

Note: Only changes to recipient rules affect the behavior of existing requests. Any other changes that you make to a project request template are reflected only in new requests.

Request tab fields

Use the Request tab to set up project request recipients and how the request is processed.

Descriptions of the fields on the project template Request tab follow.

Setup Project Request section

The following table describes the fields in the Setup Project Request section.

Table 22. Fields in the Setup Project Request section

Field	Description
Request description	The description that displays when a user adds a project request. Briefly describe the purpose of the template. Length is limited to 300 characters.
Request reapproval rule	Select one of the following options to define how project requests are handled when they are returned and then resubmitted. • If the project request is returned and then resubmitted, request is processed by all recipients again (the default). • If the project request is returned and then resubmitted, start process with the person who rejected. • If the project request is returned and then resubmitted, request owner selects the recipients that it needs to go to. In this case, on resubmitting the request the owner of the request can select only required recipients who accepted the request.

Setup Recipients section

The following table describes the fields and user interface controls in the Setup Recipients section.

Table 23. Fields in the Setup Recipients section

Control	Description
Request owner can add and/or delete recipients	If you do not add recipients, you must leave this check box checked or you receive an error message when you save the template. If this check box is checked, a project request that uses this template allows the requester to assign new recipients

Table 23. Fields in the Setup Recipients section (continued)

Control	Description
	and change any non-mandatory pre-configured recipient assignments.
Add recipient step	Click **Add recipient step** to add a row to the grid. Each row provides a set of fields that you configure to add request recipients.
Recipient Role	A drop-down list that contains the recipient roles you configured on the **Project Roles** tab. The roles are displayed in alphabetical order.
Recipient Assignment	A drop-down list that enables the following options. • User/Team: This option enables a drop-down list of users from which you select a user or team to assign to the role you selected in **Recipient Role**. If you select a team, the recipient is a team member or team manager (who assigns the request to a team member); this is determined by the option that is selected in the Request Routing Model section of the Team's summary tab. • Requester Assigned: Allows the requester to assign a user to the role you selected in the **Recipient Role** field. If you set the other fields (such as Default Duration, Sequence, and Project Owner), these values become the defaults for this request recipient. The requester can change the defaults. • Rule Driven: This option enables an icon that you click to open the Rule Builder. Then, you define rules for assigning a user to the role you selected in the **Recipient Role** field. See Rule Builder dialog *(on page 115)* for a description of the Rule Builder.

Table 23. Fields in the Setup Recipients section (continued)

Control	Description
Default Duration	Time that is allowed for each review step. The way days are counted is set up when Unica Plan is installed and configured, in the **numberOfHoursPerDay** property. See the Unica Plan Installation Guide for a description of the options that are provided for this setting. If recipients do not respond within the time that is allowed as their default duration, they receive an alert. If the recipient is a team, alerts are sent according to the Request Routing Model configured for that team.
Sequence	A field in which you enter a sequence number. Select the sequence number for each recipient to specify the order in which the recipient receives notification of the request and must approve the request. You can control whether recipients act in parallel with other recipients or before or after any other recipient. If you assign the same number to multiple recipients, they all receive notifications when it is their turn to respond. This field must contain a number; the maximum is 99. By default, each time you add a recipient the value of this field is incremented.
Project Owner	The recipient that is designated as the project owner becomes the owner if the request is accepted by all required reviewers. The project owner is always a required recipient.
Required	A check box that determines whether a recipient is required. Check the box next to each recipient who must approve. If this box is not checked, the recipient is optional. Note the following behavior for required recipients.

Table 23. Fields in the Setup Recipients section (continued)

Control	Description
	• If a recipient is required, the next recipient in sequence is not notified (and cannot respond) until the current recipient responds. • If a recipient is required, and that recipient denies the request, then the next sequential recipient is not notified. Then, the request is placed on hold and the owners are notified. • If multiple recipients act simultaneously, and one of the required recipients rejects the request, then the request process continues until all the required recipients who work simultaneously respond. After all responses from that step are complete, the system sends a rejection notification to the request owner and any recipients who previously responded. • At least one recipient must be set to Required. The system generates a warning message if someone tries to start a request that contains no required recipients.
Instructions	Opens a dialog for you to add instructions that this recipient sees on the Summary tab of the project request. Length is limited to 1024 characters.
Remove Selected Recipient Step(s)	To delete a recipient step, select the check box for a row in the grid and then click this link.

Example: building a template request rule

This example describes how to build a rule for assigning reviewers to a project request.

Assume the following scenario.

- You are setting up a project template to use when your organization creates print advertising campaigns.
- The project includes a custom attribute that is named Region that specifies the region for a project. The regions are named NA (North America), APAC (Asia-Pacific), and EURO (Europe).
- You want different creative teams to review project requests by the region that the project is in.
- The project template Project Roles tab includes Project Manager as one of the **Project Request Recipient** roles.

Your rule controls which team member reviews project requests by region, and the requester cannot add other reviewers.

1. Edit the project template and click its Request tab.
2. In the Setup Recipients section, clear the **Request owner can add and/or delete recipients** box.
3. Click **Add recipient step**.
   A new row is added in the Setup Recipients section.
4. From the **Recipient Role** list, select **Project Manager**.
5. From the **Recipient Assignment** list, select **Rule Driven**.
6. Click **Build Rule** ( ).
   The **Rule Builder** dialog opens. See Rule Builder dialog *(on page 115)*.
7. In the **Rule Builder** dialog, do the following for each of the three regions:

   a. From the **Select an Attribute** list, select **Region**.

   b. From the operator list, select =.

   c. Type the region name (NA, APAC, or EURO) in the value field.

   d. Click **Add**.

   e. From the **THEN assign the following resource** list, select a region-appropriate team.

f. Click **Save Compound Condition**.

The if-then statement displays in the **Compound Conditions** box.

8. After you build a condition for each of the regions, select a **Default Resource** to receive the request if none of the conditions is met.
9. To see your rules as a complete if-then-else statement, and to verify that the logic is correct, click **Preview**. You can also print the rule.
10. Click **Save and finish**.

    The Rule Builder dialog closes and you return to the Request tab.
11. Complete the other fields for the recipient role. See Request tab fields *(on page 131)*.

Project template Workflow tab

The Workflow tab identifies and organizes tasks for a project. When you create a template, you can include data on its Workflow tab to provide an initial workflow structure for each marketing object instance that users create. To review and define a workflow on the template, you use a spreadsheet-style interface. Users access a similar spreadsheet interface to update the supplied workflow as needed for individual instances.

For more flexibility, the data on the Workflow tab in any template or any individual instance can be saved as a separate workflow template component. Template components, which can include metrics and forms in addition to workflows, make designing templates to meet different needs more modular and efficient. Any workflow template component can be imported into any project type template or instance.

Initially, the Workflow tab displays in view mode. To set up the stages, tasks, milestones, dependencies, and other data that make up the workflow, you switch to editing mode.

Note: When you design the workflow for a template, you can specify team member roles to associate with each task. To do so, you must first define values on the Project Roles tab. For more information, see Project template Project Roles tab for including participants *(on page 129)*.

For detailed information about configuring a workflow, including options for scheduling, calculating dates, and updating tasks, see the Unica Plan User's Guide.

Workflow template tab

On the Workflow template tab, you can edit, import, and save workflow template spreadsheets.

When you initially click the Workflow tab for a project template, it displays in view mode and offers the following user interface controls.

Table 24. Controls on the Workflow tab for projects

Control	Description
Edit link	Changes to edit mode so that you can configure the workflow for this project template.
Save as template link	Saves the data on this project template Workflow tab as a separate workflow template component. Then, you can reuse the workflow in other project templates.
Import Template link	Populates the Workflow tab of the project with the tasks, approvals, milestones, dependencies, and other values that are defined in a workflow template. This action replaces all the workflow values that were previously defined for this project.
Approval Options link	Opens a dialog where you identify the options in the Approval Deny Reasons list that are relevant to approvals generated from this template. **Note:** This option is available only for installations that require reviewers to specify a reason when they deny an approval.

Table 24. Controls on the Workflow tab for projects (continued)

Control	Description
View > View as a spreadsheet	**View as a spreadsheet**. Presents the workflow in spreadsheet format. Spreadsheet view provides access to granular information about each component in tabular format. When you view the workflow as a spreadsheet, each task name is a link. • Click an approval task to open the Setup Approval dialog, where you select approvers and indicate whether markup can be used. • Click a workflow task to open the Setup Task dialog, where you indicate whether attachments can be added for the task and add notes. Spreadsheet view is the default view.
View > View as a process flowchart	**View as a process flowchart**. Presents the workflow as a process flowchart, which shows each stage in the workflow as a set of interconnected process boxes.

Note: The advanced workflow feature with rework loops is not supported.

Adding workflow to templates

To set up a project template that includes workflow data, you create the template, specify team member roles, and then edit the Workflow tab.

1. Select **Settings > Plan Settings**.
2. Click **Templates**.
3. Locate the project template section and click **Add a template**.
4. Complete the fields in the **Template Properties** form and click **Save changes**.

5. On the **Project People** tab, specify participant roles then click **Save changes**.
6. Select the **Workflow** tab and click **Edit**.
7. Add workflow and approval tasks and organize them into stages.
   For more information, see Fields in workflow spreadsheets *(on page 142)*.

   Remember to save frequently while you work.
8. Edit task dependencies, define task scheduling and duration, and enter other values to supply when users create instances from this template.
   For more information, see the *Unica Plan User's Guide*.

   Remember to save frequently while you work.
9. When the workflow is complete, click **Save and Finish**. The tab returns to view mode.
   For more information, see Workflow template tab *(on page 138)*.
10. If your organization maintains a predefined list of reasons for denying an approval, specify the reasons that apply to this template.
    For more information, see List definitions *(on page 289)*.
11. To configure more information for each task, click the task name when the Workflow tab is in view mode.
    For approval tasks, you can configure approvers. For workflow tasks, you can configure whether users can add attachments.
12. Click **View > View as a process flowchart** to display the workflow as a process flowchart.

Configuring the workflow

Click **Edit...** to configure the workflow. Then, use the toolbar to add and delete stages and tasks. You can also move stages and tasks, manage dependencies, and access other options. After you configure the workflow, you can save it as a workflow template. You can reuse a workflow template in another project template as-is or import and edit it.

1. Open the workflow tab. Click **Edit...**.
2. Click **Add Stage Row** ( ). Stages organize workflow tasks and approvals. Add tasks and approvals under stages.
3. Select dependency options for the rows you are adding.

By default, tasks are added in series: each task depends on the task that precedes it.

To use a different dependency option, click and make a selection before you add the next task to the workflow.

- No Dependencies
- In Series
- In Parallel

Each task that you add uses the same dependency option until you change the selection.

You can also change the dependencies for a task manually: click the task name and supply a comma-separated list of task numbers in parentheses.

4. Add task and approval rows.
   - Click **Add Task Row** ( ) to add the workflow and people tasks you need.
   - Click **Add Approval Row** ( ) to add the approval tasks you need.

   You can copy and paste rows. You can also fill, or copy, information from one row to the rows above and below to improve efficiency. Click for these options.

   Note: To configure the task or approval, save and return to view mode. Then, click the name of the task or approval in view mode to edit it. You can decide whether users can add attachments to tasks and approvals. You can also add notes to the task or approval.

5. Organize your stages, tasks, and approvals. Click to move up rows. Click to move rows down.

   Note: When you move a row, Unica Plan does not change its task dependencies.

6. Configure rows in the workflow spreadsheet.

   You can configure the rows in the following ways:

- Give the stages, tasks, and approvals relevant names. Click the name of the row to edit it.
- Mark tasks and approvals **Required**. Users cannot skip, rename, or delete rows that are **Required**.
- Select **Enforce Dependencies** to make sure that users update only the task after all previous steps are completed.
- Select a **Milestone** type. For more information, see Customize milestone types *(on page 150)*.
- Set the date as **Anchored**. Anchored dates cannot move, even if a previous step is late.
- Set time durations for the **Default Duration** and **Target Effort** of the work.
- Set the time that the work will **Schedule Through**. By default, Unica Plan only schedules work on business days. You can calculate schedules to include weekends, holidays, or all calender days.

7. Click **Save** as you work. Click **Save and Finish** when you are done.

Fields in workflow spreadsheets

When you configure the workflow, you add a row to the spreadsheet for each task. Then, you edit the cells in that row to capture information for the task.

The following table describes each field and cell you use to enter data about the task on the workflow spreadsheet.

Table 25. Fields and cells on the Workflow tab

Field	Description
Task Code Prefix	When users create projects from this template, the system adds this identifier as a prefix to each task ID in the workflow.
Stages and Tasks	The cells in the first column display the names of the stages and tasks in the workflow. After each task name, the number of the task or tasks that it depends on displays in parentheses. For example, a task labeled "Estimate Costs (2.3)" depends on task number 2.3. To

Table 25. Fields and cells on the Workflow tab (continued)

Field	Description
	indicate more prerequisite tasks, enter task numbers that are separated by commas. For an approval task, the rework loop task is displayed in square brackets. For example, a task labeled "Cost Approval (2.4) [2.3]" depends on task 2.4. If approval is denied, the workflow loops back to task 2.3. The status for tasks 2.3 and 2.4 are is set as `Pending`. When you are in edit mode, you can click the name of a stage or task to revise its name and dependencies. When you save your changes and return to view mode: • Click the name of an approval task to open the **Setup Approval** dialog and select approvers. • Click the name of a workflow (or people) task to open the **Setup Task** dialog and indicate whether attachments can be added for the task.
Required	Indicates a required task. In the projects that are created from this template, required tasks cannot be skipped or deleted and their names cannot be changed.
Enforce Dep.	If this task depends on other tasks, determines how strictly the system interprets dependencies. When this option is selected, the system restricts project members from updating this task until the tasks it depends on are finished.
Member Role(s)	You can associate one or more roles with workflow tasks by default. To select a role, click in the field to display an alphabetical list of project roles that were previously defined as Team Members on the **Project Roles** tab for this project template. To define more than one role, click the name of each role.

Table 25. Fields and cells on the Workflow tab (continued)

Field	Description
	Note: To specify approvers for an approval task, return to view mode and click the approval task name to open the **Setup Approval** dialog.
Milestone Type	For tasks, you can specify an optional milestone type. Your system administrator configures the options available for you to indicate that a task is a type of milestone for the project. Example milestones include Job Start, Meeting, and Event. For information about setting up milestone types, see Customize milestone types (on page 150).
Anchored dates	Indicates whether the task is affected by automated date recalculation. Select this option for tasks with fixed dates that are not affected by any date changes made for other tasks.
Default Duration	The amount of calendar time to specify as the default for this task when a project is created with this template. To enter a duration, click in the cell then click the clock icon. Fields for days, hours, and minutes display.
Target Effort	The targeted effort to specify by default for this task when a project is created with this template. To enter the effort, click in the cell then click the clock icon. Fields for days, hours, and minutes display.
Schedule Through	Indicates how to calculate time when the task is scheduled. Unica Plan has the following options to calculate time for tasks. • Bus: Business days only, exclude non-work time and weekends. • Wkd: Weekends and business days, exclude non-work time.

Table 25. Fields and cells on the Workflow tab (continued)

Field	Description
	• Off: Non-work time and business days, exclude weekends. • All: Include every calendar day.

Considerations for importing and exporting form tasks in workflow templates

You can save a project workflow as a template to use in other projects. You can export the template from the project and import it into another project. If the workflow template contains some flowchart approval and you cannot import it in an non Campaign project.

The following situations apply to importing a saved project workflow template into a new project instance. Administrators can also import workflow templates into templates for projects if the workflow template does not contain form tasks.

- When you save a project workflow as a template with a linked form task and export it, then the link to the form is maintained. However, the form is not exported with the workflow template.
- You can import Approver steps as part of workflow import. This saves your time to set-up approvers again at both the template and project instance level.

The following situations apply to importing a project workflow template into a new project.

- If you import a workflow template and the linked form is present in the new project, then the link to the form works as expected.
- If you import a workflow template and the linked form is not present in the new project, then the link is broken.

- A warning lists form tasks with broken links. The form task is shown with an icon that indicates a broken link ( ).
- If you copy or clone the workflow, broken link copies as they are. The links remain broken in any copies.
- You can save the workflow with broken links. A user who tries to complete a task with a broken link cannot access the form.

If you encounter a broken link in the workflow, the project owner can open the form task pop-up and select an available form before assigning the form task to a team member.

Approval process and options

In Unica Plan, a structured approval process distributes information to team members and elicits their comments.

As the outcome of an approval, users select one of the following options.

- Approved
- Approved with Changes
- Denied

For more information about the approval process, see the Unica Plan User's Guide.

To edit the configuration properties for Approvals, click **Settings > Configuration > Plan > umoConfiguration > Approvals**. For more information, see the Unica Plan Installation Guide.

Denied option

To customize the approval process for your organization, you can require users to select a predefined **Deny reason** when they deny an approval. When users chose Denied, they must specify a **Deny Reason**. Administrative reports are available to help you analyze the denial reasons given.

For more information about setting up the approval denied reasons, see Setting approval deny reasons (on page 149) and Selecting approval options for templates (on page 149).

Approve with changes option

The Plan **approveWithChanges** property controls the availability of the **Approve with Changes** option for approvals. The **overrideApproveWithChanges** allows users to edit the **Allow approvers to approve with changes** option while configuring approval tasks.

Consider the following table to understand more about the Approve with changes options.

Table 26. Approve with changes

approve-WithChanges	overrideApprove-WithChanges	Creating an approval task	Completing an Approval task
False	False	The **Allow approvers to approve with changes** option is cleared on an approval task form and cannot be edited.	The **Approve with Changes** option is not available on the form when a user completes an approval task.
False	True	The **Allow approvers to approve with changes** option is cleared on an approval task form. Users can edit this option.	The **Approve with Changes** option is available to the user during approval if the **Allow approvers to approve with changes** option is selected when the approval task is created.
True	False	The **Allow approvers to approve with changes** option is selected on an approval task	The **Approve with Changes** option is available when a user completes an approval task.

Table 26. Approve with changes (continued)

approve-WithChanges	overrideApprove-WithChanges	Creating an approval task	Completing an Approval task
		form and cannot be edited.	
True	True	The **Allow approvers to approve with changes** option is selected on an approval task form. Users can edit this option.	The **Approve with Changes** option is available to the user during approval if the **Allow approvers to approve with changes** option is selected when the approval task is created.

Setting Approve with Changes options

You can set the approve with changes options from the configuration properties.

The **Allow approvers to approve with changes** option is applicable to all approvals tasks in project templates, projects, and stand-alone approvals.

To set or edit the approve with changes options, complete the following steps:

1. Go to **Settings > Configuration > Plan > umoConfiguration > Approvals**.
2. To allow users to approve a task with changes, set **approveWithChanges** to `True`.
   The **Allow approvers to approve with changes** option is now selected by default in the Approval form.
3. To allow users to edit the **Allow approvers to approve with changes** option when they create an approval, set **overrideApproveWithChanges** to `True`.
4. Restart the Unica Plan web application.

Setting approval deny reasons

To require a deny reason when users deny an approval, you set a configuration property and add options to the Approval Deny Reason list.

1. Set the **specifyDenyReason** configuration property to `True` under **Settings > Configuration > Plan > umoConfiguration > Approvals**. For more information, see the Unica Plan Installation Guide.
2. Restart the Unica Plan application server.
3. Add options to the customizable Approval Deny Reason list. For more information, see Customizable lists *(on page 289)*.
4. To enable this feature for stand-alone approvals, you select the Approval Deny Reason options that apply to this type of approval. Select **Settings > Plan Settings > Associate Deny Reasons with Stand-alone Approvals**.
5. To enable this feature for approval tasks, you select the Approval Deny Reason options that apply to each project template. For more information, see Selecting approval options for templates *(on page 149)*.

Selecting approval options for templates

If your organization requires users to specify a predefined reason when they deny an approval, you identify the set of reasons in the template. You can set different reasons for denying approvals in each project template.

1. To update deny reasons for a project template, open the project template Workflow tab.
2. Click **Approval Options**. The Associate Approval Deny Reasons with Template dialog opens.
   All of the options that are defined for the Approval Deny Reason list display on the left.

3. To select approval options for the template, click an available reason then click **>>**. To select multiple reasons, you can use Ctrl+click and Shift+click.
4. Click **Save changes**.

When a user denies an approval, a **Deny reason** must be selected.

Note: For more information about the approval process, see Approval process and options *(on page 146)*.

Customize milestone types

Unica Plan offers a set of default milestone types for use in project workflows. Users can select an option from the milestone type list when a workflow spreadsheet is in edit mode. You can also customize the milestone types.

The following options are supplied as milestone types.

- Checkpoint
- Meeting
- Event
- Drop Date
- Job Completion
- Job Start

You can customize the list of milestone types for your Unica Plan installation. Changes to the milestone type list affect every workflow on your system.

To customize the options that display, you select **Settings > List Definitions** and edit the Workflow Milestone Types list. For more information, see List definitions (on page 289).

You can add a custom milestone type to set a target and forecast dates for all tasks in a workflow in a project template. To do this, create a milestone with code START. Then, create a project template. In the workflow of this template, assign one task with this milestone type. When you create and save a project with a target start date that uses this template, the task with the new milestone type copies the project start date as its own target start

date as well as its forecast start date. These target and forecast dates are then rippled to all dependent tasks in the project.

The milestone type START can only be assigned to one task in a workflow.

Creating and editing workflow templates

The workflow that you create on the Workflow tab of any project template or instance can be saved as a workflow template. Then, you can reuse the workflow in similar projects.

1. In any project template or instance, select the **Workflow** tab.
   You can start with a new template or instance, or select one that already has a defined workflow.
2. For a new template or instance, change to edit mode. Set up the tasks, dependencies, and values that you want the template to supply as a starting point for new projects.
   For an existing template or instance, review the workflow.
3. With the Workflow tab in view mode, click task names to review or define settings.
4. Click **Save as template**.
5. Enter a descriptive name for the template and click **Continue**.
6. Click **Save**. The workflow template displays on the **Workflow Templates** page, and can be imported into any template or instance.

To edit a workflow template, open a project template and import the workflow template that you want to edit into its Workflow tab.

Typically, you set up a new template for this purpose because importing a workflow template overwrites any previously defined values. You can then edit the workflow and save the workflow as a new template with the same or a different name.

Workflow template branching

Many similar projects might require small variations in workflow. You can configure project templates to use different workflows that are based on user input instead of creating separate templates. Use this feature to simplify your organization's template collection.

For example, a project to organize a tradeshow contains an attribute called "Handout" with different options for handouts in the booth. The options are brochure, booklet, demonstration DVD, and 3D piece. The workflow tasks that are required to create each handout are different, although the rest of the steps for organizing the tradeshow are the same.

In this scenario, all the workflows contain a stage that is called "Create handout." You can use template branching to select the workflow for a print object when the user selects brochure or booklet. You might make both brochure and booklet go to a default workflow for a print handout. Then, you can branch the workflow to customize one workflow template for the demonstration DVD and a third template for the 3D piece.

Workflow template branching occurs only when a project is created.

You can build workflow template branching based on the following types of form and campaign attributes.

- Single-select
- Single-select database
- Yes/No

In this example, you build the template branching based on the single-select database attribute "Handout." The values for this attribute include brochure, booklet, demonstration DVD, and 3D piece.

Branching the workflow template

You can configure the project template to use a different workflow that is based on template attributes. For example, you can configure the template to use a different workflow based on what selection the user makes in the form attribute channel: phone, direct mail, or email. Understandably, the tasks for a project might change dramatically based on these options.

You must add forms to your project template before you branch the workflow. You must create workflow templates in your system before you branch the workflow.

You can build workflow template branching based on the following types of form and campaign attributes.

- Single-select
- Single-select database
- Yes/No

When a user creates a project, the workflow branches based on the how branching is defined in the template.

1. Open the project template and configure or import the default workflow.
2. Click **Template Branching**.
3. Select a form from the list that contains the attribute on which you want to build the template branching.
   This list contains forms that you previously added to the project template on the **Tabs** tab.
   Unica Plan populates a list of attribute values.
4. Select a form attribute from the list.
   Form attributes and attribute values are controlled on the form template
5. For each attribute value, select a workflow template from the list. You can select the default workflow, or any other workflow template on your system.
6. Click **Save and Return**.

Importing workflow templates into projects

You can import a previously saved workflow template into a new project instance.

Note: Administrators can also import saved workflow templates into project templates so every project instance has the same workflow. However, administrators cannot import saved workflow templates with form tasks into project templates.

1. Open Workflow tab of the project instance.
2. Click **Edit** ( ).
3. In Edit mode, click **Tools** ( ) and select **Import workflow and save** from the menu.

4. Unica Plan warns you that importing a workflow template deletes the current workflow. Click **OK** to continue.
5. Select a workflow template and click **Import**.

Unica Plan imports the new workflow, saves it automatically, and warns you if you imported form tasks with broken links or approval steps with invalid users, teams or rules.

Click **Edit** ( ) to modify the workflow you imported.

Considerations for importing and exporting form tasks in workflow templates

You can save a project workflow as a template to use in other projects. You can export the template from the project and import it into another project. If the workflow template contains some flowchart approval and you cannot import it in an non Campaign project.

The following situations apply to importing a saved project workflow template into a new project instance. Administrators can also import workflow templates into templates for projects if the workflow template does not contain form tasks.

- When you save a project workflow as a template with a linked form task and export it, then the link to the form is maintained. However, the form is not exported with the workflow template.
- You can import Approver steps as part of workflow import. This saves your time to set-up approvers again at both the template and project instance level.

The following situations apply to importing a project workflow template into a new project.

- If you import a workflow template and the linked form is present in the new project, then the link to the form works as expected.
- If you import a workflow template and the linked form is not present in the new project, then the link is broken.

- A warning lists form tasks with broken links. The form task is shown with an icon that indicates a broken link ( ).
- If you copy or clone the workflow, broken link copies as they are. The links remain broken in any copies.
- You can save the workflow with broken links. A user who tries to complete a task with a broken link cannot access the form.

If you encounter a broken link in the workflow, the project owner can open the form task pop-up and select an available form before assigning the form task to a team member.

Importing workflow templates

You can import an existing workflow template into your project template. Then, you can customize the workflow template for your new project template.

Before importing a workflow template that references role data in list definitions, ensure that the role data in the import and export environments is the same. For example, if you export a workflow template, import the template into another environment, and then try to change the template's role data in the import environment, the role data reverts to the data that was exported.

1. Create the project template.
   If the workflow template you plan to use includes project roles, you do not need to define them. The project roles are imported with the workflow template.
2. Select the **Workflow** tab.
3. With the Workflow tab in view mode, click **Import Template**.

   The system presents a warning displays that the import overwrites the existing workflow.
4. Click **OK**.

   A list of workflow templates opens.
5. Select a template from the list and click **Import**.

The Workflow tab displays the workflow tasks and stages from the workflow template. Any roles that are referenced by task rows also display on the **Project People** tab.

6. With the Workflow tab in view mode, click approval tasks to configure approvers.
7. To modify or add stages or tasks, click **Edit**.

   Then, customize the workflow as necessary for the project template. Remember to save your changes. For more information, see Configuring the workflow (on page 140).
8. When the workflow is complete, click **Save and Finish** to return to view mode.

Exporting workflow templates

You can export individual workflow templates to use in other project templates. In some cases, you might want to edit the exported XML file and reimport the workflow template into Unica Plan.

1. Select **Settings > Unica Plan Settings**.
2. Click **Template Configuration**.
3. Click **Workflow**.
4. Click the **Export** link for the workflow you want to export.
5. Choose a location to save the XML file, and save it.
6. Open the file with a text or XML editor, make your changes, and then save the file.
7. Navigate back to the templates library (**Settings > Unica Plan Settings**).
8. Click **Import Workflow Template** and browse to your edited XML file.
9. Name the file to differentiate it from the previous version.
   For example, if you export `Marketing Collateral`, you can name your edited file `Marketing Collateral 2`. You can always rename the file later.
10. Create a template and use the new workflow or open an existing template and replace the old workflow template with the new one.

Project template Campaign tab for communicating with Unica Campaign

Use this tab to configure communication from Unica Plan to Unica Campaign if integration is enabled.

Note: After a user creates a project from a template, you cannot change a non-campaign template to a campaign template or vice versa: The **Campaign Project template** option on this tab is disabled.

After a campaign template is used to create a project, the only option on this tab that can be changed is the **Metric Data Mapping** setting. To change any of the other options, you must first delete all of the projects that were created from this template.

The Campaign tab contains the following settings.

Table 27. Fields on the Campaign tab for project templates

Field	Description
Campaign Project template	Select this check box to mark this template as a campaign project template and display the other Campaign Integration fields.
TCS® Form	Select the form that contains the Target Cell Spreadsheet to use for projects that are created from this template. The drop-down list contains all published forms that contain a TCS®.
Metric Data Mapping	XML file that contains the data map for sending metrics from an Unica Campaign campaign to an Unica Plan project for reporting purposes.
TCS® Form Display Name	The name to display for the selected form on the TCS® tab.
Partition ID	Identifies the partition of the Unica Campaign instance in which to create the campaigns that correspond to the campaign projects created from this template.

Table 27. Fields on the Campaign tab for project templates (continued)

Field	Description
	The default value is **partition1**, which is correct if Unica Campaign is installed to a single partition. If Unica Campaign is installed on multiple partitions, you can specify the partition to use for creating campaigns. You can specify any Unica Plan partition. Make sure that you specify a partition to which you have access and for which integration is enabled. For more information about setting up Unica Campaign partitions, the Unica Campaign Installation Guide.
Show TCS® tab in request	Select the check box to display the TCS® when the template is used to request a project. If the check box is clear, the TCS® displays only in campaign projects and not in requests.
Approval Required	Select the check box to require approval for all the target cells that are created in the template. If cleared, the TCS® grid does not show the **approval** column or **approve all** and **deny all**. **Note:** As part of the upgrade to version 8.2, **Approval Required** is cleared for all upgraded campaign templates. For more information, see TCS approval (on page 159).
Copy project attributes to campaign attributes	Select the check box to automatically copy project information to campaign when users click **Finish** to create campaign projects. If this check box is not selected, users can click the **Copy project attributes to campaign attributes** link to copy the data when they create a campaign project instance. By default, the check box is not selected.

TCS® approval

In integrated Unica Plan and Unica Campaign systems, campaign project templates can require (target cell spreadsheet) TCS® approval before flowcharts run in production mode. If **Approval Required** is selected on the template, then all rows of the TCS® must to be approved before the flowchart can run in production mode. If you run the flowchart in production mode and one or more rows on the TCS® associated with this flowchart are not approved, Unica Campaign generates an error.

If the project was created from a template on which the **Approval Required** check box is not selected, then approval is not required for top-down cells in the TCS®. In this case, the TCS® grid does not show the **approval** column or **approve all** and **deny all**. Leaving the approval required check box cleared saves time if campaigns do not require TCS® approval.

Note: By default, **Approval Required** is cleared. However, after you upgrade to Unica Plan 8.5, all upgraded campaign templates have **Approval Required** checked.

Import and export

If **Approval Required** is selected, the **is approved** column is included when you export the project.

If **Approval Required** is cleared, the **is approved** column does not export, and only CSV files that match are imported.

Data Mapping Definitions

The **Data Mapping Definitions** page maps data between campaign projects in Unica Plan and campaigns in Unica Campaign. Use the **Data Mapping** link on the **Template Configuration** page to configure data mapping.

The **Data Mapping Definitions** page contains the following columns:

Column	Description
Name	The name of the data mapping file.

Column	Description
Type	**Campaign Metrics Import**: Maps Unica Plan project metrics to Unica Campaign contact and response counts. If you have map files from previous versions, you might see other values in the Type column.
Used By	A list of templates that use the data map.

Note: You cannot create a map file within Unica Plan. Use a text or XML editor to create and edit the necessary map files.

Mapping Unica Campaign contact and response counts to Unica Plan metrics

If you want users to be able to import contact and response counts into Unica Plan, you must map contact count and the response types to Unica Plan metrics.

Note: Unica Campaign passes data to Unica Plan for one audience level only, the audience level that is mapped to the UA_ContactHistory, UA_ResponseHistory, and UA_DtlContactHist system tables. The audience level can be any audience level, with any number of audience key fields of any data type or name. For details on audience levels, see the Unica Campaign documentation.

The response types are stored in the UA_UsrResponseType system table in the Unica Campaign database. To map a metric to a response type, you must know the name of the response type.

The mappings are stored in an XML file.

1. In Unica Campaign, modify the list of response types in the UA_UsrResponseType table if necessary to include the response types you want to track.
2. Edit the Unica Plan metrics file being used by your system to include metrics that correspond to contact count and the response types.

3. Create a map file that associates the Unica Plan metrics with contact count and the response types.
4. Add the map file to Unica Plan.
5. Create a campaign template and select the map file from the **Metric Data Mapping** drop-down list.

The contact and response data is mapped to metrics for all projects that are created from that template.

Metrics data mapping file

You define the metrics to map your data.

The metrics data mapping file must use the container elements `<metric-data-mapping>` and `</metric-data-mapping>`.

The next line in the mapping file must be as follows:

```
<datasource type="webservice">
   <service-url>CampaignServices</service-url>
</datasource>
```

The actual mappings must be contained by the elements `<metric-data-map>` and `</metric-data-map>`.

metric

Use the `<metric>` element to define the metric in a mapping. The `<metric>` element has no value but must contain the child element `<data-map-column>`. The `<metric>` element has the following attributes:

Attribute	Description
id	The internal name of the metric
dimension-id	The number of the column into which the value from Unica Campaign should be placed. Columns are numbered from left to right. The first column is column 0.

data-map-column

Use the `<data-map-column>` element to define the data source (either contact count or a response type) in a mapping. The `<data-map-column>` element must be contained by the `<metric>` element that defines the metric to which contact count or this response type is mapped. The `<data-map-column>` element has no value but has the following attributes:

Attribute	Description
id	The data source to be mapped to the metric. For contact count, use `contactcount`. For a response type, use `responsecount_<`*`ResponseTypeName`*`>`.
type	This value should always be `number`.

Adding data mapping files

Use a text or XML editor to create or edit a data mapping file. After you have a data mapping file, you add it to Unica Plan.

1. Select **Settings > Plan Settings**.
2. Click **Template Configuration > Data Mapping**.
3. Click **Add Data Mapping**.

   The Upload Data Mapping dialog box opens.
4. Enter a name for the data mapping file.
5. Browse to the XML file that defines the data mapping.
6. Click **Continue**.

Editing data mapping files

If you want to update a data mapping file, you must first edit the XML file, and then reload it back into Unica Plan.

1. Open the data mapping XML file in a text editor and make your changes.
2. Select **Settings > Unica Plan Settings**.

3. Click **Template Configuration > Data Mapping**.
4. Click the file name that you are updating.

   The **Update Data Mapping** dialog opens.
5. Select **File**, and browse to the XML file.
6. Click **Continue**.

   You are prompted to overwrite the existing file.
7. Click **Save** to overwrite the existing file with the newer version.

Deleting data mapping files

You cannot delete a mapping file if a template uses it.

1. Select **Settings > Plan Settings**.
2. Click **Template Configuration > Data Mapping**.
3. Click the **Delete** link for the data mapping file you want to delete.

Chapter 10. Forms

A form is a collection of attribute fields that gather information about an object.

When you create a template, you select the forms that you want to include. Each form that you add becomes a separate tab or a section of the Summary tab in the object instances users create by using that template.

You create and manage forms. You select **Settings > Unica Plan Settings > Template Configuration > Forms**.

Form Definitions page

The **Form Definitions** page lists each of the custom forms that are defined for your system.

For each form, the following columns display.

Table 28. Columns on the Form Definitions page

Column	Description
Name	The display name and the description of the form for use in Unica Plan.
Table	The name of the database table that stores the values that users enter for form attributes.
Used By	A list of the templates that use this form.
Actions	This column displays an icon that represents one of the following actions for the form. A tip with the option that is available displays when you move your cursor over the icon. • **Publish**: makes the form available for use in object templates. After a form is published, Disable displays until changes are made. • **Disable**: prevents this form from appearing in the list of available forms on the template Tabs tab. Disabling a form does

Table 28. Columns on the Form Definitions page (continued)

Column	Description
	not change any existing templates that include it. After a form is disabled, the Enable action displays. • **Enable**: makes the form available in the list of available forms on the template Tabs tab. See Template Tabs tab for customizing the user interface (on page 110).
Delete/Revert	This column displays an icon that represents one of the following actions for the form: • **Revert**: undoes any changes that you made to the form since it was last published. If there are no unpublished changes, this option changes to **Delete**. • **Delete**: deletes the form. This option is only available for forms that are not used in any templates.
Export	Click to export the latest published version of the form.
Copy	Click to create a copy of the form.
Manage	Click to enable or disable the lookup values used by attributes in the form.

The list page also contains the following links.

Table 29. Links on the Form Definitions page

Link	Description
Create New Form	Click to open the **Form Editor** so you can create a form.
Import Form	Click to select a form to import into your system.

Creating forms

Before you create a form in Unica Plan, you should design it on paper or in a spreadsheet.

Be sure to consider what attributes to include on the new form, what to label them, how to group and sequence them, and where to store the data they collect. You are extending not just the Unica Plan user interface (that is, the fields that are presented), but also the out-of-box set of system tables.

Unica Plan creates the database table and columns to store the data that users enter in the forms, but you must specify the table and column names. The tables are created when you publish the form. If you want users to select values from lookup tables when entering information in the form, you must create the lookup tables manually.

Also consider which attributes you want to use on multiple forms. You should create these as shared attributes before you create the forms.

Creating a form

You can create a form from the **Form Definitions** page.

1. Click **Settings > Plan Settings**.
2. Under Other Options, click **Template Configuration**.
3. Under Template Components, click **Forms**.
4. On the **Form Definitions** page, click **Create New Form**.

   The Form Editor interface displays. For more information about working with this interface, see Form Editor interface *(on page 167)*.
5. Complete the Form Properties tab and click **Save Changes**.

   The Add an Element tab displays.
6. To use previously defined shared attributes on this form, click **Import Shared Attributes** then select them in the Custom Attributes list.
7. To add a local attribute to this form only, click **Create a New Custom Attribute**.
8. Drag the elements and attributes that you want in the form from the Add an Element tab and drop them onto the form design area.
9. Click **Save and Exit** to save the form and return to the **Form Definitions** page.

Importing shared attributes

You can import only attributes that are enabled.

For more information, see Creating and enabling a shared attribute (on page 201).

1. Open the form in which you want to use the shared attributes.
2. On the Add an Element tab, click **Import Shared Attributes**.

   The Shared Attributes dialog box displays.
3. In the list on the left, select the attributes that you want to import then click **>>** to move them into the **Selected Attributes** list.
4. Click **Import and Close**.

Form Editor interface

The Form Editor displays when you create or edit a form and consists of a form design area on the left and a set of tabs on the right.

The form design area displays the current contents of the form. You supply information for the form and its attributes with the controls on the tabs on the right: you can add elements to the form by clicking and dragging.

For more information, see Creating a form (on page 166).

There are two tabs on the right. The Form Properties tab contains the following fields.

Table 30. Form Editor interface: the Form Properties tab

Field	Description
Form Name	The name of the form for use in Unica Plan.
Database Table	The name of the database table that stores the answers that users enter in the form fields. The maximum number of characters that are allowed for the Database Table field depend on the database that is used for Plan.

Table 30. Form Editor interface: the Form Properties tab (continued)

Field	Description
	• For Oracle database - maximum 30 characters. • For SQL database - maximum 128 characters. • For DB2 database - maximum 128 characters. **Note:** You cannot use the same database table for both a form and a grid within that form. **Note:** Ensure that you do not use the "uap" and "umcm" prefixes while defining tables for forms and grids.
Form Description	A description of the form. This text displays below the form name on the **Form Descriptions** page.

The Add an Element tab contains two list boxes:

- The General Elements list box contains form elements, such as a group header to identify a related set of attributes.
- The Custom Attributes list box contains a list of the different attributes available for use on the form.

The tab also contains the following links.

Table 31. Form Editor interface: links on the Add an Element tab

Link	Description
Create a New Custom Attribute	Click to open the **Create a New Custom Attribute** dialog, where you can create a local attribute.
Create New Grid	Click to open the **Create a Grid** dialog, where you can create an editable or read-only grid.

Table 31. Form Editor interface: links on the Add an Element tab (continued)

Link	Description
Delete Selected Attribute	Click to delete the attribute that is selected in the **Custom Attributes** list box.
Import Shared Attributes	Click to open a dialog where you can select previously defined and enabled shared attributes to import for use in this form.
Create/Edit Form Attribute Rules	Click to open the **Rule Builder...** dialog, where you can define rules for hiding and making visible attributes and attribute groups. For information about configuring attribute rules, see Marking attributes and attribute groups visible based on other attributes *(on page 234)*.

After you create local attributes or import shared attributes, you can add them to the form. To add an element or an attribute to the form, you click it and then drag it into the form design area, directly below a group header.

After you add an element or attribute to the form, you click it to view or edit its settings. When you click a form element or attribute, a dialog opens with the current values, covering the tabs on the right. The dialog contains an **Edit** link so that you can specify how the selected group header or attribute is implemented on this form. For more information, see Edit Attribute Group dialog (on page 170) or Attributes reference (on page 204).

Attribute groups

Every attribute or table on a form must be in a group. Groups enable you to logically organize fields for end users.

You can also use groups to create a form with both 1-column and 2-column areas.

You can display a header for a group, but headers are not required.

Creating an attribute group

You can place a group element directly on a form or on a grid component on a form to identify a related set of attributes.

1. Open the form in which you want the attribute group.
2. Click the **Add an Element** tab.
3. Click **Attribute Group Header** in the **General Elements** list and drag it onto the form design area.

   A red cursor indicates the placement of the group header on the form: before (above) or after (below) an existing form element.
4. Drop the group header onto the form.
   "New Group <*n*>" displays.
5. Click the new group header to review the current settings for the group.
6. Click **Edit Attribute Group** to open a dialog where you can change the display name and specify other options.
   See Edit Attribute Group dialog *(on page 170)*.
7. When you are finished editing the group, click **Save and Exit** to close the window and return to the form.

Edit Attribute Group dialog

Attribute groups organize attributes on your custom forms.

Table 32. Fields for editing attribute groups

Field	Description
Group Internal Name	The unique name of the group, used internally. Do not use spaces or special characters.
Group Display Name	The group header, which displays on the form. Spaces and UTF-8 characters are allowed.
Description	A description of the group.
Show Group Heading	Select to display the group display name in the form. Clear to hide the group display name in the form.
Group Layout	How attributes display in the group. Select **One Column** or **Two Columns**.

Creating grids

Grids are spreadsheet-like tools for collecting data. A grid component can display in the user interface as either a table or a list.

Grids are either editable or read-only. An editable grid is for users to enter information, while a read-only grid displays information entered previously.

When you add a grid to a form, you set it up in two parts:

- The grid component defines the type of grid you want to add, and the database table in which to store entered data, or from which to read data to display.
- The grid attributes define each of the columns in the grid.

You determine whether a grid is editable or read-only when you create the grid component. If you want the same information to be editable in one form and read-only in another, you need to create two grid components and then set them up to include the same grid attributes.

You can group attributes within a grid by using an Attribute Group Header on the grid. A grid can contain a mix of grouped and ungrouped attributes.

Grid types

There are four grid types available. A description follows of the user interface display that results when an attribute of each type is implemented.

The following grid types are available.

Table 33. Grid types

Grid type	Description
Line Truncate View	Creates a read-only grid in which text that is too long to fit in a cell is truncated.
Line Wrap View	Creates a read-only grid in which text that is too long to fit in a cell continues on to another line within that cell.

Table 33. Grid types (continued)

Grid type	Description
Two Line Staggered View	Creates a read-only grid in which text that is too long to fit in a cell continues on to a second, indented line.
Editable Grid View	Creates an editable grid. Users enter data into the cells of the grid.

Creating an editable grid

You can create an editable grid to a new or exiting form.

1. Create or open the form that you want to include a data-entry grid.
2. One the **Add an Element** tab, click **Create New Grid**.
   The Create New Grid dialog opens to collect information for the grid component.
3. From the **Grid Type** drop-down list, select **Editable Grid View**.
4. Enter the information for the database table and other fields. See Create New Grid dialog *(on page 173)*.
5. Click **Save and Exit**.

   On the **Add an Element** tab, click to expand the list of **Form Attributes**. Your grid component displays.
6. To add a grid to a form, you must place it within a group. If the form does not already contain a group for the grid, click and drag **Attribute Group Header** onto the form design area then name it appropriately.
7. From the list of **Form Elements**, click and drag your grid component and drop it onto the group.
8. To add attributes, or columns, to the grid, expand the **Grid Attributes** list. Click and drag attributes and drop them onto the name of the grid component.
   For information about adding attributes, see Form Editor interface *(on page 167)*.

   You can change the order of the attributes by clicking an attribute name. The move icon displays above the name: click and drag that icon to move the column to the desired location.

If you want to group some of the grid attributes together, drag an **Attribute Group Header** onto the grid, and then drag the grid attributes onto the group header.

9. Click **Save and Exit** to save the form and return to the Form Description list page.

Create New Grid dialog

The Create New Grid dialog has fields that you need to understand.

Table 34. Fields in the Create New Grid dialog

Field	Description
Grid Type	The type of grid you want to create. • Line Truncate View • Line Wrap View • Two Line Staggered View • Editable Grid View See Grid types (on page 171).
Is TCS	Applies to editable grids only. Select this check box to create a Target Cell Spreadsheet for use in campaign projects. Clear this check box for all other data-entry grids.
Grid Internal Name	The name to use when creating files for the grid.
Grid Display Name	The display name for this grid.
Database Table	The database table that contains the data users enter in the grid (for editable grids) or the database table that contains the data that displays in the grid (for read-only grids). **Note:** The grid database table must not be the same as the form database table.

Table 34. Fields in the Create New Grid dialog (continued)

Field	Description
Table Key Column	For an editable grid, the name of the column to contain the parent ID (the ID of the project or marketing object that holds the grid). If multiple editable grids (including Target Cell Spreadsheets) use the same database table, they must use the same table key column. • For a read-only version of an existing editable grid, use uap_grid_row_id. • For a read-only grid not related to an existing editable grid, the name of a column that uniquely identifies rows in the table that contains the data you want to display. **Note:** You should not change the table key column after you create the grid.
Key Column Type	The data type of the table key column.
Number of rows in one page	The number of rows to display in one page of the form. This value cannot be greater than 100.
Show Export Link	Select this box to enable users to export the grid data or data selections.

For grids that display read-only data, the Create New Grid dialog presents the following additional fields.

Table 35. Fields in the Create New Grid dialog for read-only grids

Field	Description
Data Post URL	The URL of the server to which the data the user selects is sent. (This option is not available for editable grids.)

Table 35. Fields in the Create New Grid dialog for read-only grids (continued)

Field	Description
Filter By Parent Id	Select this check box to filter the read-only grid to show only entries from the current project or marketing object. (This option is not available for editable grids.)
Parent ID Column Name	The value of the Table Key Column for the grid component for the editable grid whose data you want to display as a read-only grid. (This option displays only if the Filter by Parent ID check box is selected.)
Show View Link	Select this check box to enable users to set their viewing options for the grid. (This option is not available for editable grids.)
Show Group By Link	Select this check box to enable users to specify a column by which to group the grid rows.

Display an existing editable grid as a read-only grid

In order to collect data in an editable grid and display the same data in a read only grid, you create two different forms.

Note: The form that contains the editable grid must be published before you create the read-only grid. Otherwise, you cannot save the form that contains the read-only grid. If you want the editable grid and the read-only grid to be on the same form, you must create the editable grid, publish the form, and then create the read-only grid.

The grid component in the read-only grid must have the following properties. See Create New Grid dialog (on page 173).

- The **Grid Type** must be Line Truncate View, Line Wrap View, or Two Line Staggered View.
- The **Database Table** must be the same as the database table for the editable grid.
- The **Table Key Column** must be uap_grid_row_id.

Note: Unica Plan automatically creates this column for all editable grids.

- If you want the read-only grid to display only values that users enter in the grid for this object instance (for example, this project), select the **Filter by Parent ID** check box and enter the value for the **Table Key Column** of the editable grid in the **Parent Id Column Name** field for the read-only grid.

  Otherwise, the read-only grid displays all of the values that are entered in the editable grid in all objects.

The attributes that the read-only grid contains must be exact matches of attributes that the editable grid contains. You can accomplish this requirement in one of three ways.

- Copy the form that contains the editable grid. You must delete the grid component for the editable grid, create a component for the read-only grid, and drag the grid attributes onto the new component. However, you do not have to re-create the grid attributes.
- Use shared attributes when you create the editable grid so that the attributes can be imported into the **Form Editor** when you create the read-only grid.
- Re-create the attributes in the **Form Editor** when you create the read-only grid. The attribute properties must match the original attributes exactly.

Note: The one exception is that read-only grids cannot contain Single Select Object Reference or Multi Select Object Reference attributes. If the editable grid contains attributes of these types, you must replace them with Single List Object Reference attributes. See Attribute types list *(on page 208)*.

Display a grid as a list

You can reference a grid, and display it on another form as a list.

To configure the list correctly, it can help to understand how Unica Plan stores grids.

- A list view is read-only and displays all of the rows in a specified database table.
- A grid view is either editable or read-only. The rows in the grid "belong" to the parent of the grid; that is, to the project or marketing object instance that includes the grid on one of its custom forms.

For example, if users create two projects from the same template, the rows added by the corresponding grids in both projects are added to the same database table, but the grid for each project has access to only its own data.

To achieve this, a database table that holds grid data must have two columns: one to uniquely identify each row, and one to identify the parent ID: the ID of the project or marketing object that contains the grid.

The **Table Key Column** that you set for a grid holds the parent ID for the row. All rows in a single grid have the same value for this column. As a result, it does not uniquely identify a row of data.

The **Forms Editor** automatically generates a column, uap_grid_row_id, for each grid. A list view just requires a column that uniquely identifies rows. Therefore, to use the same table as the grid for a list view, you can specify uap_grid_row_id as the key column for the list. Do not use the same key column that you specified when you specified the grid data table.

Example

To illustrate, look at this example:

- The Tradeshow template contains a custom tab, Staff. The form for the Staff tab contains a grid.
- Two tradeshow projects exist, TRS001 and TRS002.
- TRS001 and TRS002 have object IDs of 121 and 122 respectively.
- Data exists in the Staff grid for both TRS001 and TRS002.

The database table that holds the data for these two grids might look like the following illustration:

Data in Table 'dyn_staffdetails' in 'SJMPlan73' on 'FORTUNA'

object_id	uap_grid_row_id	manager	emp_id	emp_name	salary
121	118	Y	1001	Mary Manager	45000
121	119	N	1002	Art Artiste	25000
121	120	N	1003	Larry Lawyer	200000
121	121	N	1004	Carl Contributor	25000
121	122	Y	1005	Charlie CEO	1000000
122	123	N	5000	Huey Lewis	25000
122	124	Y	5001	Isaac Bashevis Sinç	75000
122	125	N	5002	Carl Sagan	100000
122	126	Y	5003	Emiliani Torrini	300000

The first several rows belong to the grid on TRS001. The final few rows belong to the grid on TRS002.

Each project displays only its own portion of the data in this table. However, if there is a list that uses this table for reference, it displays all the rows in the table, as shown here:

TVCListStaff:

View | Export Data

Employee ID	Name	Base Pay	Manager?
1001	Mary Manager	$45,000.00	Yes
1002	Art Artiste	$25,000.00	No
1003	Larry Lawyer	$200,000.00	No
1004	Carl Contributor	$25,000.00	No
1005	Charlie CEO	$1,000,000.00	Yes
5000	Huey Lewis	$25,000.00	No
5001	Isaac Bashevis Singer	$75,000.00	Yes
5002	Carl Sagan	$100,000.00	No
5003	Emiliani Torrini	$300,000.00	Yes

Since the column name uap_grid_row_id is reserved, you must not use this as a column name when you specify the columns for the grid.

You can filter a list so that it displays only the grid entries from its own object (project or marketing object). Continuing the previous example, create a project template that includes two custom tabs:

- Staff Form: contains a grid that you use to enter and edit members of the staff.
- Staff List: displays the entries from the Staff Form grid as a list.

To ensure that only the entries from the current project display on the Staff List tab, filter on the parent ID of the list.

When you create the grid component for the list, set the following values:

- **Filter By Parent ID**: selected.
- **Parent Id Column Name**: object_id (as shown in the database table earlier in this example). This value must match the value in the **Table Key Column** for the grid.

Example: creating lists of marketing objects

This example describes how to display a list of marketing object references.

Scenario

You have a project with four subordinate marketing object types:

- Two different brochures
- One mailer
- One resource bundle

After you create the project, you periodically check to see which participating marketing objects have already been created.

This example describes the steps necessary to create this scenario in Unica Plan.

Assumptions

The following items exist in Unica Plan:

- A project template named "Event planning".
- Marketing object templates for Brochures, Mailers, and Resource bundles.

Tasks

To implement this scenario, you do the following tasks.

1. Using the **Forms Editor**, set up the following forms:
    - Create a form to hold a custom Text - Single-Line attribute, **Originating Project**.

      After users create a participating marketing object, they enter the value of the project code for the originating project in this field.
    - Create a form for a custom tab, Participating Marketing Objects.

      For the custom tab, you add a Single Select Object Reference attribute to a form. To configure this attribute, you must first create a custom view.

    See "Creating the custom tab and attribute" below for details about creating these forms.
2. Create a custom view. See "Creating the custom view" below.
3. Add the forms to the appropriate templates.
    - Add the form with the Originating Project attribute as a section on the Summary tab of the Brochures, Mailers, and Resource bundles templates.
    - Add the form with the list as a custom tab on the Event planning project template.
4. Create the object instances from the corresponding templates:
    - A project, EventStuff001.
    - A brochure, Brochure001, with the originating project set to the project code for EventStuff001.
    - A mailer, Mailer001, with the originating project set to the project code for EventStuff001.

    When users open the Participating Marketing Objects tab for EventStuff001, they see the details for the associated marketing objects.

Creating the custom tab and attribute

We need a tab that can hold the list. We add this tab to a project template in Unica Plan. We also need a custom form to hold the project code. The custom view depends on both these forms using the same database table, so we create them both in this section.

1. In the **Forms Editor**, set up two forms, both using the same database table.
2. Create the database table as follows:

Field	Value
Table Name	dyn_mo_table
Display Name	dyn_mo_table
Key Column Name	po_id
Attribute Name	po_id

3. Create the two forms:

Field	Form 1	Form 2
Internal Name	OriginatingProj	linkedMOs
Display Name	Originating Project	Associated MOs
Description	Holds a single attribute that points to the originating project.	Form to display the linked Marketing Objects in a list.
Attribute Name Database Table	dyn_mo_table	dyn_mo_table

4. For the list grid component, specify:

Field	Value
Reference Database Table	proj_mos_by_proj_-code
Reference Table Key Col-umn	mo_id
Filter By Parent ID	checked
Parent Id Column Name	ProjID

5. Create a grid attribute of type Single Select Object Reference. In the Object Reference Properties section, specify:

Field	Value
Object Reference ID Column	mo_id
Object Reference Type Column	comp_type_name

6. Create a form attribute to hold the project code as a text attribute. For this attribute, specify:

Field	Value
Attribute type	Text - Single-Line
Internal Name	PID
Form	Originating Project
Display Name	Project Code
Database Column	PID

7. Save the forms.
8. Run the SQL script to create the dyn_mo_table and its columns.

Creating the custom view

Typically, you create a custom view before you can add an object reference to a list view. In this example, we reference a marketing object on a project, where the marketing object contains a text field that holds a project code.

This example uses three tables to create the view: uap_projects, uap_mktgobject, and the custom table dyn_mo_table. The view is named proj_mos_by_proj_code.

Prerequisite custom table

Before you create the view, ensure that you created the custom table, dyn_mo_table, and that it contains the following columns:

- po_id: key column, specified in the DB Tables tab for the form
- PID: text column created in the Forms Attribute tab to hold the project code as text.

Custom view details

The view contains the following columns:

- proj_code and project_id from uap_projects
- name, comp_type_name, and mktg_object_id from uap_mktgobject

The actual SQL code to create the view follows:

```
create view proj_mos_by_proj_code (
   asscProj, MOName, ProjID, mo_id, comp_type_name) As
   select PROJ.name as asscProj, MO.name as MOName,
   PROJ.project_id as ProjID, MO.mktg_object_id as mo_id,
   MO.comp_type_name as comp_type_name
   from uap_projects PROJ, dyn_mo_table MOT, uap_mktgobject MO
   where PROJ.proj_code = MOT.PID and MOT.po_id = MO.mktg_object_id
```

The following table shows the column names and some sample rows from this database view:

asscProj	MOName	ProjID	mo_id	comp_type_name
BRAIN-001	RB-005	101	147	creatives
BRAIN-001	RB-006	101	148	creatives
Event Horizon	CampaignMAIL01	149	145	creatives
Event Horizon	CampaignBRO01	149	142	creatives
Event Horizon	CampaignRB01	149	143	creatives
Event Horizon	CampaignRB02	149	144	creatives

Target cell spreadsheets

A Target Cell Spreadsheet (TCS®) is an editable grid component that has a set of predefined attributes. You use Target Cell Spreadsheets when Unica Plan-Unica Campaign integration is enabled.

The TCS® specifies the data that users must enter in a campaign project to define the target and control cells for the campaign. Each column on the TCS® corresponds to an attribute. Values for the predefined, or default, attributes are automatically passed to Unica Campaign. In addition to these default attributes, you can add an unlimited number of custom attributes to a TCS®.

Cell attributes and grid attributes

A TCS® can contain attributes that get passed to Unica Campaign (cell attributes) and attributes that display only in Unica Plan (grid attributes).

Use cell attributes for information you want to pass to Unica Campaign. For example, attribute values you want to include in an output list, contact history, or report must be created as cell attributes.

Use grid attributes for descriptions, calculations, and data that is not needed in Unica Campaign.

Target Cell Spreadsheets and forms

You add a TCS® to a form by creating a new editable grid and identifying it as a TCS®. (The form can contain other attributes in addition to the TCS®.) When you place a TCS® grid component on a form, it contains the default cell attributes. You cannot delete the default attributes.

Transfer of cell attribute data

When a user links a flowchart cell to the TCS® row, the data values supplied for the default attributes are passed to Unica Campaign automatically. Custom cell attributes are available automatically in Unica Campaign as Unica Campaign Generated fields in contact processes. For details on Generated fields, see the Unica Campaign User's Guide.

Target Cell Spreadsheets and templates

Each campaign project template can contain only one TCS®.

Creating a Target Cell Spreadsheet

You can create a Target Cell Spreadsheet by following these steps.

Before you create a TCS, you must create any custom cell attributes that you want to include. Cell attributes map to Unica Campaign and can be created only as shared attributes.

1. Select **Settings > Plan Settings**.
2. Under **Other Options**, click **Template Configuration**.
3. Under **Template Components**, click **Forms**.
4. On the **Form Definitions** page, click **Create New Form**.
   The **Form Editor** interface displays.

   Note: Ensure that you do not use the "uap" and "umcm" prefixes while defining tables for forms and grids.

5. Complete the **Form Properties** tab and click **Save Changes**.
   The **Add an Element** tab displays.
6. Click **Create New Grid**.
   The **Create New Grid** dialog opens.
7. From the **Grid Type** drop-down list, select **Editable Grid View**.
8. Select the **Is TCS** check box.
9. Complete the remaining options and click **Save and Exit**.

   On the **Add an Element** tab, click to expand the list of **Form Attributes**. Your TCS grid component displays.
10. To add a TCS grid to a form, you must place it within a group. If the form does not already contain a group for the grid, click and drag **Attribute Group Header** onto the form design area then name it appropriately.
11. From the list of **Form Elements**, click and drag your TCS® grid component and drop it onto the group.

    The default cell attributes display on the grid. See Default cell attributes for Target Cell Spreadsheets (on page 186).
12. Add any attributes that you want to include in the TCS. You can do either of the following steps.

- Import custom cell attributes and add them to the TCS to create more columns. these columns are passed to Unica Campaign.
- Create or import grid attributes and add them to the TCS to create more columns. These columns display only in Unica Plan.

Note: Ensure that you do not use the "uap" and "umcm" prefixes while defining tables for forms and grids.

13. Click **Save and Exit** to save the TCS and return to the **Form Definitions** list page.

Default cell attributes for Target Cell Spreadsheets

All TCS® grids include the predefined cell attributes by default. These attributes are available specifically for use in a TCS® grid, and not displayed on the **Shared Attributes** page.

Descriptions of how these attributes display to users in a campaign project follow.

Table 36. Default cell attributes

The following three-columned table describes cell names in one column, publish requirements in the second column, and their description in the third column.

Name	Required to publish the TCS®?	Description
Cell Name	Yes	Text field.
Cell Code	No	Text field.
Description	No	Text field.
Is Control Cell	Yes	Drop-down list with Yes and No options.
Control Cell	No	Drop-down list of control cells.

Table 36. Default cell attributes

The following three-columned table describes cell names in one column, publish requirements in the second column, and their description in the third column.

(continued)

Name	Required to publish the TCS®?	Description
Assigned Offers	No	A selection control that you can use to select one or more offers or offer lists.
Is Approved	No	Drop-down list with Yes and No options. This column is included only if **Approval required** is checked in the campaign project template.
Flowchart	No	Read-only field that displays the name of the flowchart in which the cell is used.
Last Run	No	Read-only field that displays the date and time that the flowchart that contains this cell was last run.
Actual Count	No	Read-only field that shows the last run count for this cell (count of unique audience IDs in the cell).
Run Type	No	Read-only field that shows the run type for the last run for the flowchart that contains this cell (production or test; flowchart, branch, or process box).

After you add a TCS® grid, you can edit the Attribute Display Name, Description, and some of the other properties for the individual cell attributes. You cannot edit the predefined values for these properties:

- Attribute Category
- Attribute Type
- Attribute Internal Name
- Attribute Display Name

- Attribute Database Column Name
- Can group by
- Form Element Type
- Special behavior

Publishing forms

You can only add published forms to a template.

Each time you edit a form, you must publish it again.

Publishing a form

You can publish a form from **Plan Settings > Template Configuration**.

1. Click **Settings > Plan Settings**.
2. Click **Template Configuration**.
3. Click **Forms**.
4. Click the **Publish** link for the form you want to publish.

   If the Publish link is not available, the form has not been changed since it was last published.

Exporting forms

You can export forms only after they are published. Any changes that are made to the form since it was last published are not included in the exported form.

To export a form, you must know the database application that is used by the Unica Plan system that will import the form. You specify the database application when you export the form.

When you export a form, Unica Plan creates a compressed archive file that includes the following.

- A map file of the form in XML format.
- A create script for the database application you specify.

- A drop script for the database application you specify.
- A properties file for each locale for which a translation exists.
- A create script, drop script, and insert script for every lookup table that is used by a Single-Select - Database or Multiple-Select - Database attribute on the form.

Exporting a form

To export a form, you must know the database application that is used by the system that imports the form.

1. Click **Settings > Plan Settings**.
2. Click **Template Configuration**.
3. Click **Forms**.
4. Click the **Export** link for the form you want to export.
5. Select the database application that is used by the Unica Plan installation that will import this form.
6. Click **Export**.

Importing forms

You can import forms only from a system that is running the same version of Unica Plan.

Only Unica Plan administrators can import forms.

You can import a form in one of two ways.

- By importing a complete, previously exported compressed form archive file.
- By importing a single previously exported form (XML) file.

Importing a compressed form archive file imports the form, any localized versions, and scripts to update lookup tables that are referenced by form attributes.

Importing a form (XML) file imports only the form. The form uses the language of the locale in which it was created. If the form attributes use lookup tables, you must create or edit the lookup tables manually.

You can import a new version of a form that exists in the system. If the existing form is unpublished, the new version replaces the old one. If the existing form is published, the **Form Definitions** page lists the newly imported version below the old version and the **Publish** action is available. To replace the published version with the new version, you must republish the form.

Importing a form

You can import a form in either an XML file or compressed form archive file.

1. Click **Settings > Plan Settings**.
2. In the Other Options section, click **Template Configuration**.
3. Click **Forms**.
4. Click **Import Form**.
   The **Import Form** dialog opens.
5. Select one of these options:
   a. To import a single form XML file, select **File**.
   b. To import a compressed form archive file, select **Form Archive**.
6. If you are importing a single XML form file, enter a name for the form.

   Use only alphanumeric characters, the space character, and the underscore character in the form name.

   If you are importing a form archive, Unica Plan takes the form name from the archive.
7. Browse to the file you want to import.
8. If you are importing a compressed form archive file, select **Drop Lookup Tables**, **Create/Update Lookup Tables**, or both.
9. Click **Continue**.

Troubleshooting when importing forms

This section describes how to correct some common errors you may receive when you attempt to import a form into the **Forms Editor**.

Error	Solution
Duplicate form name	The form name is the same as one that already exists in the system. Rename the form file, or open a new form and reimport the form file.
Cannot use names	Some <element> tags have the same name. Rename any duplicate names in <element> tags or open a new form and reimport the form file.

Moving forms from one computer to another

You can move a form from one computer to another by exporting the form from one computer and importing it into another installation of Unica Plan.

For example, you can move a form from a development installation to a test installation to a production installation.

Managing lookup values for a form

Single-Select - Database and Multiple-Select - Database attributes present lists of values to users, who then select one or more values from the list.

You manage attribute lookup values as follows:

- Work directly with your database administrators to add or remove values in the lookup table that is associated with the attribute.
- Disable lookup values on the **Form Definitions** page, as described here. Disabling values on the **Form Definitions** page prevents users from selecting a value without removing it from the database. If the same lookup table is referenced on more than one form, you can disable a value for one form and enable it for another on the **Form Definitions** page.

Notes® on disabled lookup values

Note the following system behavior for disabled lookup values:

- The status (enabled or disabled) for lookup values is kept in the `uap_lkup_manager` system table.
- You can disable a value that is selected on existing objects. When users revisit such an object, they see **disabled** next to the value.
- Disabled values are included in the results from an advanced search if they meet the search criteria. Disabled values are indicated by the text **disabled** next to the value.
- If you edit a Single-Select - Database and Multiple-Select - Database attribute, the status for all values of that attribute is reset to enabled.
- If a value is set as the default for a form, and is later disabled, the behavior is as follows:
    - The disabled value is still used for existing objects.
    - If a user revisits an existing object, and changes any answers on that form, the user must choose a different value for the field that contains the disabled value.

Disabling lookup values without changing the database table

Without changing the database table, you can disable lookup values.

1. Click **Settings > Plan Settings**.
2. In the Other Options section, click **Template Configuration**.
3. Click **Forms**.

   The **Forms Definitions** list page displays.
4. Click **Manage** for the form that contains the lookup values you want to edit.

   The **Manage Lookup Values** dialog opens.
5. Clear the **Enabled** check box for any lookup value that you want to disable.

   Clicking the check box toggles the setting: values that contain a check are enabled, and values that are not checked are disabled.
6. After you make your changes, click **Save changes**.

Copying forms

You can copy any published form that is enabled. Unica Plan copies the last published version of the form.

The name of the copy is "Copy of <form_name>". If "Copy of <form_name>" exceeds 50 characters, Unica Plan displays an error message and you must specify a new name for the copy.

The database table name of the copy is `copy_of_<original_table_name>`. If this table name exists in the database, you must change it before you can save the form. Any localized properties files that exist for the original form are copied for the new form.

To copy a form, click the **Copy** icon in the row for the form on the **Form Definitions** page.

Enabling data posting of list selections

When you create a read-only list, you can enable users to send their data selections to a specified server by clicking a link in the form.

When users click the Post Data link, the selected rows display in a new popup window.

Data posting is done using name-value pairs, as per the HTML POST method. Name is the column name, and value is the value of the column in the selected row. If the user selects multiple rows, value pairs are comma-separated.

For example, assume that a list has two columns, ID and Name, and the data post URL is set to `http://serverRPT/testServlet`. Assume that the list has the following values:

Table 37. Example list values

ID	Name
1	name1
2	name2
3	name3

If a user selects the first and third rows, and then posts the data, the system generates the following HTML form into a new window:

```
<form name="lvcPost" method="POST"
     action="http://serverRPT/testServlet">
     <input type="hidden" name="ID" value="1,3">
```

```
     <input type="hidden" name="NAME" value="name1,name3">
</form>
```

If a posted column contains multiple, comma-separated values, these values are enclosed in double quotation marks ("") when posted. The quotation marks identify these values as belonging to a single column, as the posting method normally comma-separates the values for each column.

Adding a form to existing objects

When you add a new form to an existing object template, the new form does not display in objects previously created from the template. You can manually edit the database table for the form to make the new form display in all objects of a specified type (for example, all projects).

To have the new form appear in existing objects, you must insert all the object IDs for that object type into the database table for the new form using the following SQL statement:

```
INSERT INTO table_name (object_id) SELECT object_id
  FROM object_system_table
```

where

- *table_name* is the name of the table for the form
- *object_id* is the object identifier column for the object type
- *object_system_table* is the name of the system table for the object

This table specifies the ID column and system table name for each object type:

Object	ID column	System table
project	project_id	uap_projects
program	program_id	uap_programs
plan	plan_id	uap_plans

Object	ID column	System table
invoice	invoice_id	uap_invoices
marketing objects	mktg_object_id	uap_mktgobject

For example, if you added a form with a table named **dyn_x** to a project template, run the following SQL statement to add the form to all existing projects:

```
INSERT INTO dyn_x (project_id) SELECT project_id FROM uap_projects
```

Chapter 11. Attributes

An attribute defines a piece of information that you want to gather from a user. For example, the information can be text, integers, dates, or a choice from a predefined list. In Unica Plan, you define attributes to collect these different types of information, and then place them on forms. Then, you can add or more forms to templates as tabs. When users create items, they select a template. Each attribute that is included in the template corresponds to a field or other user interface control on the tabs that collect information.

Standard and custom attributes

Unica Plan delivers a set of standard attributes that can be used to collect information for all marketing objects. The standard attributes include name and description. To collect additional information, you create custom attributes, enable them, and add them to forms.

Before you begin to create custom attributes, Unica Plan attributes can be shared or local, and that they are categorized based on how they can be used on forms.

Shared and local attributes

Custom attributes are either shared or local, depending on whether they can be used repeatedly on different forms or only on a single form.

- Shared attributes can be imported into and used on any form. You create a shared attribute by clicking **Settings > Unica Plan Settings > Template Configuration > Shared Attributes**.
- Local attributes apply to a single form only. You create a local attribute directly on that form by clicking **Create a New Custom Attribute**.

Attribute categories

Custom attributes are grouped into categories that are based on how they can be used to collect information. The attribute categories follow.

- Form attributes can be placed on any form.
- Grid attributes can be used in a grid interface.

The following attribute categories are available when Unica Plan and Unica Campaign are integrated. These attributes collect information that is mapped to Unica Campaign.

- Campaign attributes can be used in campaign project templates.
- Cell attributes can be used in Target Cell Spreadsheets.
- Offer attributes can be used in offer templates. Offer attributes are available when optional offer integration is enabled.

Campaign, cell, and offer attributes are available only as shared attributes.

Marking attributes as required

When you create an attribute, you can specify special behavior characteristics for the attribute, including whether the attribute is Required. When you select this special behavior for an attribute and the attribute is implemented on a form, a red double asterisk (**) appears next to the corresponding field. The system also checks to assure that a value is supplied. Users receive an error if they try to save without selecting a value.

Note: You cannot mark an offer attribute as required. You define special behavior for offer attributes on a form-by-form basis.

Marking attributes and attribute groups as visible based on other attributes

You can define attribute rules that make attributes visible based on a single-select and single-select from database drop-down attribute. You can also make attribute groups visible based on a single-select and single-select from database drop-down attribute. Attribute rules are supported for plan, program, projects, project requests, marketing objects, invoice, and assets. Attribute rules for attribute groups are supported for plan, programs, projects, project requests, marketing objects, invoice, and assets.

To make attributes and attribute groups visible based on other attributes, you must define rules at the form level.

For example, you can configure a form rule to make the State attribute visible if the user selects `United States` from the Country attribute drop-down list.

Validations of dependent attributes and attribute groups are executed only when the attribute or attribute group is visible after rule evaluation.

For example, if you select any value other than `United States` for the country, then the State attribute is hidden. Therefore, if there are any validations on it (such as required behavior, check integer only, and so on) , those validations are skipped because the attribute is hidden after rule evaluation.

Standard attributes

A set of standard attributes is defined for all marketing objects.

The standard attributes follow.

Table 38. Standard marketing object attributes

Attribute	Description
Name	The display name for the marketing object.
Description	The text description entered for the marketing object, either when it was created or edited.
TemplateName	The ID for the marketing object template that this marketing object was created from. You set this ID when you create a marketing object template.
Code	The object code for the marketing object.
SecurityPolicy	The ID for the security policy associated with this marketing object. This ID is a foreign key into the **uap_security_policy** table, where you can find the name of the associated security policy.
Status	**Active** or **Deleted**. All marketing objects have a status of Active until they are deleted.
State	The current state for the marketing objects. Each marketing object type has its own set of states and state transitions.

Table 38. Standard marketing object attributes (continued)

Attribute	Description
CreatedBy	The user ID for the user who created the marketing object. User IDs are listed in the uap_user table.
CreatedDate	The creation date of the marketing object.
LastModUser	The user ID for the user who last modified the marketing object.
LastModDate	The date of the last modification made to the marketing object.
ComponentID	The internal name for the marketing object type that this marketing object is based on.

In systems that integrate Unica Plan with Unica Campaign, additional standard offer attributes are available. See the Unica Plan and Unica Campaign Integration Guide.

Attributes for Plan-Campaign integration

In systems that integrate Unica Plan and Unica Campaign, you use Unica Plan to create and enable campaign and cell attributes, and place them on forms and then into campaign project templates.

In systems that also enable offer integration, you use Unica Plan to perform these tasks for offer attributes to create offer templates.

When your templates are complete, users add and maintain campaign projects and offers in Unica Plan, and periodically publish the results to Unica Campaign.

Unica Campaign attributes

When Unica Plan and Unica Campaign are integrated, you create custom Unica Campaign attributes in Unica Plan. All Unica Campaign attributes are shared, and you use Unica Plan to add them to the forms that make up your campaign project templates.

When users create the linked campaign for a campaign project from a template that includes custom campaign attributes, the corresponding attributes are created in Unica

Campaign. If users change the data they enter in a field that is created by a campaign attribute after they create the linked campaign, they must update the campaign to send the new information to Unica Campaign. You use campaign attribute descriptions and form descriptions to inform users which fields require them to update the campaign.

Cell attributes

Cell attributes are Unica Plan attributes that map to Unica Campaign for use in Target Cell Spreadsheets. Unica Plan includes a set of default cell attributes that are included in every TCS®.

You can also create custom cell attributes in Unica Plan. When users create the linked campaign for a campaign project from a template that includes custom cell attributes, the corresponding cell attributes are created automatically in Unica Campaign.

Offer attributes

When offer integration is enabled, a set of standard offer attributes is supplied in Unica Plan to correspond to the standard attributes in Unica Campaign. You can also create custom offer attributes in Unica Plan.

All offer attributes are shared attributes.

To work with offer attributes, select **Settings > Unica Plan Settings > Template Configuration > Shared Attributes**.

Creating, editing, and deleting attributes

You can create, edit, and delete shared and local attributes.

You create, edit, and delete shared attributes from the **Shared Attributes** page: select **Settings > Plan Settings > Template Configuration > Shared Attributes**. You must manually enable shared attributes before you can use them on a form. After you enable a shared attribute, you cannot edit or delete it.

You create, edit, and delete local attributes directly on forms. Local attributes are enabled automatically when you create them.

Creating and enabling a shared attribute

You can create and enable a shared attribute by using Unica Plan shared attributes function.

1. Select **Settings > Plan Settings**.
2. Click **Template Configuration**.
3. Click **Shared Attributes**.

   The page that displays contains one section for each shared attribute category.
4. Click **Create a <*category*> Attribute** for the attribute you want to create.

   The **Create a New Shared Attribute** dialog opens.
5. Supply values to define the attribute.
6. Click **Save and Exit** to create the attribute and return to the **Shared Attributes** page, or click **Save and Create Another** to create the attribute and enter values for another new attribute.

   You can select a different attribute category.
7. On the **Shared Attributes** page, click **Enable** in the row for each new attribute to make it available for use on a form.

Editing a shared attribute

You can edit a shared attribute. Only edit a shared attribute if it is not imported.

1. Click **Settings > Plan Settings**.
2. Click **Template Configuration**.
3. Click **Shared Attributes**.
4. Click the attribute name of the attribute you want to edit.
5. Make the changes that you want and then click **Save and Exit**.

Deleting a shared attribute

You can delete a shared attribute only if it is not enabled. After it is enabled, it cannot be deleted.

1. Click **Settings > Plan Settings**.
2. Click **Template Configuration**.
3. Click **Shared Attributes**.
4. In the row for the attribute you want to delete, click **Delete**.

Shared Attributes list page

The Shared Attributes list page lists every shared attribute that is defined in the system. The attributes are organized by attribute category: form, grid, campaign, cell, and offer.

For each attribute, the following columns are displayed.

Table 39. Information about the Shared Attributes page

Column	Description
Display Name	The display name of the attribute. This name displays on forms.
Type	The attribute type.
Used By	A list of the forms that use this attribute.
Enable/Delete	Click **Enable** to make an attribute available for use on forms. Once you enable the attribute, **Enable/Delete** is replaced by **Enabled**. Click **Delete** to permanently delete an attribute that is not yet enabled.

The list page contains the following links.

Table 40. Links on the Shared Attributes page

Column	Description
Create a Form Attribute	Click to create an attribute to use on a form.
Create a Grid Attribute	Click to create an attribute to use on a grid.

Table 40. Links on the Shared Attributes page (continued)

Column	Description
Create a Campaign Attribute	Click to create an attribute that maps to Unica Campaign.
Create a Cell Attribute	Click to create an attribute that maps to Unica Campaign to use on a Target Cell Spreadsheet.
Create a Offer Attribute	Click to create an attribute that maps to Unica Campaign, if optional offer integration is enabled.

Creating a local attribute

You can create a local attribute in a form.

1. Open the form for which you want to create the attribute.
2. On the Add an Element tab, click **Create a New Custom Attribute**.
3. Specify the information for the attribute.
4. Click **Save and Exit** to create the attribute and return to the form or click **Save and Create Another** to create the attribute and display the attribute screen for a new attribute.

When you return to the form, expand the list of Form Attributes. The new attribute displays, and you can drag it onto the form and drop it into a group.

Editing a local attribute

You can only edit a local attribute after it is placed onto the form. You cannot edit an attribute that is in the **Form Attributes** list on the Add an Element tab.

1. Open the form for which you want to edit the attribute.
2. Click the attribute on the form.

   A dialog opens.

3. Click **Edit Custom Attribute**.
4. Make the changes that you want, then click **Save and Exit**.

Deleting a local attribute

You cannot delete an attribute that is on a form. You must remove the attribute from the form first.

1. Open the form that contains the local attribute you want to delete.
2. Locate the attribute on the form, then click to select it.
3. Click the **Delete** icon ( ) at the upper left.
   The attribute is removed from the form.
4. On the Add an Element tab, expand the Form Attributes list and then select the attribute in the **Custom Attributes** list.
5. Click **Delete Selected Attribute**.
   A confirmation dialog opens.
6. Click **OK**.

Attributes reference

The options that you see when you create or edit an attribute depend on the category and attribute type of the attribute.

However, many of the fields are the same across all categories and attribute types.

Standard attribute fields

You enter standard information for most attribute types when you create or edit them.

This table describes the standard fields for most attribute types. After you select the **Attribute Type**, more fields display. For more information about the attribute types, see Attribute types list (on page 208).

Table 41. Basic Options

Field	Description
Attribute Category	The category of attribute. • For a local attribute, defaults to Form Attribute. You can specify Grid Attribute. • For a shared attribute, the category is supplied based on the link you selected on the **Shared Attributes** page.
Attribute Type	The type of attribute. The attribute type controls the type of data the attribute holds, how it is entered into the database, and the options for the user interface controls that display. Available types vary depending on the attribute category you selected. See Attribute types list *(on page 208)*. **Note:** You cannot change the attribute type after you save the new attribute. If you select the wrong type, you must delete the attribute and create a new one.
Attribute Internal Name	The unique name of the attribute, used internally. Do not use spaces or special characters. Ensure that the attribute internal names do not start with a number.
Attribute Display Name	The display name for the attribute, which is used in the form and in the user interface of object instances. Spaces and UTF-8 characters are allowed.
Attribute Database Column Name	Name of the database column where values for the attribute are stored. By default, this name is the same as the **Attribute Internal Name**. Ensure that the attribute database column names do not start with a number. Do not exceed the character limit for your database, and avoid including words that are reserved for your database. See Database considerations for attribute database columns *(on page 208)*.

Table 41. Basic Options (continued)

Field	Description
Edit database column name	Select to edit the supplied **Attribute Database Column Name** value. Not available for system-supplied attributes.
Description	A description of the attribute.
Help Text	A short message that displays as a descriptive tooltip next to the user interface control.

This table describes the additional information that you define for most grid attributes:

Table 42. Grid Attribute Options

Field	Description
Sortable	Select this option to enable users to sort table data based on the values in this column.
Sort Type	The direction in which to sort values in this column. Select Ascending to sort in ascending order or Descending to sort in descending order.
Can group by	When this check box is enabled, this attribute groups the data in read-only rows that are based on the value of this attribute. This attribute applies only after you add it to a read-only grid.
Alignment	Alignment of the attribute in the table. You can select Left, Center, or Right. The default is Left.
Summary function	Available only when the **Attribute Type** is Decimal, Integer, Money, or Calculated. Performs a simple calculation on the column and displays it in a summary row at the bottom of the grid. Options are Sum, Average, Min, or Max. If no columns in the grid have a summary function, the summary row is not present.

This table describes the standard display information that you enter for all attributes.

Table 43. Display Options

Field	Description
Form Element Type	The type of user interface control to display for this attribute on the form. Examples of user interface controls include text field, check box, and drop-down list. The types that are available depend on the selected **Attribute Category** and **Attribute Type**.
Special behavior	The options are None, Required, or Read-only. • Select Required to prevent users from saving the form without supplying a value for this field. When selected, another text field for an **Error message to Display When Not Populated** displays. • Select Read-only to display the attribute, but not allow users to supply a value. The default is None. These behaviors do not apply to the Image or Calculated attribute types, or to system-supplied attributes.
Special behavior (Offer attributes only)	Offer attributes have different special behaviors, which synchronize their behavior with offers in Unica Campaign. The options are Parameterized, Static, and Hidden. • Parameterized corresponds to Required. Users cannot save the form without supplying a value for this field. • Static corresponds to None. • Hidden attributes do not display in the object instance. However, the value is sent to Unica Campaign when you publish the offer.

Database considerations for attribute database columns

Be careful when setting the **Attribute Database Column Name** value for an attribute. Your database has a set of reserved words, and using any of them for attribute names can cause errors when Unica Plan writes to the database.

Each database management system has a different set of reserved words. It is not feasible to list them all here, as they can change. The following is a short list to illustrate the issue; for an exhaustive list, refer to your database documentation.

DBMS	Some reserved words
MS SQL	Boolean, Browse, File, Group, Plan, Primary
Oracle	Cluster, Group, Immediate, Session, User
DB2®	Blob, Column, Group, Rollback, Values

If you are using an Oracle database, there is a 30-character limit for the **Attribute Database Column Name** value. For all other databases, the limit is 32 characters.

If you are integrating Unica Plan and Unica Campaign, avoid CLOB fields, as Unica Campaign does not support them.

If you are using a Microsoft™ SQL Server database, note that the Identity option for the primary key field is not supported.

Attribute types list

The following attribute types are available in Unica Plan.

Table 44. Attribute types

Attribute type	Description
Text - Single-Line	Presents a field for a single line of text.
Text - Multi-Line	Presents a field for a multiple-line text response.

Table 44. Attribute types (continued)

Attribute type	Description
Single-Select	Displays the items in a hardcoded drop-down list or as a set of radio buttons for users to select a single item. Not available for cell attributes. You can build rules to make other attributes visible in the form using single-select attributes.
Single-Select - Database	Displays the values from a database lookup table in a drop-down list or as a set of radio buttons for users to select a single item. Not available for cell attributes. You can build rules to make other attributes visible in the form using single-select database attributes.
Multiple-Select - Database	Displays the values from a database lookup table in a multi-select list or in a group of check boxes for users to select one or more items. Only available for form and grid attributes.
Yes or No	Provides controls for users to choose between two options (such as True and False). You can select a drop-down list, a set of radio buttons, or a group of check boxes.
Date Select	Displays a field that is formatted to accept a date, and a control for users to select a date from a calendar.
Integer	Displays a field that is formatted to accept an integer value, such as a percentile or weight.
Decimal	Displays a field that is formatted to accept a fractional number, such as 3.45.
Money	Displays a field that is formatted to accept a monetary value.

Table 44. Attribute types (continued)

Attribute type	Description
User-Select	Displays a list of all system users so that users can select a user. (Only available for form and grid attributes.)
External Datasource	This attribute type provides controls to open an LDAP search dialog and populate the field with Active Directory users. You can use this attribute type on the Summary tab or a custom tab. This attribute is only available for form attributes.
Calculated	Displays and stores the result of simple calculations on other fields.
URL Field	Displays a hyperlink to a web page. Only available for grid attributes.
Single Select Object Reference	References marketing objects on a form or grid. Only available for form and grid attributes.
Multi-Select Object Reference	References marketing objects on a form or editable grid. Only available for form and grid attributes.
Image	Displays a user-specified graphic. Only available for form attributes.
Creative URL	Provides a control for users to select an existing digital asset or add another asset. This attribute then displays a hyperlink to that asset. Only available for the standard Creative URL attribute.
Rich text type	Allows the user to enter the text and format it

The following attribute types are available for local attributes only.

Table 45. Attribute types available for local attributes only

Attribute type	Description
Object Attribute Field Reference	Displays an existing attribute of a marketing object. Only available for form and grid attributes.
Single List Object Reference	Used to reference marketing objects on a read-only grid. Only available for grid attributes.
Dependent fields	Adds fields whose values are constrained by another field.

Campaign, cell, and offer attributes

Only attribute types that exist in both Unica Plan and Unica Campaign are available for campaign and cell attributes.

For systems that also integrate offers, the same constraint applies to offer attributes with the following exception: when published to Unica Campaign, the attribute type for the Creative URL offer attribute changes to Text Field - String.

Table 46. Attribute types for campaign, cell, and offer attributes in Unica Plan

Attribute type	Campaign attributes	Cell attributes	Offer attributes
Text - Single-Line	X	X	X
Text - Multi-Line	X	X	X
Single-Select	X		X
Single-Select - Database	X		X
Multiple-Select - Database			
Yes or No	X	X	
Date Select	X	X	X
Integer	X	X	

Table 46. Attribute types for campaign, cell, and offer attributes in Unica Plan (continued)

Attribute type	Campaign attributes	Cell attributes	Offer attributes
Decimal	X	X	X
Money	X	X	X
User-Select			
External Data source			
Calculated	X	X	X
URL Field			
Single Select Object Reference			
Multi-Select Object Reference			
Image			
Creative URL			X
Rich text type			

Note: For attributes that have the Single-Select - Database attribute type, Unica Plan passes the lookup value (not the display value) of the selection to Unica Campaign. You determine the lookup value and the display value when you create the lookup table.

There is no attribute type in Unica Plan that corresponds to the "Modifiable drop-down list" that is available for custom attributes in stand-alone Unica Campaign.

Text attribute types

There are two types of text attribute types: single-line and multi-line.

To display text on a form or in a grid component, Unica Plan provides two attribute types.

- Text - Single-Line: Adds a small text box that allows users to enter and view only one line of text.
- Text - Multi-Line: Adds a larger, rectangular text box that allows users to enter and view multiple lines of text.

In addition to the standard attribute fields, you define these display options for text attributes.

Table 47. Display options for text attributes

Field	Description
Maximum length for the field	The maximum number of characters that users can enter in the field. If you do not want to enter a maximum length, select the Use CLOB check box.
Default value for the field	The value that is stored in the field if the user does not enter a value.
Use Clob	Use the CLOB data type. If this check box is selected, any value in the **Maximum length for the field** field is ignored. This option is not available for campaign attributes.

Rich text type

Rich text attribute allows the user to enter the text and format it.

- It allows the change in font/bold/italic/bullet lists/numbered lists.
- It allows coloring.
- It allows formatting as table.
- It supports copy or paste from Word or Excel.
- It is available as Form attribute only.

Note: Forms with Rich text are not supported in Offer template. This attribute type does not support copy of image.

To display formatted text on a form component, Unica Plan provides the Text - rich text type. This rich text type uses Clob. In addition to the standard attribute fields, you define the default value option.

Table 48. Display option for Rich text type

Field	Description
Default value for the field	The formatted value which is stored in the field if the user does not enter a value.

Single-Select attribute types

You add the Single-Select attribute type to a form when users can choose one value from a relatively short list of options that do not change often.

You define the options that users can choose when you create the attribute in the **Allowed values for the field** field. You also define whether the attribute displays as a drop-down list or as a radio button group in the **Field Type** field. Users can select only one choice from the list.

You can define form attribute rules for this attribute type. The rules make another field required if the user selects a certain value from this attribute.

Note: This attribute type is not available for cell attributes.

Table 49. Options for Single-Select attributes

Field	Description
Allowed values for the field	A text box that you use to enter a new value. After you enter a value, click **Add** to add the value to the list of allowed values. The list of values is presented on the form in the same order that it displays in this list. Click the controls to the right of this list box to organize the list:

Table 49. Options for Single-Select attributes (continued)

Field	Description
	• **Remove**: Deletes the selected value. • **Up**: Moves up the selected value in the list. • **Down**: Moves the selected value down in the list.
Default value for the field	A drop-down list that you use to specify the default value for the attribute. You can select any of the **Allowed values**.

Single-Select - Database attribute types

The Single-Select - Database attribute type functions like the Single-Select attribute type, except that the list of choices comes from a database table that contains valid items. If users can select from a list of options that is relatively long, or might change, consider the Single-Select - Database attribute type.

You can make a Single-Select - Database attribute dependent on the value in another field. For example, you might make the choices in a list of cities dependent on which state is selected.

You can also define form attribute rules for this attribute type. The rules make another field visible if the user selects a certain value for this attribute. For example, if the user selects Canada or France from a single-select database attribute that contains various countries, you might make a Province attribute visible elsewhere in the form.

Note: This attribute type is not available for cell attributes.

When you create a Single-Select - Database attribute, you specify the lookup table and other information as described in the following tables.

Additional Basic Options fields

Single-Select - Database attributes also have the following basic options.

Table 50. Options for Single-Select - Database attributes

Field	Description
Filter database table name	A value to use to filter the drop-down list of table names for the **Use values from this database table** field. Only those lookup tables with the specified text in their table names are included in the list. If this field is empty, the list contains all lookup tables in the database.
Use values from this database table	Select the table that contains the values that you want to display to users.
Key Column	Select the primary key for the table.
Display Column	Select the database column that contains the values you want to display on the form.
Sort-by Column	Select the column that determines the order of the list when it displays on the form.
Ascending / Descending	Select the sort order for the list.
This field is dependent on the following column	To make the values that display in this list dependent on the selection that the user makes for another field, select the check box and specify the database column from the drop-down list. This option is not available for global attributes.

Additional Grid Attribute Options field

Single-Select - Database attributes also have the following grid option:

Table 51. Grid options for Single-Select - Database attributes

Field	Description
Do Not Cache Lookup Values	Select to update the list of options every time a user saves or refreshes the grid.

Additional Display Options fields

Single-Select - Database attributes also have the following display options:

Table 52. Display options for Single-Select - Database attributes

Field	Description
Default value for the field	Select a default value for the attribute, or leave this field blank if you do not want to have a default value for the attribute. (The list contains all values in the database column that is specified in the **Display Column** field.)

Multiple-Select - Database attribute types

You can define an attribute that allows the user to choose multiple values from a valid list of options.

For example, when users specify the products for a particular marketing campaign for a bank, they can select one, two, or all of the following options:

- 1-year CD
- 5-year CD
- Bank Credit Card

To offer multiple selection in a field, you use an attribute with a type of Multiple-Select - Database. Creating a multiple selection attribute is similar to creating a Single-Select attribute, with some additional setup.

Note: This attribute type is not available for campaign, cell, or offer attributes.

When you create a Multiple-Select - Database attribute, you must specify the lookup table, and other information as described in Single-Select - Database attribute types (on page 215).

When you create Multiple-Select - Database attributes, keep in mind the following restrictions:

- The value in the **Attribute Database Column Name** field must be unique across attribute categories. (You cannot use the same database column name for both a grid attribute and a form attribute.)
- You must not change the data type of the key column after you create the attribute.

You must also specify the following display option.

Table 53. Option for Multiple-Select - Database attributes

Field	Description
Multi-Select Joint Table Name	The name of the Joint table to use for this attribute. Every Multiple-Select - Database attribute must have a unique Joint table.

Yes or No attribute types

You can create an attribute that can only have one of two values (true/false, yes/no, etc.).

For example, you can have a form where the user enters either Yes or No to a question. Use the Yes or No attribute type for this purpose.

A Yes or No field can display as a checkbox, a drop-down list, or a radio button group.

Note: This attribute type is not available for offer attributes.

Yes or No attributes have the following additional basic options:

Table 54. Options for Yes or No attributes

Field	Description
Default Value	Specifies the default value for the attribute. You can choose Yes, No, or Not Available. (Not Available is the same as No if you chose Checkbox for the form element type.)

Table 54. Options for Yes or No attributes (continued)

Field	Description
Display name fields	Specify the display name for each possible value. The defaults are Yes, No, and Not Available. Display names are not used if you select Checkbox as the form element type in the display options.
Sort Order fields	The values in these fields specify the order in which the possible values are listed on the form. By default, the order is Yes, No, and Not Available. If you clear the Sort order field for a value, that value does not display to users. The sort order does not apply if you choose Checkbox as the form element type in the display options.

Decimal attribute types

You can use Decimal attributes to display non-integer values.

For example, use a decimal attribute to represent a field that contains percentages.

Decimal attributes have the following additional basic option:

Table 55. Option for Decimal attributes

Field	Description
Number of decimal places for the field	The number of digits users can enter after the decimal point. The maximum value is 7.

Decimal attributes have the following additional display option:

Table 56. Display option for Decimal attributes

Field	Description
Default value for the field	The value to use if the user does not enter a value.

Money attribute types

Money attributes represent monetary values, such as salary or item cost. The currency sign is set from the locale information for the user.

Money attributes have the following additional basic option:

Table 57. Basic Option for Money attributes

Field	Description
Number of decimal places for the field	The number of digits users can enter after the decimal point. The default value is 2. You can specify more decimal places if the attribute displays a conversion rate (typically 5 decimal places) or per unit costs that are in micro-cents. The maximum value is 7.

Money attributes have the following additional display option:

Table 58. Display option for Money attributes

Field	Description
Default value for the field	The value to use if the user does not enter a value.

Calculated attribute types

A Calculated attribute is a read-only field whose value is calculated based on a specified formula.

When you save a form that contains calculated attributes, Unica Plan checks the formulas to ensure that they are valid.

A Calculated attribute has the following additional basic options.

Table 59. Options for Calculated attributes

Field	Description
Formula	The formula that calculates the value. Any attributes that are used in the formula must be included on the same form as the calculated attribute.
Number of decimal places for the field	The number of digits to display after the decimal point

Formula syntax

You can perform the following binary operations.

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)

You can perform the following operations on an arbitrary number of comma-separated operands.

- **Sum**: for example, `Sum(Salary, 1000, Bonus)`
- **Avg**: arithmetic average, for example, `Avg(BudgQtr1, BudgQtr2, BudgQtr3)`
- **Min**: select the minimum value, for example `Min(IQ, 125)`
- **Max**: select the maximum value, for example `Max(Sale1, Sale2, Sale3, Sale4)`

An operand can be any of the following.

- A decimal constant (for example, 2.5).
- The attribute internal name of an attribute on the current form of the following types: Money, Integer, Decimal, or Calculated. If the formula references an attribute that is not included in the form, saving the form results in an error.

Calculated attribute example

Assume the **Wages** form contains the following currency fields: **BaseSalary**, **Bonus**, **Insurance**, and **FedTax**. You could create a calculated field named **Net pay**, and enter the following formula for it: `BaseSalary+Bonus-FedTax-Insurance`

Example to avoid

Since you can reference one calculated field within another, take care to avoid infinite recursion. For example, consider a form with the following attributes.

- **Salary**: an integer or money attribute
- **Commission** = Salary + (Bonus * 0.10)
- **Bonus** = (Commission * 0.5) + 1000

The **Commission** and **Bonus** attributes refer to each other, creating an infinite loop when the system attempts to calculate the values.

Grid attribute example

Calculated fields can be used in grids and on forms. A simple example is if a grid contains columns for units and cost per unit, you can create a column for the grid to represent the total cost: `Units * CostPerUnit`

URL Field attribute types

To add a hypertext link to grids and forms, you use the URL Field attribute. For grids, you can specify a link to a URL for each row that gets added to the grid.

For URL Field attributes, you specify database columns that hold the actual URL (**DB Column for URL**) and the link text that displays in the final grid or form (**Database Column**).

Note: This attribute type is not available for campaign, cell, or offer attributes.

For example, suppose that you have a grid that contains vendor data, and for each vendor you are required to specify a website for the vendor. In the Forms Editor, you can create a URL Field attribute as follows.

Table 60. Example settings for adding a vendor URL field to a grid

Field	Value	Description
Attribute Type	URL Field	Specifies the URL Field attribute type.
Attribute Internal Name	vendorURL	Unique identifier for the attribute.
Attribute Display Name	Vendor URL	The label that appears in the user interface.
Attribute Database Column Name	textURL	The database column added to hold the display text for the link.
DB Column for URL	linkURL	The database column added to hold the actual URL. You do not have to enter **http://**. For example, to link to Google, you can enter either `www.google.com` or `http://www.google.com`.

After you set up an object in Unica Plan that uses this form, users who add grid rows or populate a form specify a URL. In a grid, users can specify a URL for each row. Users can then click a link to open the website in a new window.

Object Reference attribute types

You use Object Reference attributes to associate marketing objects with projects or other marketing objects. An Object Reference attribute creates a field with a selector attached.

Users can search in the selector for a particular marketing object and then add it to the project or marketing object they are creating or editing.

Multi Select Object Reference and Single Select Object Reference attributes are similar; however, the resulting user interface field can contain multiple entries versus a single entry.

You cannot add these attributes to read-only grids. To add marketing object references to read-only grids, use the Single List Object Reference attribute.

Note: These attribute types are not available for campaign, cell, or offer attributes.

To specify an Object Reference attribute, you enter the following information specific to this attribute type:

Table 61. Options for Object Reference attributes

Field	Description
Marketing Object Type	The marketing object type of the items you want on the list that is displayed to the user.
Template ID	The ID of a specific template for the specified marketing object type. If you select the **Auto Create** check box below, this template is used to create the object. Otherwise, only marketing objects that users create from the specified template display to users.
On Click go to	Select the destination screen when you click the object link on a form: • **Summary Tab**: opens the summary page of the marketing object • **Analysis Tab**: opens the analysis page **Note:** This field is available only for grids.

Table 61. Options for Object Reference attributes (continued)

Field	Description
Modifies	Use this option when the object that contains this form is intended to change or update the contents of a marketing object (for example, a Change Order or Work Request project). **Note:** This field is available only for forms.
References	Use this option to indicate that the marketing object is only referenced, not modified. **Note:** This field is available only for forms.
Auto Create	Select this option to create an "empty" marketing object if users choose a template that contains a form with this attribute when they create an object. Note that: • This check box is not available for Multi Select Object Reference attributes. • This check box has no effect if the form is added to a marketing object, as marketing objects cannot automatically create other marketing objects. This field is available only for Single Select Object Reference attributes on forms.

Image attribute types

To enable users to display a graphic on a tab in a project or marketing object, you use an Image attribute.

The attribute creates a display area for the image and a field with a Browse button so users can select the graphic to display.

Note: This attribute type is not available for grid, campaign, or cell attributes.

Creative URL attribute types

The Creative URL attribute gives users the ability to include a digital asset from a Unica Plan asset library in an offer.

The Creative URL attribute type provides the user interface controls for users to select an asset and then display a hyperlink to the asset.

Table 62. Options for Creative URL attribute types

Field	Description
Default value for the field	Provides a control for users to select an existing digital asset from an asset library, or to add an asset and then select it. The corresponding field then displays the name of the selected asset as a link to digital file. The same controls display in the user interface for offer instances that include an attribute with this type.

Object Attribute Field Reference attribute types

You add a local Object Attribute Field Reference attribute to a specific form to display information about a marketing object that is linked to the form.

For example, if the form contains a Single Select Object Reference attribute for a marketing object named **Brochure01**, you can also add an Object Attribute Field Reference attribute to display any of the attributes for **Brochure01** (such as its status).

Note: You cannot create an Object Attribute Field Reference attribute that corresponds to a Multi Select Object Reference attribute.

The resulting object attribute field information is for display only. Users cannot edit it.

This attribute type is only available as a local attribute.

You can reference both standard marketing object attributes and custom attributes. For custom attributes, you must know the attribute name and the name of the form that contains the attribute in the marketing object template. For a list of standard marketing object attributes, see Standard attributes (on page 198).

Object Attribute Field Reference attributes require the following additional basic options:

Table 63. Additional basic options for Object Attribute Field Reference attributes

Field	Description
Attribute Name	The name of the marketing object attribute you want to reference. To reference a standard attribute, select it from the drop-down list. To reference a custom attribute, enter the name in the form `<form_name>.<internal_name>` where • `form_name` is the name of the form that contains the custom attribute in the marketing object template • `internal_name` is the value of the Attribute Internal Name field for the custom attribute
Reference Object	The internal name of the attribute on the current form that references the marketing object.

Single List Object Reference attribute types

You add a local Single List Object Reference attribute to a specific form.

You can reference marketing objects on a list, similar to how you use a Single or Multi Select Object Reference attribute to reference marketing objects on a grid.

You can display a grid as a list for an object (project or marketing object). For details, see Display a grid as a list (on page 176).

This attribute type is only available as a local attribute for grids.

Example: creating lists of marketing objects (on page 179) provides an example of using a Single List Object Reference attribute.

To specify a single list object reference attribute, you must enter the following information specific to this attribute type.

Table 64. Options for Single List Object Reference attributes

Field	Description
On Click go to	Used to choose the tab of the object that opens when you click an object link from a list view. You can choose to navigate to either the Summary Tab or the Analysis tab.
Object Reference ID Column	Used to display a list of marketing objects in a list view. If you select this option, the **Object Reference Type Column** field becomes active. Enter a value that corresponds to the object instance ID column for the objects you are mapping.
Object Reference Type Column	Used in concert with the **Object Reference ID Column** field to display marketing object references on list views. Enter a value that corresponds to the object type column for the objects you are mapping.
Grid Object Reference Column	Used to display a grid as a list view. If you select this option, you must enter the following information in the field: • The name of the form that contains the grid. This field indicates the name of the form in Unica Plan, chosen when the form was uploaded into Unica Plan. • The internal name of the Single Select Object Reference attribute as defined on the grid. The syntax is `<form_name>.<attribute_name>`.

Table 64. Options for Single List Object Reference attributes (continued)

Field	Description
	For example, if you have a grid that has a Single Select Object Reference attribute with an internal name of **Brochure**, and the grid is contained on a form that is named **EventCollateral** in Unica Plan, you enter **EventCollateral.Brochure** in this field.

Dependent field attribute types

Dependent fields are attributes whose values are constrained by another field.

For example, if you want to have a field that displays all the cities for a selected state, you could make the city field dependent upon the state field. You can make an attribute dependent on another attribute only when you add it to a specific form. That is, when you create a shared attribute, you cannot make it a dependent field when you create it, only within the context of the forms you add it to.

This section presents the city/state example.

First, you need to create the lookup tables for cities and states. Here are the first few rows of these two tables.

The `lkup_state` table:

state_id (primary key)	state_name
1	Massachusetts
2	New York

The `lkup_city` table:

city_id (primary key)	city_name	state_id (foreign key pointing to primary key for lkup_state)
1	Boston	1
2	Cambridge	1

city_id (primary key)	city_name	state_id (foreign key pointing to primary key for lkup_state)
3	New York	2
4	Albany	2

Once you have these tables, you create the parent (state) and child (city) attributes.

For the state attribute, use the following values:

Field	Value
Database Column	state_id
Use values from this database table	lkup_state
Key Column	state_id
This field depends on the following column	Leave this box clear.

For the city attribute, use the following values:

Field	Value
Database Column	city_id
Use values from this database table	lkup_city
Key Column	city_id
This field depends on the following column	Check this box and select state (the Internal Name you defined for the state attribute).

Note the following:

- You can have a Multiple-Select - Database attribute be dependent upon a Single-Select - Database attribute, but not the other way around. In the example above, the city field could be a Multiple-Select - Database attribute, but not the state field.
- You can sort lookup values based on either the text description or the ID for the lookup value.

Visible attributes and attribute groups based on other attributes

You can make an attribute visible based on what was selected for any combination of all the other single-select and/or single-select database drop-down attributes. You can also make attribute groups visible based on values of other attributes. This feature makes forms much simpler to fill in, only containing the attributes or attribute groups for which the user needs to provide input based on everything else they have already filled in on the form.

Important: In previous versions of Plan, you could make an attribute or attribute group required based on values of other attributes. This feature is not available from version 10.0.0.1 onwards. From version 10.0.0.1 onwards, making attribute groups visible based on values of other attributes is a new feature. It replaces the earlier feature of making attributes or attribute groups required.

Upgrading to version 10.1: If you have used the required attributes feature in previous versions and have upgraded to version 10.1, the rules for required features are not retained in version 10.1. To make an attribute required, you must edit the attribute and select the **Required** option for the attribute. However, this makes the attribute a required attribute on all the forms that it is included in.

Attribute rules are supported for plan, program, projects, project requests, marketing objects, invoice, and assets. Rules for attribute groups are supported only for plan, program, projects, project requests, and marketing objects.

To make attributes or attribute groups visible based on other attributes, you must define rules at the form level.

For example, you can configure a form rule to make the State attribute visible if the user selects `United States` from the Country attribute drop-down list.

Validations of the dependent attributes and attribute groups are executed only when the attribute or attribute group is visible after rule evaluation. For example, if you select any value other than `United States` for the country, then the State attribute is hidden. If there are any validations on it (such as required behavior, check integer only, and so on) , those validations are skipped because the attribute is hidden after rule evaluation.

This rule is executed at the instance level and in the **Preview** page of the **Form Editor**.

At the instance level:

- In non-edit mode (after instance creation when the form is loaded).
- In edit mode (when you edit the form).
    - On form page load in edit mode.
    - When user changes the value of single-select drop-down list in edit mode.
- In wizard mode (at instance creation).

Editing rules behavior

After you create rules for visible attributes and attribute groups that are based on other attributes, the following criteria for editing rules apply.

- You receive a warning if you try to delete a value from a single-select attribute and a rule is defined on that value. You can delete the value only after you delete the associated rules.
- If you delete any value from a single-select database attribute, Unica Plan checks to see whether rules are defined on that value. If a rule is defined on the value and you delete the value, the rule is flagged as invalid. You can then modify the rule.
- If you try to delete an attribute or attribute group that is used for defining a rule, you receive a warning. If you delete the attribute or attribute group anyway, the attribute or attribute group is removed from the rules.
- If you try to make an attribute behavior read-only, and that attribute is used in a rule, you receive a warning. To make the attribute read-only, you must remove it from the defined rule.

- If you try to change the look-up table, key column, or display column of a single-select database attribute that is used in a rule, you receive a warning. You must remove the attribute from any existing rules before you change the look-up details.
- When you edit a form with existing rules, the form is not automatically republished. To apply a changed rule, republish the form.
- When you try to disable any look-up values of single-select database attribute through manage look up option from form listing page, you receive a warning message. You must remove the values from existing rules before you disable the look-up values.
- If you add an attribute to an attribute group that is selected for a rule, the attribute is automatically included in the rule.
- If you add multiple attribute groups to a form rule, ensure that the attribute groups have unique display names. Otherwise, the form rule does not operate correctly.

Importing and exporting behavior

After you create rules to make attributes and attribute groups visible that are based on other attributes, the following criteria for importing and exporting forms with rules apply.

- If you export a form with attribute rules, the rules are also exported. Such forms can be imported to another Unica Plan system along with the attribute rules.
- If the form you import contains corrupted or broken rules, then you receive a warning, so you can fix the rules.

Column layout adjustment details

When any dependent attributes or attribute groups are shown or hidden based on rule evaluation, the layout of Unica Plan is adjusted automatically.

If an attribute or attribute group is hidden, then all rows for the attribute or attribute group are hidden and the corresponding number of rows below the hidden rows automatically move up in place of hidden rows. This adjustment occurs in the following layouts.

- If the group has a one column layout (it will always have one attribute or none)
- If the row has an attribute

- If the group has a two column layout
- If the row has only one attribute (it is always placed at left side)

In a row with two attributes but only one attribute is hidden, the following adjustments occur in the layout based on which column contains the attribute (or attributes) that is hidden.

- If the left side attribute is hidden, then the right side attribute and the next row below it remains as is.
- If the right side attribute is hidden, then the left side attribute and the next row below remains as is.
- If both attributes are hidden, then the whole row is hidden and the next row below moves up automatically.

Marking attributes and attribute groups visible based on other attributes

You can decide which attributes to make visible in a form based on the values of other attributes in that form. For example, if the user selects United States in the country attribute, you can make visible its dependent State attribute. You can also make attribute groups visible based on values of other attributes in a form.

You can build rules on single-select and single-select from database attributes only. When you select the option value from the drop-down that triggers the rule, another attribute or attribute group becomes visible.

1. Open or edit a form.
2. Add attributes and attribute groups to the form as needed.
   If you want to build a rule to make State visible when users select United States as the Country, you must create attributes for Country and State.
3. Click **Create/Edit Form Attribute Rules**.
   The Rule Builder opens with existing rules and an area to create new rules.
4. Create or edit rules as needed by following these steps:

a. In the **Select an Attribute** lisbox, select the attribute that triggers the rule. Then, in the next listbox, select the **In** or **Not In** option to indicate whether the rule is triggered by the selected attribute value or attribute values that are not selected. In the third listbox, select the attribute value for the rule. For example, to trigger the rule when United States is selected as the Country, select Country in the **Select an Attribute** listbox, the **In** option, and United States as the attribute value.

b. In the **Attributes and Attribute Groups** box, select the attribute or attribute group that you want to make visible and use the **right arrow** button to move the selected attribute or attribute groups to the **Selected Attributes and Attribute Groups** box. For example, to make the State attribute visible when United States is selected, move the States attribute to the **Selected Attributes and Attribute Groups** box.

   When you select an attribute group, the attribute group header and all its attributes are added to the **Selected Attributes and Attribute Groups** box.

c. Click **Save Compound Condition** to save the rule. The rule is displayed in the **Condition** tab at the top of the screen. The **Preview** tab shows the rule's code.

d. Repeat steps a through c for each rule that you want to add to the form. Use the **Up** and **Down** arrows to specify the sequence for executing the rules.

5. Click **Accept and Close** after you create, edit, and delete attribute rules.
6. **Save** the form.

When you add rules to forms, be careful not to make conflicting or circular rules.

If you edit rules on a form that is already published, you must publish the form again to use the modified rule.

When you change or delete attributes or attribute groups, Unica Plan warns you if that the attribute or attribute group is used in a rule. If you delete the attribute or attribute group anyway, the attribute or attribute group is removed from the rules. If a rule is invalid, it is flagged in the rule builder for your attention.

All attributes and attribute groups that you do not build rules for are by default made visible and are shown.

Chapter 12. Metrics

Metrics measure the performance of an object and are always numeric. Typical metrics include financial metrics, such as cost and revenue, and performance metrics, such as the number of contacts and the number of responses in a particular marketing campaign.

You can define a metric to calculate its value based on other metric values. For example, you can define campaign profit as the revenue minus the cost. You can also define metrics that roll up from projects to programs and from programs to plans.

You associate metrics with metrics templates, which are in turn associated with the templates for other objects. As a result, when you add an object, the metrics identified through both templates display on the Tracking tab.

To organize the metrics you define, you can create groups within your metrics templates. Once established, a group can be added to other metrics templates as needed. You can also define metrics dimensions to track different values for each metric: for example, Actual value, Target value, Projected value (optimistic) and Projected value (pessimistic). Metrics dimensions apply to all metrics templates, and appear to users as entry columns on the Tracking tab.

Types of metrics

In addition to metrics that are user-entered, there are three types of metrics: computed, rollup, and planned. You set the metric type when you add the metric to a particular metrics template. As a result, the same metric can be computed for a project and roll up in a program or plan.

Computed metrics

To specify that a metric is computed, rather than user-entered, you check the **Computed** box and enter a formula when you add the metric to a metrics template.

For example, you create an ROI (return on investment) metric. When you add it to a metrics template, you define that it is computed using the following formula:

```
((TotalRevenue - TotalCost)/TotalCost)*100
```

- When you define a formula for a metric, you use the Internal Name defined for each metric in the formula.
- You can use the following operators in the **Formula** field: +, -, *, /, SUM, AVG, MIN, MAX, and ROLLUP.

Note: If you include a NULL value in the formula, these operators treat the NULL value differently. Aggregation functions (SUM, AVG, MIN, and MAX) ignore the NULL value. Arithmetic calculations treat the NULL value as 0; however, if you enter #/0 or #/NULL, Unica Plan displays `#DIV/0!`

Metric rollups

When you add a metric to a metrics template, you can specify that the metric will "roll up" from the child object to the parent object. For example, project metrics can roll up to the parent program level, and program metrics roll up to the parent plan level.

Metrics that roll up can display on the Tracking tab of the parent object.

- Any project metrics that you configure for rollup display in the Project Rollups table on the Tracking tab of the parent program.
- Any program metrics that you configure for rollup display in the Program Rollups table on the Tracking tab of the parent plan.

For example, to track the number of responders to all projects in a program, you define the following metrics:

- **NumberOfRespondersPassed**, representing the number of responders from the projects.
- **NumberOfProgramResponders**, representing the number of responders in the program.

Next, you add the metrics to metric templates as follows:

- For the project metric template, add a group (for example, **Performance**) and add the **NumberOfRespondersPassed** metric to it. Do not select **Computed** or **Roll-up** when adding the metric to the group.
- For the program metric template, add a group (for example, **Performance**) and add the **NumberOfProgramResponders** metric to it. Do not select **Computed** or **Roll-up** when adding the metric to the group.
- For the program metric template, add the **NumberOfRespondersPassed** metric in two places:
    - To the metrics template, without a group: click **Manage Metrics** and check the **Roll-up** box.
    - To any group (typically, a group that matches the group in the project metric template: in this example, **Performance**). Clear the **Roll-up** box. Check **Computed** and enter the following formula: `NumberOfProgramResponders +ROLLUP(NumberOfRespondersPassed)`

Planned metrics

To include goals and performance expectations for your plans and programs, you can identify metrics as being planned. Planned metrics are similar to rollups in that values are inherited from one object in the hierarchy to another, but in the opposite direction: child objects inherit planned metrics from parent objects.

To define a planned metric, check both the **Roll-up** and **Is Planned** boxes when adding the metric to the plan or program metrics template. Planned metrics appear in the Planned column of the metrics table on the Tracking tab of each child object.

Metric creation overview

To add metrics to objects, create metric templates.

1. From Unica, select **Settings > Plan Settings > Template Configuration > Metrics**.
2. Add metrics dimensions (optional).
3. Add metrics.
4. Add a metrics template.

5. Manage the metrics on the metrics template by adding them to groups or to the template itself, and by defining the type for the metric.
6. To localize the metrics, export, translate, and import a properties file for each locale (optional).

Working with metrics, metrics dimensions, and metrics templates

To work with metrics, metrics dimensions, and metric templates, go to **Settings > Unica Plan Settings > Template Configuration > Metrics**.

Metrics and metrics templates are sorted alphabetically by the ID field. Metrics dimensions are sorted by the order in which they were added.

- To add a metric, metrics dimension, or metrics template, click the corresponding **Add** link on the Metrics Templates page. You can add an unlimited number of metrics and metrics templates. You can add up to five metrics dimensions; each one applies to all of your metrics templates.
- To edit a metric, metrics dimension, or metrics template, click the corresponding **Edit** link on the Metrics Templates page.
- To delete a metric, metrics dimension, or metrics template, click the corresponding **Delete** link on the Metrics Templates page. You cannot delete a metric or metrics template that is used by any other object: the **Delete** link is disabled if the item is used by another object.

Metrics properties

When adding or editing metrics, you supply values for the following fields.

Table 65. Metrics properties

A two-column table describing the properties of a metric.

Property	Description
Internal Name	The ID of the metric. Avoid spaces and special characters.

Table 65. Metrics properties

A two-column table describing the properties of a metric.

(continued)

Property	Description
	Metrics are sorted on the Metrics Templates page by this Internal Name. When you identify a metric as computed in a metrics template, you use the internal name to identify each metric in the formula you supply. **Note:** The Internal Name must consist of English alphanumeric characters.
Display Name	The name of the metric when used in Unica Plan. **Note:** Limit this name to three 10-character words (or less). The display limit in metric rollup tables within Unica Plan is 32 characters. For example, "Mailed Client Savings" displays in full, while "Savings Mailed to Prospective Clients" does not display completely. You can translate the Display Name using properties files.
Description	Descriptive text for the metric. This text is useful for determining the purpose of the metric.
Unit Type	The type of metric. Choose from Number, Decimal, Percent, or Money.
Display Format	How the metric appears on the Tracking tab for an object. Typically, the Display Format corresponds to the Unit Type. Select:

Table 65. Metrics properties

A two-column table describing the properties of a metric.

(continued)

Property	Description
	• # - number or decimal • #% - percent • $# - money While you select $# when you define monetary metrics, users can enter values for the metric in the currency for their defined locale.
Precision	The number of digits of precision, up to 9. The precision controls the number of digits after the decimal point for the metric value. Values are rounded using the "half-up" rule. If the digit to the left of the discarded digit is odd, round up. If the digit to the left of the discarded digit is even, round down. For example: • 9/2=4.5 Since the number before 5 is 4 (even number), round down to 4. • 7/2=3.5 Since the number before 5 is 3 (odd number), round up to 4.

Metrics dimensions properties

When adding or editing metrics dimensions, you supply values for the following fields.

Table 66. Metrics dimensions properties

A two-column table describing the properties of a metrics dimension.

Property	Description
Display Name	The name of the dimension to be used in Unica Plan. Appears as a column heading on the Tracking tab when users enter metrics for an object. You can translate the Display Name using properties files.
Description	Descriptive text for the dimension. This text is useful for determining the purpose of the dimension.
Type	• Actual: Use to capture metrics that are entered manually or loaded into Unica Plan from Unica Campaign or some other tracking software. • Target: Use to capture values for metrics that your organization uses for planning and setting targets. Target dimensions are the only dimensions that display in the wizards used to create objects. • Other: Use for any dimension that is not Actual and that you do not want to display in the wizards used to create objects.

Creating metrics templates and metric template groups

A metrics template is a collection of metrics. You add metrics to Unica Plan by adding metrics templates.

Likewise, you attach metrics to an object template by selecting a metrics template.

Each metric template works with only one object type: **plan**, **program**, or **project**. Because each object template can use only one metrics template and there is only one template file for plans, you should not define more than one metrics template with a type of **plan**.

A metric can belong to more than one metric template.

Metrics in a template can be, but do not need to be, organized into metric groups. A metrics template can contain a mix of grouped and ungrouped metrics.

Creating or editing a metrics template

Before you can add metrics to object templates, you organize the metrics into metrics templates.

1. Click **Add Metrics Template** or **Edit** on the Metrics Templates page.
2. Enter or edit the **Internal Name**, **Display Name**, and **Description fields**.
3. Select a type of object to use this metrics template: **plan**, **program**, or **project**.

   Note: Since there is only one template for plans, do not define more than one metrics template with a type of **plan**.

4. Add metrics to the template.
   - To add a metric to the template without using a group, click **Manage Metrics**.
   - To add a group of metrics, click **Add Metrics Group**.

     You can select any existing group or create a group.
5. Select an individual metric and define the properties for the metric in this template.
   - If the metric is entered individually by users in the object, clear the **Computed by Formula**, **Roll-up**, and **Planned** check boxes.
   - If the metric is computed, select the **Computed by Formula** check box and enter a **Formula**.
   - If the metric is collected from other metrics, select the **Roll-up** check box. Rollup metrics are available in plan or program templates only.
   - If the metric is planned, select the **Roll-up** and **Planned** check boxes. Planned metrics are available in plan or program templates only.
6. Click **Save changes** to save the metrics template.

Important: When you edit a metrics template, the changes affect new objects only.

For example, you have a project that uses the "basic campaign" metrics template. You add a metric to this metrics template. The existing project does not get the new metric. However, when you add a project that uses the "basic campaign" metrics template, it does include the new metric.

Metrics groups

In a metrics template, you create metrics groups to organize metrics that are similar or to share common sets of metrics in multiple metrics templates.

After creating a metrics template, you can add a metrics group. Click **Manage Metrics** next to the name of the metrics group to add metrics to the group. You can also order the metrics within the group in the Manage Metrics dialog. The metrics show in this order in your reports.

When you change a metrics group, the change affects all of the metrics templates that include the group. For example, you create a "Financials" metrics group in your "basic campaign" metrics template. Later, you add the "Financials" metrics group to your "seasonal campaign" metrics template. You then edit the "seasonal campaign" metrics template and add a metric to the "Financials" metrics group. The new metric is now also included in the "basic campaign" metrics template.

You can delete a metrics group from a metrics template. If another metrics template contains the same metrics group, the metrics group still exists. You can still add the metrics group to other metrics templates. If you delete all instances of a metrics group from all metrics templates, you also delete the metrics group from Unica Plan.

Exporting and importing metrics templates

If you have multiple Unica Plan systems, you can transfer the metadata for your metrics from one instance to another by exporting and importing metrics templates.

To transfer metrics templates between systems, select **Settings > Plan Settings > Data Migration** and click either **Export** or **Import** next to **Templates**. Select the Metrics check box to create or receive a compressed archive file.

For more information about data migration, see Exporting and importing metadata (on page 302).

Note: To import a metrics specification file that was exported from a pre-8.5 version, on the Metrics Templates page click **Import Metrics Template** and select the XML file.

Chapter 13. Multiple locale support

For organizations with Unica Plan users in multiple locales, you can translate the labels and text strings in the user interface for each locale.

The **supportedLocales** and **defaultLocale** configuration properties, found under **Settings > Configuration > Unica Plan**, identify the locales for your organization. These properties are set during installation.

You can localize the following objects in Unica Plan:

- Standard marketing object types and custom marketing object types. See Localized object types *(on page 247)*.
- Template properties. See Multi-locale support for templates *(on page 249)*.
- Form attributes. See Localizing forms *(on page 253)*.
- Metrics. See Localizing metrics *(on page 256)*.
- Alerts. See Customizing an alert notification message *(on page 282)*.
- Lists, including user roles for project templates. See Localizing lists *(on page 257)*.

Note: Changes to the `<Plan_Home>/messages/com/hcl/umo/core/UMOMessages_<locale>.properties` files to localize system warning and error messages are not supported. System upgrades and other processes overwrite these files.

Localized object types

To localize user interface labels and text strings for object types, you edit .xml and .properties files for the objects.

If your organization supports multiple locales, the user interface labels and text strings for marketing object types can be translated into the language of each locale. To localize these labels and text strings for your organization, you follow the procedure to rename an object type. You edit the `sysmodules.xml`, `sysmenu.xml`, and `UMOConfigurationMessages_<locale>.properties` or

`UMOMktObjectConfigurationMessages_<locale>.properties` files for each supported locale.

- For standard marketing object types, Unica Plan supplies a properties file for each locale. This file contains a set of properties that define user interface labels and strings for each standard marketing object types. These files are the `<Plan_Home>/messages/com/ibm/umo/ext/UMOConfigurationMessages_<locale>.properties` files.
- The first time that you create a custom marketing object type, Unica Plan creates another properties file for each locale. This file contains properties that define labels and strings for the custom marketing object type. Each time that you add a custom marketing object type, the system adds a set of properties for it to these files. These files are the `<Plan_Home>/messages/com/ibm/umo/ext/UMOMktObjectConfigurationMessages_<locale>.properties` files.

In these files, the `list` parameters define labels and text strings for the page that displays when users select the menu item for the object type. The `ui` parameters define labels and text strings for the pages that display data for a single instance of that object type.

For example, for the project marketing object type, this parameter defines the label for the "Owner" column on the project list page:

```
projectlist.columnList.PROJECT_OWNER.header
```

In the English language properties file, the value for this parameter follows:

```
projectlist.columnList.PROJECT_OWNER.header=Owner
```

In the German language properties file it follows:

```
projectlist.columnList.PROJECT_OWNER.header=Besitzer
```

For more information, see Renaming marketing object types (on page 39).

Localized format and symbol settings

The *<Plan_Home>*`/conf/`*<locale>*`/format_symbols.xml` file stores values to localize the display of dates, currency, months, and days of the week for each supported locale.

Note: HCL discourages editing the `format_symbols.xml` files.

If you do edit the `format_symbols.xml` file, follow these guidelines:

- In Windows™, you must save the file in UTF-8 format, rather than the Windows™ default of ANSI.
- If you edit the date/time settings, you must supply the same date format value for both `<date-format>` and `<date-time-format>`. Otherwise, you will receive errors while working with Unica Plan.

Multi-locale support for templates

Templates in Unica Plan have features to support organizations that use multiple locales. You can use the properties file to localize the template. You can also localize some aspects of templates through the Unica Plan user interface.

When you add a program or project template in Unica Plan, the system saves a `properties` file for the template. The file is saved under the Unica Plan home folder as follows:

<Plan_Home>`\templates\db\properties`
`\`*<template_id>*`_`*<default_locale>*`.properties`

Where *<template_id>* is the internal identifier that is assigned to the template, and the *<default_locale>* is specified by the **Settings > Configuration > Plan > defaultLocale** property.

The `properties` files for program and project templates stores values for the following fields.

- **Properties** tab: **Name**, **Description**, **Default Name**. In the program or project instance that you create from the template, you can localize default labels on the Summary tab.
- **Attributes** tab: Standard attributes. In the program or project instance, these standard attributes are labels on the Summary tab.
- **Tabs** tab: **Display Name**. In the program or project instance, the **Display Name** customizes the titles of forms on the **Summary** tab or more custom tabs.
- **Attachments** tab: **Name**. In the program or project instance, **Name** localizes the names of the folders, which organize attachments on the Attachments tab.
- **Custom Links** tab: **Display Name** and **Description**. In the program or project instance, these fields describe custom links, which display on the Summary tab or new custom tabs.

For example, the following is an en_us version of a `properties` file.

```
attachment_folder.display_name.folder1=Reference Attachments
template.default_name.tradeshow=Tradeshow
template.display_name.tradeshow=Tradeshow Template
custom_link.display_name.new=New Custom Link
tab.display_name.contact=Contact Info
template.description.tradeshow=Use this template for requests/projects
to prepare for tradeshow attendance.
tab.display_name.tradeshowsummary=Tradeshow Attributes
attachment_folder.display_name.folder2=Project Deliverable(s)
```

You can localize these values for as many other locales as your organization supports.

The `properties` file for the default locale is used in the following situations.

- The locale is supported but there is no corresponding `properties` file for it.
- The locale is not supported.

Note: Localizing a template does not localize any forms added to provide custom tabs or sections on the Summary tab. You must localize forms separately. For more information, see Localizing forms *(on page 253)*.

Localizing templates through the properties file

Export and edit the properties file to localize field values on templates.

You can only localize the following field names by editing the `properties` file.

- **Tabs** tab: **Display Name**. In the program or project instance, the **Display Name** customizes the titles of forms on the **Summary** tab or more custom tabs.
- **Attachments** tab: **Name**. In the program or project instance, **Name** localizes the names of the folders which organize attachments on the Attachments tab.
- **Custom Links** tab: **Display Name** and **Description**. In the program or project instance, these fields describe custom links, which display on the Summary tab or new custom tabs.

Note: You can customize some other field values directly in the Unica Plan user interface. For more information, see Globalizing standard attributes *(on page 251)*.

1. Make a copy of the `properties` file.
2. Change the name of the new file to `<template_id>_<other_supported_locale>.properties`.
3. Edit the file to translate the value for each field and save the file.

Globalizing standard attributes

You can customize the attributes on plan, program, and project templates for your organization and locales. You can globalize standard attributes that display on the plan, program, or project Summary tab by editing the Attributes tab.

On the Attributes tab, you can also determine whether the attributes are Required, Standard, or Hidden to streamline the template. You can add a custom help-tip, which displays when the user hovers the mouse over the attribute. The help-tip might tell users what information the field gathers when they make a new instance. These features determine how the standard attributes display on the Summary tab when users create a program or project from the template.

1. Edit a plan template. Create or edit a program or project template. Open the Attributes tab.
2. To change the label of the attribute in the default locale, click the edit icon in the **Label** column. Type the new label in the text box.
3. To add a help-tip, click the edit icon in the **Help-tip** column. Type the help-tip in the text box.
4. To set the attribute to Required, Standard, or Hidden, click the edit icon in the **Display type** column. Select Required, Standard, or Hidden from the drop-down menu.

   Note: On the plan template, the following attributes are always set to Required: Name, Plan Code, and Program Areas. On the program template, the Name and Program Code are always set to Required. On the project template, the Name and Project Code are always set to Required.

5. To globalize the attribute label and help-tip, click the edit icon in the **Language** column.

   Note:

   The **Language** column appears only if your organization supports multiple locales.

   Each locale your organization supports displays in the **Modify details for name** window. If you do not see a specific language, that locale is not supported.

a. Enter translated text for each locale.
b. Click **Save changes**.

6. Click **Save changes**.

Localizing forms

If your organization supports multiple languages, you can make forms available in multiple languages so users can work with forms in their own languages. You localize a form by exporting it and then creating a translated properties file for each locale that you want to support.

When you export a form, Unica Plan creates a compressed file that contains a form properties file for your locale. You can create a copy of the properties file for each locale your organization supports. Then, you can translate the group names, field names, descriptions, and help tips into the appropriate language for each locale. If the form uses lookup tables and the database contains localized versions of the lookup tables, you can edit the properties file to reference the correct lookup tables.

Properties file names

Properties file names must be in the following format.

`<form_name>_<locale>.properties`

where `<form_name>` is the name of the form, and `<locale>` is a locale code. The following locale codes are recognized.

Code	Language
de_DE	German
en_GB	English (Great Britain)
en_US	English (United States)
es_ES	Spanish
fr_FR	French

Code	Language
it_IT	Italian
ja_JP	Japanese
ko_KR	Korean
pt_BR	Portuguese
ru_RU	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Example properties file

```
columngroup.group1.header=group1
columngroup.group1.description=first group
columngroup.offer.header=offer
columngroup.offer.description=second group
columngroup.offer2.header=offer
columngroup.offer2.description=third group
column.business_unit_id.label=Business Unit
column.business_unit_id.message= Business Unit is a mandatory field
column.business_unit_id.helptip= Business Unit is used for
column.init_type_id.label= Initiative Type
column.init_type_id.message= Initiative Type is a mandatory field
column.offer_codes.label=Offer Code(s)
column.effective_date.label=Effective Date
column.drop_date.label=Drop Date
column.business_unit_id.lookuptable=lkup_business_unit
tvccolumngroup.group1.header=group1
tvccolumngroup.group1.description=group1 description
tvccolumngroup.group1.helptip=group1 helptip
tvccolumn.tvc_not_used_ref_1.label=Single Marketing Object
```

Localizing forms by editing

You can also localize forms by having users in the locale open the form and manually edit the names and descriptions. When the user saves the form, Unica Plan saves the translations that the user entered. Then, the translated form displays for other users with that default locale. However, this process takes more time and is efficient only when the number of forms and the number of supported locales are small.

How Unica Plan determines the properties file to use

When a user displays a form, Unica Plan uses the properties file for the first locale in this list for which a properties file exists.

1. User's locale
2. Default locale for the system
3. Locale in which the form was created

Localizing forms

You can localize forms by exporting the form and modifying the properties file.

If the form uses lookup tables and you want to provide localized versions of those lookup tables, create the localized tables before you localize the form. You need the names of the localized tables.

You can only export forms that are published.

1. Export the form.
2. Extract the properties file from the compressed archive form file.
3. Create a copy of the properties file for each locale your organization supports.
4. Open each properties file in a text editor and translate the display text into the appropriate language for that file. If the form uses localized versions of the lookup tables, replace the lookup table names with the names of the localized lookup tables in the properties file.
5. Add the new properties files to the form compressed archive file.
6. Import the form into Unica Plan.

The new version of the form displays indented below the published version that you exported. The **Publish** icon is now available.

7. Publish the form to replace the previously published version with the version you imported.

   The localized text from the properties files is uploaded to the database and is available to all users.

Localizing metrics

You can export the properties file and then modify it to localize metrics.

To generate a properties file for translation, click **Export Properties File** on the **Metrics Templates** page. You download a compressed file that contains the properties file for your locale. The file name is in the format `metric-definition_<locale>.properties`.

The display name keys and description keys for metrics are saved in the metrics-related tables for plans, programs, and projects. To distinguish keys from real values, use the key prefix, which is `$_$`.

While Unica Plan is running, the system replaces metrics keys with values from the metrics properties file.

The following is an example of a properties file.

```
$_$.metric.AVFee.display=Audio Visual Fee ($)
$_$.metric.AVFee.description=Audio Visual Fee
$_$.metric-group.BoothExpenses.display=Booth Expenses
$_$.metric-dimension.metricValue0.display=Actual
$_$.metric-template.CampaignProject.display=Campaign Project
$_$.metric-template.CampaignProject.description=Metrics for
     Campaign Project Template
```

Note: Alternatively, you can work with users who have different locales that are set as preferences. A user from each locale can change the **Display Name** and

Description for the metrics. This method is efficient if your organization supports only a few locales.

Importing a metrics properties file

After you translate the `metric-definition_<locale>.properties` file, you upload the file for the new locale.

1. Click **Import Metrics Template** on the **Metrics Templates** page.
2. Select the **Properties file** check box.
3. Click **Browse** to select the properties file.
4. Click **Continue**.

Localizing lists

When you save a list, the system generates a properties file for the appropriate list. The file name is `<list_category>_<defaultLocale>.properties`.

For example, if you edit the list of business areas, and your default locale is en_US, the system generates the following file:

`BUSINESS_AREAS_en_US.properties`

The file is saved to the directory specified in the **managedListDir** property: click **Settings > Configuration > Unica Plan > umoConfiguration > attachmentFolders**. A code from underlying table is a key, and name from underlying table is a value.

Translate the list and create a properties file for each Unica Plan supported locale.

You can localize user roles, as well. The localized roles for a project template are based on the localized Roles list.

Note: You must use the default locale to create and update list definitions.

Chapter 14. Security

You configure security policies that grant users permission to access specific objects and features through assigned access roles.

Every installation of Unica Plan requires Unica Platform. You access the features of both applications through an integrated user interface. To set up security for Unica Plan, you use features of both Unica Platform and Unica Plan. Before you begin, see the Unica Platform Administrator's Guide for information about security features in Unica Platform.

You create and manage users and user groups with Unica Platform. The permissions that are granted to an access role are determined by a security policy. Any user who is not assigned any roles is governed by the default security policy, Global.

Unica Plan provides security through several layers of access roles, and access roles can be assigned to users in different ways. For example, administrators assign default, or baseline, access roles to users, and then the project leaders who create projects further specify which users can participate and with what roles.

Global security policy

The Global security policy acts as the default security policy for the system. The term "global" does not indicate that every user has full access to everything, but rather that this security policy is associated globally with every user by default.

You can create more security policies that extend the Global security policy, but the Global security policy is always in effect, regardless of any other security policies you create.

The Global security policy:

- Applies to anyone who logs in to Unica Plan
- Cannot be disabled
- Supersedes all other policies. When the system determines access rights for a user, the role that the user has in the Global security policy is always considered.

- Contains permission settings for the default roles, Plan Administrator and Plan User. The permission settings for these roles are used as the fallback or default for any user who does not currently qualify for a role in another security policy, object role, or project role.

Roles

In every organization, the people who use an application have different responsibilities and complete different tasks.

To support secure collaborative work in Unica Plan, you grant or block access to system features by defining roles and assigning them to users.

When Unica Platform is installed, it supplies the Global security policy. The Global security policy has a set of predefined roles for each application in the Unica suite, including Unica Plan. You can alter these predefined roles as needed and then assign them to users, or set up your own security policies and roles.

To control access to specific objects and projects in Unica Plan, you also use object access roles and project roles.

Default security policy roles

Default roles, Plan Administrator and Plan User are included in the default Global security policy.

These roles provide application access levels for Unica Plan in Unica Platform and they are always in effect.

- When, using Unica Platform, you add a user to a user group that has the PlanAdminRole level of access, that user is assigned the Plan Administrator user role in Unica Plan. By default, users with this role have access to all administrative and configuration settings.
- When, using Unica Platform, you add a user to a user group with the PlanUserRole level of access, that user is assigned the Plan User user role in Unica Plan. By default, users with this role are granted few permissions.

These role assignments display on the **Settings > Unica Plan Settings > User Permissions** page.

You cannot override these role assignments on the **User Permissions** page, and you cannot remove these roles from the Global security policy. To change the default security policy role that is assigned to a user, you must change the user group assignment by using Unica Platform.

Any changes that are made to users in Unica Platform are reflected in Unica Plan only after you synchronize the user database tables. User synchronization occurs automatically at regular intervals, as specified by **Settings > Configuration > Plan > umoConfiguration > usermanagerSyncTime**. Or, to manually synchronize the user database tables, you or another administrator can start **Settings > Plan Settings > Synchronize Users**.

Custom security policy roles

Custom security policy roles are exactly that: roles that you set up within the security policies that you add.

These roles are meant to control access to Unica Plan functionality that is based on a job title or function that characterizes user responsibilities in your organization as a whole.

For example, marketing managers need full access to all plans, programs, and projects. Individual marketers only need to view plans and programs, but need to both view and create projects. To accommodate these different functional needs, you add a security policy with different roles for Marketing Managers and Marketers.

The roles that you add to security policies, plus the default roles (Plan Administrator and Plan User), are the roles that you assign to individual users on the **User Permissions** page.

Object access roles

When users do their work in Unica Plan, the system assigns them the appropriate object access role.

Each object type in Unica Plan has a set of object access roles. For projects and approvals, object access roles are also called "access levels."

For example, a person who creates a project has the project owner access level, and the users who are assigned to project roles have the project participant access level. Object roles for projects and approvals are also called access levels because project participants with the appropriate permissions can assign both object access roles and project roles to participants.

Every object type has an owner, by default the person who created it. Many object types also have extra roles, as described in the following table.

Table 67. Object types and associated roles

Object type	Object access roles / Access levels
Plan	Plan Owner, Plan Participant
Program	Program Owner, Program Participant
Project	Project Owner, Project Participant, Project Requester
Request	Request Recipient, Request Owner
Asset	Asset Owner
Account	Account Owner
Approval	Approval Owner, Approval Approver
Invoice	Invoice Owner
Teams	Teams Manager, Teams Member
Custom marketing object type	<*marketing object type*> Owner For example, if you have a marketing object type that is named Creatives, its object role name is Creatives Owner.

Object access roles support general system processing. As a result, they are not accessible through custom security policies.

Project roles

Project roles represent the job functions of the people who participate in a project or make project requests.

Template developers construct a list of appropriate roles on the Project Roles tab of each project template. Then, when you configure security policies, you select the template and the system displays the template's project roles along with the other access roles. That way you can configure different permissions for different templates based on project roles in addition to system, object, and security roles.

Additionally, within individual templates, you can configure different permissions for different tabs (both custom and default tabs). For example, perhaps participants from one project role shouldn't be allowed to edit the workflow; they should just be able to view it. Or perhaps only the user who participates in a project role named "accounting" for an individual project should be allowed to edit the **Budget** tab no matter what their other access roles might be.

Custom security for template tabs based on project roles can be disabled, if necessary. To do so, click **Settings > Configuration > Plan > umoConfiguration** and set the **customAccessLevelEnabled** configuration property to `false`.

Security policies and permissions

Security policies are made up of roles that reflect job titles or responsibilities.

Each role in a security policy is a set of permissions: rules that grant or block user access to features and options in Plan Unica Plan. For example, you can configure security policies that ensure the following.

- Managers have access to all the projects in their business units.
- Users' access to projects is based on both their business unit and their job function.
- Some users can create projects while others must use requests to start a project.

Any object that is created in Unica Plan: plan, program, project, request, and so on, is governed by a security policy. The security policy that is assigned to a new object is determined by the template that was used to create it.

Permissions

You manage permissions for a security policy on the **Permissions for** <security policy> page.

See Creating a security policy (on page 267).

To set the permissions for a role in a security policy, you use a set of tables. There is a different table for each marketing object type (plan, program, etc.). The table columns present all of the roles in the security policy, and the rows represent features or options, grouped by the tab on which they are accessed.

When you configure security for projects or requests, you also select a template. When you do, extra columns display for each project role that is defined in the template.

The permissions that you configure for security policies control access throughout all the functions of Unica Plan.

For example, the results of searches are constrained by access rights: if a user does not have access to the Workflow tab of a specific project, tasks from that project do not display in the **All Tasks** search. Additionally, if a user does not have the ability to add attachments to a project, that user does not receive the alerts that are triggered when another participant performs an attachment task.

The security policy in effect at a given time for a given object (project, plan, or program, for example) depends on the security policy that is specified by the object's template. For example, when template developers create project templates, they specify the security policy on the template Summary tab. Then, when projects are created from that template, access to those projects is determined by the security policy that is specified in the template.

Security policy permission settings

To configure permissions on the **Permissions for** <security policy> page, you click the table cell at the intersection of a role and a feature or option.

Clicking in the cell toggles through the following settings:

Table 68. Permission settings on the Permissions for <*security policy*> page

Symbol	Name	Description
	Granted	Grants users in the role access to the function.
	Blocked	For system and security policy roles only, denies users in the role access to the function. (You cannot block functions by project or object roles.) A permission block supersedes any other setting. For example, a user's project role grants access to the Budget tab for the projects that are created from a specific template, but the security policy role blocks access to the tab. That user cannot access the Budget tab.
	Inherited	Not explicitly defined. An explicit permission setting for the project and object roles that are assigned to the user is inherited. If none of the other roles has the permission set, the setting in the appropriate default Global security policy role (Plan Administrator or Plan User) is inherited. If none of the roles grants the user permission, the function is blocked.
N/A	Not accessible	For project and object roles, indicates that the permission setting cannot be changed.

If a user qualifies for more than one role, the permissions are cumulative. For example, if a user's security role grants different permissions than the role in the current project, that user has all of the permissions that are granted by both of the roles.

Planning security policies

Before you begin to add and configure security policies, determine the security needs of your organization and then plan your security strategy.

First, determine how many security policy roles and project roles you need. Then, determine whether you need to create a security policy with different roles, or whether you can simply modify the roles that are supplied by the Global security policy to meet your needs.

- If all of the business units in your organization follow the same rules, or if you can implement the appropriate differences in access through a combination of project and security policy roles, it makes sense to implement one security policy: a modified Global security policy. You can add as many roles as necessary to the Global security policy.
- If there are numerous functional groups in your organization that require different types of access, leave the Global security policy in its default state and add a security policy with one or more roles for each functional group.
- At any time, a user can have an object role, a project role, and a security policy role. It is best practice to assign a user one security policy role only, from a single security policy. Therefore, if you have users who multi-task in such a way that they need more than one security policy role in addition to their project and object roles, it is recommended that you create more security policies and assign that user one role from each of the appropriate security policies.

As a best practice, try to implement the smallest number of security policies possible. Within a single security policy, you can configure different permissions for each marketing object type. You can also configure different permissions for each of your project and request templates. Additionally, for each project template you can configure different security role and project role permissions for each tab (custom as well as standard) separately for projects and project requests.

When you set up permissions for the roles, the individual permission settings are granular. For example, if you want users in a particular role to be able to edit the Summary tab of a project, you must grant that role both Edit and View permissions. If you forget to grant the View permission, users in that role do not see the Summary tab, so their permission to

edit it is useless. In another example, it would not make sense to grant permission to post messages without also granting permission to read them.

Note: When a project or a request is added as a child to a Program, it possesses the same security policy as the parent object, i.e., the Program.

Configure security policies

After you determined your security strategy, you configure and create the appropriate security policies and roles and then specify to which users the people with those roles can assign project roles and access levels.

Editing the Global security policy

If you can implement the security strategy for your organization with one security policy, use the Global security policy as your single policy.

In most cases, you do not change the default permissions for the supplied Plan Administrator and Plan User system roles. Instead, you add new security policy roles to implement your security goals.

1. Select **Settings > Plan Settings > Security Policy Settings > Global**.
   The **Properties for** <*security policy*> page displays.
2. Add a security role:
   a. Click **Add Another Role**.
   b. Enter a **Name** and **Description** for the role.

   Repeat these steps for each role you add.
3. Click **Save and Edit Permissions**.
   The **Permissions for** <*security policy*> page displays. This page contains a table interface so that you can grant or block access to features for each role.
4. To configure permissions for objects other than projects and requests, from the **Access to** list select the marketing object type. Use the check boxes to configure permission settings for each security role. See Security policy permission settings *(on page 263)*.

You repeat this procedure for each object type in the **Access to** list.

5. To configure project permissions, complete these steps.
    a. From the **Access to** list, select the Project object type.
    b. In the General section, configure the **Add Projects** and **View Project in the List** permissions for each object and security role.
    c. Select a project template. The security policy displays a column for each project role specified for this template on the Project Roles tab in the Team Members section. An access control section displays for each tab in the template.
    d. Configure permissions for each tab in the template, including any custom tabs, for the project, object, and security roles.
    e. Repeat steps c) and d) for each project template.
6. To configure request permissions, complete these steps.
    a. From the **Access to** list, select the Request object type.
    b. Configure the **Add Requests** and **View Request in the List** permissions for each object and security role.
    c. Select a project template. The security policy displays a column for each project role specified for the template on the Project Roles tab in the Project Request Recipient section. An access control section displays for each tab in the template.
    d. Configure permissions for each tab in the template, including any custom tabs, for the project, object, and security roles. When configuring requests, remember that the permissions you set for the **Request Recipient** object role must match the permissions for at least one of the roles in the recipient project.
    e. Repeat steps c) and d) for each project template for which you want to configure custom permissions for requests.
7. For custom marketing object types, be sure to configure permissions for each template.
8. Click **Save and finish**.

Creating a security policy

If you must use more than one security policy to implement the security setup for your organization, leave the Global security policy in its default state and complete the following steps.

1. Select **Settings > Plan Settings > Security Policy Settings > Add Security Policy**.
2. On the **Properties for** <*security policy name*> page, enter a **Policy Name** and **Description**. The name must be unique.
3. In the Roles section, enter a **Name** and **Description** for the first two roles that are planned for this security policy. If you need more than two roles, click **Add Another Role**.
4. Click **Save and Edit Permissions**.
   The **Permissions for** <*security policy*> page displays. This page contains a table interface so that you can grant or block access to features for each role.

   You use the **Access to** list to select each object type, and then configure permissions for the roles in the security policy. See Editing the Global security policy (on page 266).
5. Click **Save and finish**.

   To disable the security policy at any time, go to the **Security Policy Settings** page and click **Disable**. Disabling a security policy means that users cannot select it in any subsequent projects, requests, or approvals that users create and you can no longer assign users to the security policy.

Configuring the user visibility option for roles

To restrict the list of users that display in the **Select Team Members** or **Select Member Access Levels** dialog for users with a given security policy role, you configure the user visibility feature for that role.

When users create programs, plans, projects, and so on, they specify which users or teams are participants and, for projects, which users or teams are assigned the project roles. By default, there are no restrictions on which users or teams can be added as participants or assigned a project role.

1. Select **Settings > Plan Settings > Security Policy Settings**.
2. On the **Security Policy Settings** page, scroll to the appropriate security policy and click the name of the role. The **Role:** *<role name>* page displays for user visibility selection.
3. From the list of user groups and teams on the left, select the groups or teams that you want users with this role to see in the list and then click **>>** to move them to the list on the right.
   When users who have the selected role add participants or assign project roles, they select from the set of users who are in a group in the list on the right.

   Note: When the selection box on the right is empty (the default case), there are no restrictions and users who have this role see all groups and teams when they add participants or assign project roles.

4. Click **Save changes**. The **Security Policy Settings** page displays.
5. Repeat steps 2 through 4 for each role that you want to configure.

Assigning security policy roles

When you finish adding roles to security policies, you can assign the roles to the appropriate users.

If a user is not explicitly assigned a security policy role, the system uses the Global security policy to determine the permissions for that user.

You assign security policy roles directly to individual users on the **User Permissions** page.

1. Select **Settings > Plan Settings**.
   The **Administrative Settings** page displays.
2. Click **User Permissions**.
   The **User Permissions** page displays.
3. Expand the user group that the user belongs to and click to select the user.
   The **Properties for** *<user name>* page displays.
4. Click in the **Available Roles** list box to expand the security policy with the security role you want to assign this user.
5. Select the role, then click **>>** to move it to the **Selected Roles** list.

It is a best practice to assign one role from one security policy to each user.

6. Click **Save changes**.

   The role displays under the Assigned Roles column on the **User Permissions** page.
7. Repeat steps 3 - 6 as needed for other users.

Controlling access to templates

You can control which users can create and view templates.

As described in the sections about security policies, you use permissions in security policies to control the following kinds of access:

- Security policy roles control which users can create new projects, plans, programs, and so on.
- Security policy roles control which users are allowed to view and interact with items that other users create, even if they cannot create those items themselves.
- Project and object roles control which tabs users have access to when they create projects.

You also use security policy roles to specify which project or request templates a user can select when you create an item.

When a template developer creates a template, the Summary tab includes one or more security policy fields. The value that is specified in the security policy field determines which users can access the template; if you do not have a role that grants you the ability to create objects of that type in the security policy that is assigned to the template, the template does not appear in your template list when you create an object of that type.

Access controls for projects and requests

Depending on how your organization manages projects, you might configure security policies so that only certain users can create projects while other users must create requests for projects that other users approve or reject.

When this is the case, it is also possible that one group of users creates the projects from the requests and then a different group of users works with the projects created by the first group.

To support this business case, project templates have two security policy settings:

- The "view" policy specifies which users can select the template when they create projects or requests for projects. The template developers can specify one or more view policies for each project template.
- The "use" policy specifies who can access a project after a project is created from a request.

The "use" policy can be determined in one of two ways:

- The template developer specifies the security policy in the template's Summary tab.
- The template developer configures the template so that the user creating the project or project request can specify the "use" policy.

The way a "use" policy is determined is referred to as the "security policy use model." When the use model is set to `Template`, the template developer specifies the "use" policy. When the use model is set to `User`, the person creating the project request from the template selects a security policy from the list that is visible.

Example security configuration for project requests

This example describes an organization, XYZ Corporation, that has a Plan team, a strategic marketing team, and some miscellaneous marketers.

Users create two types of projects and requests: trade shows and strategic accounts.

- Trade show projects: junior marketers create requests for trade show projects. The requests can be submitted to anyone in the marketing organization, and the resulting projects can be worked on by anyone as well.
- Strategic account projects: junior marketers also create requests for strategic account projects, but they can supply information on the Summary tab only. Additionally, requests can be submitted to members of the strategic marketing team only, and the strategic marketing team is the only team that participates in the projects.

Security policies

The system administrators at XYZ corporation configured two security policies.

- **Marketing Ops**, for members of the marketing operations team. Security for the templates is configured as follows in this policy:
    - Trade show template: all project roles have access to all tabs.
    - Strategic Accounts template: the Request Owner role has access to the Summary tab only.
- **Strategic Marketers**, for senior members of the marketing staff. Security for the templates is configured as follows:
    - Trade show template: all project roles have access to all tabs.
    - Strategic Accounts template: all project roles have access to all tabs.

Template permissions

To set up the workflow described above, the template developers configured the templates with the following permissions.

- The Summary tab of the **Tradeshow** template has the following security policy settings:
  - **Security Policy Use Model**: User. The user who creates the request specifies the security policy to apply to the request.
  - **View Security policies**: Marketing Ops., Strategic Marketers. (Any user can select the Tradeshow template.)
  - **Use Security Policy**: Blank. When the use model is set to User, the Use security policy field is disabled. When users create projects or requests from this template, they must specify the security policy.
- The Summary tab of the **Strategic Accounts** template has the following security policy settings:
  - **Security Policy Use Model**: Template. The template developer sets the value in the Use security policy field.
  - **View Security policies**: Marketing Ops., Strategic Marketers. (Any user can select the Strategic Accounts template.)
  - **Use Security Policy**: Strategic Marketers. This means that the user who creates the request cannot specify the security policy for it. Instead, the requests created from this template are assigned the Strategic Marketers security policy. Then, only the senior marketers who have security roles assigned from the Strategic Marketers security policy can access the project requests and the projects created from those requests.

Example usage

Consider the following users assigned to the following security polices:

- Strategic Accounts security policy: Mary Manager, Strategic Sam
- Marketing Ops.: Junior Jim, Sophomore Sally

Users create requests and projects as follows:

Table 69. Example project requests

Project or request	Work steps
Trade show project	Junior Jim creates a trade show request and submits the request to Strategic Sam. Strategic Sam approves the request and sets Vendor Vinny as the owner of the project.
Strategic accounts project:	Junior Jim creates a Strategic Accounts request, SA01, providing information for the only tab that he has access to, the Summary tab. The request is automatically assigned the Strategic Accounts security policy and Jim cannot change it.

Summary

- Anyone can create a request for a trade show or a strategic accounts project.
- Anyone can be the recipient of a trade show request, and anyone can be assigned to a trade show project.
- Only users with roles from the Strategic Accounts security policy can work on strategic accounts projects.

Chapter 15. Setting up alerts

Alerts are notifications of important changes that users need to know about, or reminders about actions that they need to take.

Examples of alerts include notifications that a project status changed, or that an approval response is due.

Unica Plan users receive their alert notifications in these ways:

- In Unica Plan: the **Alerts** icon ( ) appears at the top of every page, and includes a counter that increments for each alert received. Users click this icon to view notification messages in the **Alerts** dialog.

  Tip: You configure how frequently the system updates the alert count. See Changing the refresh interval for the alerts count *(on page 287)*.

- By email: users who are set up with a valid email address receive notifications as messages in their email applications.

To set up alerts, you define values for configuration properties. You configure how often the system checks for the event-triggered alerts that track past changes separately from the alarm-type alerts that remind users of upcoming events. See Event-triggered alerts (on page 276) and Reminders (on page 276).

You also specify default alert subscriptions. For each Unica Plan object type, you select which team roles, by default, receive alerts and for what types of changes and reminders. See Set default alert subscriptions (on page 277).

Tip: Users can override the default notification subscriptions for specific instances of the Unica Plan objects. For information about how users work with alerts, see the *Unica Plan User's Guide*.

Optionally, you can customize the text of the messages that are sent for alerts. Different messages can be defined for each of the Unica Plan object types that trigger alerts. See Alerts Settings page (on page 283).

Event-triggered alerts

Event-triggered alerts are notifications that Unica Plan sends in response to system events.

They track changes that already occurred. For example, when someone creates an approval, the system sends the user who is its approver an alert.

Unica Plan can send notifications for event-triggered alerts almost immediately after they occur. Each time an event that triggers an alert occurs (such as a new approval request), Unica Plan adds it to a queue. At a specified interval, Unica Plan checks the queue and sends out notifications for all of the waiting events. By default, this interval, or polling period, is every 5 seconds. You can change this default frequency by setting the **notifyEventMonitorPollPeriod** property: Click **Settings > Configuration > Plan > umoConfiguration > Notifications**.

For information about all of the event-related configuration properties, see the Unica Plan Installation Guide.

Reminders

Any alert that is not triggered by the occurrence of a single, specific event is called a reminder or alarm.

Typically, this type of alert involves the relationship of an object (such as a task or project) to time, or to another object.

You configure how many days in advance to send different types of reminder notifications. For example, the **notifyProjectAlarmMonitorScheduledStartCondition** property sets how many days before the start date of a project to send start notifications to users, and the **notifyProjectAlarmMonitorTaskOverdueCondition** property sets how many days after the end date of a task to notify users that a task did not finish. Alternatively, you can indicate that you do not want any notifications to be sent for a reminder type by supplying `-1`.

You also configure how often the system iterates through the object instances to determine whether reminder notifications need to be sent. For example, you configure Unica Plan to send a notification 1 day before every task is scheduled to start. To determine whether any of these notifications need to be sent, Unica Plan:

1. Determines the current date.
2. Compares it to the scheduled start date of every workflow task.
3. Finds the difference.
4. Sends a reminder for every task that has a difference of 1 day or less.

You can configure different polling periods for different types of objects. Because this process uses more system resources than the check for event-triggered notifications, the default polling period for each one is 60 seconds. You configure these properties to change the default polling period:

- **notifyProjectAlarmMonitorPollPeriod**
- **notifyApprovalAlarmMonitorPollPeriod**
- **notifyAssetAlarmMonitorPollPeriod**

To change the default values for these properties, click **Settings > Configuration > Unica Plan > umoConfiguration > Notifications**.

For information about all of the configuration properties, see the Unica Plan Installation Guide.

Set default alert subscriptions

Administrators can define a set of default alert subscriptions. For each object type, you select the team members who receive different types of alerts by subscribing or unsubscribing their object access roles.

You set up default alert subscriptions for the following Unica Plan objects and access roles.

Table 70. Object access roles that you can subscribe to receive alerts

Object type	Object access roles
Project	• Project Owner • Project Participant • Project Requester
Request	• Request Recipient • Request Owner
Program	• Program Owner • Program Participant
Approval	• Approval Owner • Approval Approver
Asset	Asset Owner
Invoice	Invoice Owner
Account	Account Owner
Plan	• Plan Owner • Plan Participant
Checklists	• Checklist Owner • Checklist Editor • Checklist Viewer • Todo Owner • Todo Assignee
Custom marketing object types	Owner
Offer *	Offer Owner

* Applies only to installations with Unica Plan-Unica Campaign integration enabled.

To set up default alert subscriptions, click **Settings > Unica Plan Settings > Default Alert Subscriptions**. For each object type, a list of different alerts displays. To subscribe team members with a corresponding access role, you select a check box. For example, to send notifications when a new member is added to a project, you could select the check box for project owners and project participants, and clear the check box for project requesters.

Notes® on setting default alert subscriptions

When you work with default alert subscriptions, note the following.

- You are creating a default setting. Users (who have appropriate security permissions) can change these default settings for each object instance.
- When you modify the default alert subscriptions, existing object instances are not affected. Your changes affect only object instances that are created after your modifications.

User overrides of default alert subscriptions

For each object instance, users can review the alerts that they are subscribed to by default and change them. To do so, they open a program, project, or other object instance, click **Communicate** ( ), and select **Subscribe to Alerts**.

Setting default alert subscriptions

You can set default alert subscriptions by selecting the appropriate check boxes from the **Default Alert Subscriptions** page.

1. Select **Settings > Plan Settings > Default Alert Subscriptions**.

   The **Default Alert Subscriptions** page lists all of the alerts, which are grouped by object type. See Default Alert Subscription page (on page 280).
2. To subscribe the team members who have one of the object access roles, select the appropriate check boxes.

   To unsubscribe team members, clear the appropriate check boxes.
3. Click **Save changes**.

Default Alert Subscription page

The **Default Alert Subscription** page is divided into sections for each marketing object type: projects, requests, programs, approvals, and so on.

There is also a section for each custom marketing object type that is defined on your system. A list of the alerts that can be sent displays in each section. The alerts are grouped by type.

- **Change Tracking**: This section lists the event-triggered alerts. For example, the **A new project is created from a request** alert appears in the **Change Tracking** subsection for projects.
- **Reminders**: This section lists the reminder alarms that are triggered by a comparison between the current date and the scheduled or target dates for object instances. For example, **A project is 3 days late** appears in the **Reminders** subsection for projects.

For information about configuring these different types of alerts, see Event-triggered alerts (on page 276) and Reminders (on page 276).

Reminders and workflow task dates

When you set up the default alert subscriptions, for project workflow tasks you can send notifications that are based on the target dates, the forecast dates, or both.

On the **Default Alert Subscriptions** page, the section for **Projects | Reminders** includes these alerts that are based on project target dates:

- A workflow task is targeted to start within *n* day(s)
- A workflow task is targeted to end within *n* day(s)
- A workflow milestone is targeted to end within *n* day(s)
- A workflow task is overdue according to targeted dates (alerted at most for *n* day(s))
- A workflow task is late according to targeted dates (alerted at most for *n* day(s))

These alerts are based on forecast dates:

- A workflow task is forecasted to start within *n* day(s)
- A workflow task is forecasted to end within *n* day(s)
- A workflow milestone is forecasted to end within *n* day(s)
- A workflow task is overdue according to forecasted dates (alerted at most for *n* day(s))
- A workflow task is late according to forecasted dates (alerted at most for *n* day(s))

You configure notification properties to set the number of days for each type of alert. See Reminders (on page 276).

Customizing notification messages for alerts

Administrators can customize the subject, message text, header, and footer for alerts. You can also localize messages: you select the locale for every message that you customize.

You can customize alert messages for the entire system or for selected templates. For example, you can customize the message that is sent out whenever a program is started. Alternatively, you can customize the message for one specific program template only, such as the Tradeshow sample template.

To customize the default message to send for an alert, you use the **Customize Alerts** option on the Administrative Settings page. To customize the message for a specific template, you use the Customize Alerts tab, which is available for every type of template. See Customizing an alert notification message (on page 282).

Tip: If you configure custom permissions for the tabs of a project template, the system filters the alerts appropriately. For example, if some project participants do not have access to the **Attachments** tab, the system does not send them alerts about attachments.

When you construct custom messages, you can include variables and links to system tabs in addition to text. When you add these items, they display as system-defined tags. When the system sends the alert notification, it replaces any tags in the message with values appropriate for that marketing object type and object instance.

Adding variables

You can insert placeholder variables for standard planning object data or system data related to an event. You can insert variables into the subject, body, header, and footer.

Tip: On the **Alerts Settings** page, see the **Attributes** list for the variables you can add.

Adding links to standard tabs

You can include a link to any standard tab presented for the selected marketing object type. When you include a link to a tab, the email message contains a link directly to the specified tab for the object instance. For example, when a project is started, the notification message can include a link to its Workflow tab.

In templates, you can add links to tabs to the subject, body, header, and footer. At the system level, you can add tab links only in the subject and body (not in the header and footer).

Customizing an alert notification message

You can customize an alert notification message by adding or modifying a template and clicking its Customize Alerts tab.

1. Determine whether to customize an alert system-wide, or for a particular template only.
    - To customize an alert system-wide, click **Settings > Plan Settings > Customize Alerts**.
    - To customize an alert for a particular template, add or edit the template then click its Customize Alerts tab.

    The **Customize Alerts** page or tab displays.
2. Select a **Locale**.

Note: If your system supports multiple locales, repeat this procedure to supply locale-specific custom text for each alert you customize.

3. Select a **Planning Object**.

   If you are configuring a template, this field may be disabled. For example, if you are working with a project template, **Project** is selected in this field and you cannot select anything else.
4. Select an **Alert Event**.
5. Click **Get Alert Details**.
   The values currently defined for the message subject and body display. For alerts that are the result of a staffing change or that otherwise involve a specific user, additional fields that you can use to store optional, personal text for the affected user display.
6. Enter or edit the text of the **Subject** and **Message Body** for the alert.
7. Optionally, specify attributes and links to tabs for the subject, body, header, or footer.

   Note the following.

   - To include variables and links to tabs, use the list on the right side of the page for the section you are editing. See Customizing notification messages for alerts *(on page 281)*.
   - If you are using detailed task alerts, you can only customize the alert header and footer for workflow task alerts.
   - If you are customizing alerts system-wide, you cannot add links to tabs in the header and footer.
8. In the **Customize Alerts** section, click **Save changes**.
9. Optionally, click **Get Header and Footer**.
   The values currently defined for the message header and footer display.
10. Enter text and include variables and links to tabs for the **Header** and **Footer** of the alert.
11. In the **Customize Header Footer** section, click **Save changes**.

Alerts Settings page

The **Alerts Settings** page (or tab) contains sections where you customize a message subject and body and its header and footer.

For more information, see Customizing an alert notification message (on page 282).

Customize Alerts section

The top area of the page contains the controls to customize the message subject and body.

Table 71. Customize Alerts section

Field	Description
Locale	Select the locale for the custom text. If your system supports multiple languages or locales, then for each alert you customize, be sure to supply text for all of the supported locales.
Planning Object	Select the object for which the custom text applies. **Note:** Does not apply to templates.
Alert Event	Select the type of alert that triggers this custom message text.
Get Alert Details	Click to fill the **Subject** and **Message Body** fields with the current or default text for this alert. For certain alert events, updates the fields to display 2 subject fields and 2 message fields. See General and personalized message fields *(on page 285)*.
Subject	Contains the subject for the alert. Enter or replace text, attributes, and links to tabs to change the subject.
Message Body	Contains the message text for the alert. Enter or replace text, attributes, and links to system tabs to change the message.
Attributes / Tabs	Select either **Attributes** or **Tabs** to add variables or links to system tabs to the subject or message text. See Customizing notification messages for alerts *(on page 281)*
<<	To include an attribute or tab link, click its name then click **<<** to move it into the **Subject** or **Message Body** field.

Customize Header Footer section

The bottom area of the page contains the controls to customize the message header and footer.

Table 72. Customize Header Footer section

Field	Description
Locale	Select the locale for the custom text. If your system supports multiple languages or locales, then for each alert you customize, be sure to supply text for all of the supported locales.
Get Header & Footer	Click to fill the **Header** and **Footer** fields with the current or default text for this alert.
Header	Contains the heading text for the alert. Enter or replace the text to change it.
Footer	Contains the footer text for the alert. Enter or replace text, attributes, and links to system tabs to change.
Attributes / Tabs	Select either **Attributes** or **Tabs** to add variables or links to system tabs to the subject or message text. See Customizing notification messages for alerts *(on page 281)*
<<	To include an attribute or tab link, click its name then click **<<** to move it into the **Header** or **Footer** field.

General and personalized message fields

For certain types of alerts, you can send one notification message to the user who is most directly affected by the change and a different message to other team members.

For example, when a project starts, the system sends out the same alert to all affected users. However, when a workflow task is assigned to a specific user, the system sends out a message (referred to as the personal message) to the assignee, and a general message to all of the other affected users.

After you select the **Alert Event** on the **Alerts Settings** page or tab, click **Get Alert Details**. If the selected event can accommodate both a personal message and a general message, the page refreshes to show two fields for the message subject, personal and general, and two for the message body, personal and general. See Alerts Settings page (on page 283).

Example of a custom alert message

In this example, you construct a custom message for the alerts that notify users that a new project is created from a request.

You provide the custom message for the user who needs to take action.

1. On the **Alerts Settings** page:
    - **Locale**: English (or choose your locale)
    - **Planning Object**: Request
    - **Alert Event**: A project request is submitted
2. Click **Get Alert Details**. The page refreshes to show the current general and personal messages.
3. Select and delete the default text for the **Subject (Personal)** and the **Message Body (Personal)**.
4. Use the lists of **Attributes** and **Tabs** to construct the following subject and message.

    Subject (Personal)

    <attribute>Logged in User</attribute> would like you to approve the request, <attribute>Request Name With Code/</attribute>

    Message Body (Personal)

    Hello <attribute>Recipient</attribute>,

    Your approval is needed to start this project. This request was created on <attribute>Created Date</attribute>.

    You can approve the project here: <tab link="Summary">Summary tab for the project</tab>
5. Click **Save changes** in the Customize Alerts section.

Assume that Connie Contact submits a request to Mary Manager. Mary receives the following alert:

```
Connie Contact would like you to approve the request, "July Magazines
 (TRS100)"

Hello Mary Manager,
Your approval is needed to start this project. This request was created on
June 15, 2008.
You can approve the project here: Summary tab for the project.
```

Changing the refresh interval for the alerts count

You can change the refresh interval for the alerts count by updating the **alertCountRefreshPeriodInSeconds** setting.

When users log in to Unica Plan, the **Alerts** icon ( ) shows the number of unread notifications that they have.

By default, the system updates this count every 3 minutes (180 seconds). You can change this default frequency by setting the **alertCountRefreshPeriodInSeconds** property: Click **Settings > Configuration > Plan > umoConfiguration > Notifications**.

Note: Decreasing the refresh interval can have performance implications in a multi-user environment.

For information about configuration parameters, see the Unica Plan Installation Guide.

How Unica Plan determines the alert sender

When Unica Plan sends an alert notification through email, the sender email address is the first valid address of the following.

1. The user who initiated the action that triggered the alert.
2. The owner of the object instance.
3. The value of the **notifyDefaultSenderEmailAddress** property. Click **Settings > Configuration > Plan > umoConfiguration > Email**.

If none of these email addresses is valid, Unica Plan issues a warning (to the log file) and does not send an email alert.

Setting up the collaboration tool

From version 12.1.2 onwards, you can use Plan to configure a collaboration tool with Unica Plan. In version 12.1.2, Unical Plan provides out-of-the-box settings that you can use to collaborate using Slack. Slack is not packaged with the Unica installer. You have to install Slack separately.

As an administrator, you must configure settings for Plan project owners to set up a collaboration tool and also configure settings for team members to be part of the collaboration. To achieve this you must configure integrations that will allow a project owner to set up the collaboration tool and members to be part of it. For more details, see the Unica Content Integration Installation and Configuration Guide.

Chapter 16. List definitions

The user interface of Unica Plan presents several list box controls that you can configure to provide a customized set of options.

Examples include a list of the roles, or job functions, held by people in your organization, and a list of the types of days observed as non-working days, such as national and company holidays. As an administrator, you populate these lists by defining options for them.

For information about other ways you can customize Unica Plan, see Unica Plan interface customization options (on page 39).

Customizable lists

You can customize the lists in Unica Plan by providing your own site-specific options.

The lists that you can customize, and where users encounter them in the user interface, follow.

Table 73. Customizable lists

List type	Description	Location
Business Areas	An area of business to which a plan can belong. Used primarily to allocate budgeted funds.	When users create or edit a plan, they can specify a **Business Area** for it. Users access this list in the Plan Summary section.
Program Areas	A unit that groups one or more programs in a plan. Program areas are especially useful when users allocate funds to a related group of programs that are linked to the plan.	When users create or edit a plan, they can specify a **Program Area** for it. Users access this list in the Plan Summary section.
Cost Categories	A category that helps to define budget or invoice line item costs.	When users create or edit a line item for an invoice or budget,

Table 73. Customizable lists (continued)

List type	Description	Location
		they can select a **Cost Category** for it.
Vendors	The name of the business to which an invoice line item was purchased. For more information, see Vendor column for budgets (on page 125).	When users create or edit an invoice, they must select a **Vendor Name** for it. Users access this list in the Invoice Summary section.
Roles	Job functions or skill sets. Roles make it easier to assign people to tasks in a project. **Note:** These roles are functional, and are not the same as the security roles that determine access to areas of the Unica Plan interface.	Users identify the roles that are needed to complete the project, and assign team members to those roles, on the People tab. Then, on the Workflow tab, users can assign roles or team members to tasks.
Non-working Date Types	Categories for non-work time. For example, national holidays, corporate off-sites, and company holidays. For more information, see System-wide non-work dates (on page 14).	When administrators enter Non-working Business Days, they identify the **Type** for each date. Then, when users add or edit project workflow tasks, they can specify a **Schedule Through** setting. Task schedules can include only business days, business days and weekends, business

Table 73. Customizable lists (continued)

List type	Description	Location
		days and these non-working date types, or all dates.
Workflow Milestone Types	Milestones that can be included in a project workflow.	When users add or edit a project workflow task, they can identify it as a project milestone by selecting its **Milestone Type**.
Approval Deny Reasons	Reasons for rejecting items that are received for approval. Applies only to installations that require a reason when an approval is denied. For more information, see Setting approval deny reasons (on page 149).	When users respond to an approval, they can choose to deny approval. To do so, they must specify a **Deny reason**.
Coremetrics® Clients	Client IDs and names from an IBM Digital Recommendations implementation. Applies only to installations that integrate Unica Plan and Unica Campaign, and that also enable optional offer integration. For more information, see the Unica Plan and Unica Campaign Integration Guide.	Administrators can configure offer templates so that users can select category IDs and names from a Digital Recommendations system, rather than entering them manually. To configure the template, administrators supply the URL for Digital Recommendations and one or more valid **Client IDs**.

Adding options to a list

As an administrator, you can populate customizable lists with values.

For a description of the lists you can customize, see Customizable lists (on page 289).

1. Click **Settings > Plan Settings**.

   The **Administrative Settings** page displays.
2. Click **List Definitions**.

   The **List Definitions** page displays.
3. Click the name of the list that you want to populate with options.

   The **List Properties** page displays.
4. Supply values for the fields on this page. For detail on each field, see List properties *(on page 292)*.
5. Click **Save changes** to save your changes.

List properties

To define the entries for a customizable list, you supply values on the **List Properties** page and then click **Save Changes**.

For more information, see Adding options to a list (on page 291).

Field	Description
List Name	Displays the name of the selected list.
Description	Enter a description of the list. Unica Plan provides a default description that you can edit or replace.
Display	Specify how you want options for this list to display and be ordered: by code number followed by name, or by name followed by code number.
Storage Location	Displays the name of the database table that stores options for this list.
New or Selected Item	To add an option, enter a unique identifying code and display name, then click **Accept**. Unica Plan adds the option to the `List Items` field.

Field	Description
	To edit an existing option, click the option in the **List Items** field to select it. These fields display the code and name for you to edit as needed. Click **Accept** when your changes are complete.
List Items	This field displays all options that are defined to populate the list.

This page also provides controls to enable, disable, and remove options in the **List Items** field. For more information, see Enabling, disabling, or removing a list option (on page 293).

Control	Description
Disable	Keeps the selected option in the database but does not display it in the Unica Plan user interface. The **List Items** field displays disabled options in gray font. If you disable an option that has already been selected for objects, the option remains associated with those objects. However, users cannot select the option for any other objects.
Enable	Restores a disabled option to full operation in the user interface. By default, new options are enabled.
Remove	Deletes the selected option from the Unica Plan user interface and the **List Items** field. You cannot remove an option that has already been selected for objects.

Enabling, disabling, or removing a list option

As the needs of your organization change, you can remove options that are no longer in use. When you remove an option from a list, it is permanently deleted. If you want to add the option again, you must readd it.

After you add an option to a customizable list, it displays in the user interface as a value that users can choose.

You can also disable an option, which keeps it available for possible future use while preventing it from appearing on a list. When the option is needed again, you enable it.

1. Follow the procedure that is described in Adding options to a list *(on page 291)*.
2. From the **List Items** field, select a value to enable, disable, or remove.
   You can also use Ctrl+click or Shift+click to select multiple items.
3. Click **Disable**, **Enable**, or **Remove**.
4. Click **Save changes**.

Chapter 17. Implementing project health rules

To help project owners and participants track the overall status of their projects, you configure the system to calculate project health.

Unica Plan supplies a set of key performance indicators (KPIs) for projects. To objectively determine whether a project is healthy or in a warning or critical state, you select KPIs and supply threshold values for them in health rules. To use different criteria to determine the health of different types of projects, you can associate each of the rules you create with one or more project templates. The system runs batch jobs to automatically apply the correct rule to each project and update health status indicators in the user interface.

To implement project health rules for your organization, you:

1. Assess the default health rule, and configure custom health rules as needed.
2. Assign health rules to project templates.
3. Schedule the start time for the daily batch job, and how frequently during each day additional batch jobs run. For more information, see the **Unica Plan > umoConfiguration > Scheduler >daily** and **intraDay** configuration properties in the Unica Plan Installation Guide.
4. Optionally, customize the labels and color indicators that display for the health statuses.

You can export health rules that are defined for one Unica Plan system and import them into another. For more information, see Exporting and importing metadata (on page 302).

When implementation is complete, project owners and participants can use the following methods to monitor health status:

- Subscribe to email notifications generated by the daily batch job.
- Add a Project Health Status portlet to the dashboard.
- Review indicators in the Project Health column of the project list page.
- Open the Project Health tab for an individual project.
- Run the monthly and trend project health reports.

For more information about these features, see the Unica Plan User's Guide.

Key performance indicators

Health rules include key performance indicators (KPIs) that evaluate project performance against an objective threshold.

For example, you consider a project healthy if it has a budget overrun of less than 5%, but you consider a project with a 10% budget overrun to be in a critical state.

Table 74. KPIs for project health

KPI	Description
% Milestone Overdue	For all In Progress and Completed project milestones, compares the total delay time to the total planned time.
% Budget Overrun	For the project, compares the total estimated budget less actual expenses to the total estimated budget. This KPI requires the Financial Management module.
% Time Overdue	For all In Progress and Completed project tasks, compares the total delay time to the total planned time.
% Tasks Delayed	For the project, compares the number of overdue tasks to the total number of incomplete tasks.
% Milestone Tasks Delayed	For the project, compares the number of overdue milestones to the total number of incomplete milestones.

How the system determines overdue tasks and milestones

For several of the KPIs, the system must determine whether tasks or milestones are overdue. To make this determination, Unica Plan compares the current date (today) to the Forecast/Actual date and the Target end date for the task or milestone.

Note:

- The system uses the date, time, and timezone of the server as the current date.
- The system uses the timestamp of the daily batch job that calculates health status as "today".
- The system always categorizes tasks with the Skipped status as Not Overdue.

Table 75. Results of date comparisons

State / Condition	All other states	Finished
Today > F/A end > Target end	Overdue	Overdue
Today > Target end > F/A end	Overdue	Not Overdue
Target end > Today > F/A end	Not Overdue	Not Overdue
Target end > F/A end > Today	Not Overdue	Not Overdue
F/A end > Target end > Today	Not Overdue	Overdue
F/A end = Blank AND Target end > Today	Not Overdue	Overdue*
F/A end > Today > Target end	Overdue	Overdue
F/A end = Blank AND Today > Target end	Overdue	Not Overdue

* When the task status changes to Finished, the system supplies the Actual end date with a timestamp. During the next health status calculation, the system reassesses the task as Not Overdue.

Default health rule

Before you create your own project health rules, assess the default health rule supplied by the system.

You can use it as a model when you design your own customized rules for determining project health. You can also assign it to project templates as needed.

Table 76. Conditions in the default rule

IF	THEN
% Tasks Delayed = 0 AND % Time Overdue = 0 AND % Budget Overrun <= 0	Healthy
ELSE IF % Tasks Delayed <= 5% AND % Time Overdue <= 5% AND % Budget Overrun <= 5%	Warning
ELSE IF % Tasks Delayed > 5% AND % Time Overdue > 5% AND % Budget Overrun > 5%	Critical
OTHERWISE	Unknown

When you design a custom rule for your organization, note:

- The default health rule includes a condition that resolves to each of the possible project statuses: Healthy, Warning, and Critical. Include a condition for each status in your custom rules.
- The operators and values for the conditions cover a complete range of possible values, without any unassigned values. The system assigns the Unknown status to any value that is not included.

For example, you change the first condition to set the % Budget Overrun KPI to < 0 instead of <= 0. As a result, the system determines the status of under budget projects, with no task or time delays, to be Unknown instead of Healthy.

- The sequence of the conditions affects what status the system calculates for a project.

  For example, you change the sequence of the conditions in this rule so that the condition that resolves to Warning comes first. As a result, this rule never assigns the Healthy status to any projects.

Configuring project health status rules

You can add, edit, and delete project health status rules, and assign rules to project templates.

Before you change an existing rule, verify that no other administrators are working with that rule.

1. Click **Settings > Plan Settings > Health rules**.
   A page lists the health status rules.
2. To add a rule, click **Add health rule**.
   The system adds lines with fields for the rule name and a description.
3. Supply a name and description, then click **Build Rule** ( ).
   A dialog opens for you to enter the if-then statements that determine project health.
4. Supply the first **IF** clause:
   a. To select the KPI, click **Select an Attribute**.
   b. Click to select an operator.
   c. Enter a threshold value.
   d. Click **Add**. The clause displays in the center of the dialog.
   e. To include more than one KPI in the clause, select **And** or **Or** and then repeat these steps.
5. Supply the **THEN** clause: select the health status to assign to projects that meet the condition.
6. Click **Save Compound Condition**. The condition displays at the top of the dialog.

7. Repeat these steps so that the rule includes conditions for the Healthy, Warning, and Critical health statuses.
8. To review the rule as a series of IF... THEN statements, click **Preview**. To change the rule after your review, click **Conditions**.
9. The system applies the conditions in the order that is shown at the top of the dialog. If necessary, change the sequence by using the check boxes and the **Up** and **Down** controls.
10. Click **Save and finish**.
11. Click **Save Rule** ( ) for the rule.

To implement the rule, you assign it to one or more project templates. See Assigning a rule to a project template (on page 300).

To edit a rule, verify that no other administrators are working with that rule. Then, return to the Health rules page and click **Edit rule** ( ) for the rule.

You can delete a rule that is not assigned to any project templates. Verify that no other administrators are working with that rule. Then, on the Health rules page, under **Actions** click for the rule.

Assigning a rule to a project template

You use this procedure to change the assignment of a rule to one or more project templates.

1. Click **Settings > Plan Settings > Health rules**.
   A page lists the health status rules.
2. For the rule you want to assign, click **Edit rule** ( ).

Note: Before you edit a project health status rule, verify that no other administrators are working with that rule. Object locking does not apply to project health status rules.

3. Click **Modify Template Association** ( ).
   A dialog opens with a list of enabled project templates on the left, and a list of templates the rule is assigned to on the right.
4. Use **Select >>** and **Remove <<** to move selected templates from one list to the other.
   Use Ctrl+click and Shift+click to select multiple templates.
5. Click **Save & Close**.
   The Health rules page lists the project templates with the rule.
6. Click **Save Rule** ( ) for the rule.

   Note: Until you click Save Rule, your selections are not saved. To cancel your edits, under **Actions** click for the rule.

You can also assign a rule to an individual project template. Edit the template, and on the Properties tab select the **Project Health Status Rule**.

Customizing labels and colors

You can customize the label and color indicators for each possible health status.

1. Click **Settings > Plan Settings > Health status**.
   A page displays the current label and color indicator for each of the possible health statuses.
2. To enter a different display name for a status, click in its **Label** field.

   Note: Enter the label for your default locale. Localization is not supported.

3. To use a different color indicator for a status, click **Choose Color** then click one of the colors.
4. Click **Save > Save and Exit**.

Chapter 18. Exporting and importing metadata

You can transfer data structures (metadata) between Unica Plan systems using the export and import features.

To transfer metadata efficiently from one Unica Plan system to another, you export the metadata from one instance and import it into another instance.

For example, you create templates on a test server and then test and refine them to assure they meet the needs of your organization. When you are ready to deploy the templates for general use, you use the export feature on the test server to create a compressed archive file, then use the import feature on the production server to load the file and install the templates.

Unica Plan offers options to package and migrate metadata in bulk. You can migrate the following types of metadata in bulk.

- Security policies and related user roles
- Project heath status rules
- Teams
- Marketing object types
- Templates

When you migrate metadata from one Unica Plan system to another, be aware that:

- Both the source and target systems must be running the same version of Unica Plan.
- The source and target systems can be running under different operating systems.
- The source and target systems can be using different types of database servers.

Exporting metadata

When you export metadata in bulk, Unica Plan iterates over your entire database for all items of the selected type. As a result, the export process can be time-consuming.

The export process results in a compressed archive file containing one or more xml files. For some types of metadata, a number of additional files are also exported, including properties files or SQL scripts. All exported data uses UTF-8 encoding to preserve locale-specific data.

Exporting metadata in bulk

You can export metadata in bulk by completing the **Export Template** dialog.

1. From the **Settings** menu, select **Plan Settings**.
2. Click **Data Migration**.
3. Next to **Templates**, **Teams**, **Security Policies**, or **Marketing Object Types**, click **Export**.
4. If you are exporting templates, the **Export Template** dialog opens.
    a. Select the types of templates to include in the export. By default, all template types are selected.
    b. Specify the **Database Type** of the system that will receive the template metadata through an import operation. The selected database type determines the format of SQL script files generated during the export.
    c. Click **Export**.
5. If you are exporting any other type of metadata, or after you complete the **Export Template** dialog, the standard File Download dialog opens. To proceed with the export, click **Open** or **Save**.

Results of exporting templates

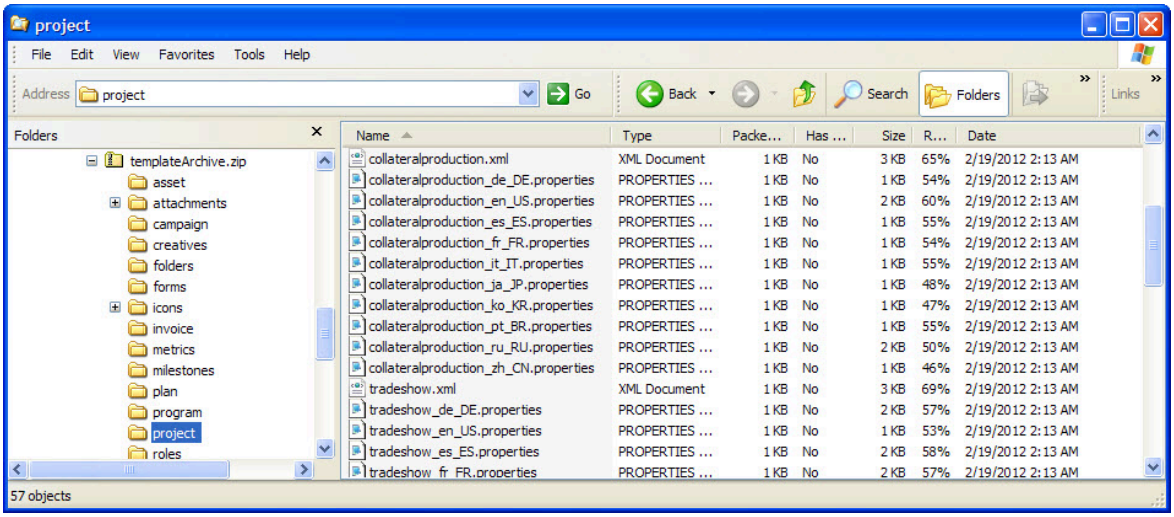
When you export templates, you choose one or more different types of templates to export, such as plan, project, or creative templates.

You also specify the database type of the target system receiving the template metadata.

For the selected template types, Unica Plan produces a compressed archive file that contains:

- An xml file named `<type>_templates.xml`, with metadata for every template of that type.
- An individual directory for each template type that contains a `<name>.xml` file and a set of localized properties files for every template of that type.

  For example:



- Directories for any items associated with templates of the selected types, such as `roles` and `milestones`, with localized properties files for those items.
- An `attachments` directory (if any included item has an attachment), with a subdirectory named for each item containing those attachments.
- A `forms` directory containing form definition files in XML format and separate SQL scripts for the selected database type. These scripts give you control over how you update the target database to work with the new templates when you import them: you can drop all tables and then create new tables for template data, or run only the create or insert scripts to add new columns and tables without deleting existing tables and data.

Table 77. Generated script files

File	Description
`create.sql`	Adds columns to existing tables, and creates new tables needed for the templates.
`cre-atelkup.sql`	Adds columns to existing lookup tables, and creates new lookup tables needed for the templates.
`drop.sql`	Deletes existing tables used by the templates. If you do not mind possibly deleting data, you run this script before `create.sql` to ensure that the database is set up correctly.
`droplkup.sql`	Deletes existing lookup tables used by the templates. If you do not mind possibly deleting data, you run this script before `cre-atelkup.sql` to ensure that the database is set up correctly.
`in-sertlkup.sql`	Inserts data into the lookup tables. This script makes it possible to save complete lookup tables (schema plus data) with the template archive.

Results of exporting project health rules

When you export metadata for project health rules, the compressed archive file contains a single `health_status_rule.xml` file.

The file includes the conditions and results, name, and description of every rule on your system.

Results of exporting teams

When you export metadata for teams, the compressed archive file contains an individual xml file for every team defined on your system.

Each file is named `team<ID>.xml`.

Results of exporting security policies

A compressed archive file contains information from the export of metadata for security polices.

When you export metadata for security policies, the compressed archive file contains:

- An individual xml file for every security policy defined on your system, named `securityPolicy<ID>.xml`. All related user roles are included in this file.
- The `securityPolicyFunctions.xml` file, which contains the list of permissions referred to in each `securityPolicy<ID>.xml` file.

Results of exporting marketing object types

When you export metadata for marketing object types, the compressed archive file contains an individual subdirectory for every supported locale, such as `en_US` for English (US).

Each subdirectory contains the following xml files:

- `compTypes.xml` contains the metadata of every marketing object type.
- `globalstates.xml` contains the metadata for every status defined on your system.
- `mo_<name>_state.xml` is provided for each marketing object type. These files contain the metadata for the transitions defined between statuses.

Importing metadata

To import metadata into a Unica Plan system, you select a previously exported archive file.

The import process validates the archive and its component files: the source and target systems must have the same version of Unica Plan installed, and all files must be correctly formatted.

Unica Plan data structures are interrelated. Duplicating the data structures of a source system is an iterative process: you import archive files, perform manual configuration, and potentially reimport archive files.

When you have metadata of more than one type to import, this sequence is the most efficient.

1. Marketing object types

   Import marketing object types first to assure that when you import templates, any templates for those marketing object types are also imported.
2. Project health rules
3. Templates
   - The global security policy for the target system is assigned to any templates that have a security policy that is not present on the target system.
   - Templates with defined rules that involve users who are not present in the target system are not imported.
   - Templates with defined rules that involve teams that are not present in the target system are imported, but do not function.

   Continue by importing security policies and teams, and then either import the template archive again or make updates to the target system as needed.
4. Security policies

   User visibility definitions for teams and user groups that are not present on the target system are not imported. Continue by importing teams, and then either import the security policies again or make updates to the target system as needed.
5. Teams

   After you import teams, review and update rules and user visibility definitions as needed.

After you import a metadata archive, be sure to review the results in the target system. Configure users, rules, teams, user groups, security policies, and templates as needed to integrate new structures into the target system.

Importing template metadata

The following procedure applies when you import an archive of template metadata.

1. From the **Settings** menu, select **Plan Settings**.
2. Click **Data Migration**.
3. Next to **Templates**, click **Import**.

The **Import Template** dialog opens.

4. Click **Browse** to select a previously exported archive file.
5. Select the **Template Types** to import. By default, all template types are selected.
6. In the **Update Database** section, select optional database scripts to run during the import process.
    - Drop Tables
    - Create/Update Tables
    - Drop Lookup Tables
    - Create/Update Lookup Tables

    If you do not select any of the scripts, the import process overwrites data values for the template, but does not update corresponding database tables.

    Note: Selecting all of these actions fully imports the selected templates and associated files. However, if the archive file contains templates that exist on the target system and you drop tables, the import process deletes all data for all objects created using the pre-existing templates.

    For example, if you use import to update the metadata of a campaign project template and you drop tables, you lose all data in the TCS® in any project created using that template.

    If you are concerned about overwriting data, you can examine the SQL script files in the template archive and create the necessary tables and columns manually.
7. Click **Continue**.
   A summary page lists the templates to import and any warnings about current template files to be overwritten.
8. Click **Save**.

Note: The Unica Plan installation does not install the optional example templates supplied with the product. To use the example templates, follow this procedure to import them. The example template archive files are in the `\tools\admin\sample_templates` folder under your Unica Plan installation. A different archive file is supplied for each supported database type. For example, use

`sample_templatesDB2` if you are using a DB2® database. For details about the example templates, see List of example templates *(on page 85)*.

Results of importing templates

You can import campaign project templates that you created before your upgrade; however, the templates are disabled.

Users cannot use them to create projects. Template import fails if the template archive contains any of the following items:

- A campaign project template created with the same name as a campaign project template already in the system.
- A campaign project template with the same name as a non-campaign project template already in the system (and vice versa).
- A TCS® form with the same name as a non-TCS® form already in the system (or vice versa).
- A shared attribute with the same name but a different data type as a shared attribute already in the system.

Shared attributes used in forms in the template archive are created as shared attributes in the target system.

Importing metadata

The following procedure applies when you import an archive of metadata for project health rules, teams, security policies, or marketing object types.

1. From the **Settings** menu, select **Plan Settings**.
2. Click **Data Migration**.
3. Next to **Project Health Status Rules**, **Teams**, **Security Policies**, or **Marketing Object Types**, click **Import**.
   The **Import** dialog opens.
4. Click **Browse** to select a previously exported archive file.
5. Click **Continue**.

A two-part summary of the items in the archive displays:

- Items to be created: that is, items with a unique identifier that does not exist on the target system.
- Items to be overwritten: that is, items with a unique identifier that exists on the target system.

6. Select the items that you want to import.
7. Click **Save**.

Results of importing project health rules

When you import metadata for project health rules, the import process compares the unique identifier of each rule to the rules that exist on the target system.

- If a project health rule does not exist on the target system, the import process creates it using the archive.
- For project health rules that do exist on the target system, the import process overwrites values for the rules and conditions, name, and description.

Note: Associations established on the source system between project health rules and project templates are exported and imported with template metadata.

Results of importing teams

When you import metadata for teams, the import process compares the unique identifier of each selected team to the teams that exist on the target system.

If a team does not exist on the target system, the import process creates it using the archive and then:

- Checks security policy data in the archive against security policies that exist on the target system. Team-related data for security policies that exist are copied from the archive. If none of the security policies in the archive exist on the target system, the new team is assigned the default Global security policy.

- Checks member data in the archive for members that exist on the target system. If members exist on the target system and meet definitions in the routing model, they are added to team. If any resulting members or managers do not meet the definitions of the routing model, the team is not imported.

For teams that do exist on the target system, the import process:

- Overwrites values for the team, including the description, status, and skill sets.
- Checks associated security policy data for security policies that exist on the target system. Team-related data for security policies that exist are copied from the archive. If none of the security policies in the archive exist on the target system, the team is assigned the default Global security policy.
- Updates the routing model with data from the archive.
- Checks member data in the archive for members that exist on the target system. Members associated with the team in the archive are added to the team if they exist on the target system. Members associated with the team on the target system are removed if they are not allocated to any task, approval, or project request and they are not present in the archive. If any resulting members or managers do not meet the definitions of the routing model, the team is not imported.

For all teams added or updated on the target system, the import process also:

- Copies alerts and notification settings for each team to the target system.
- Adds an entry to the Analysis tab for the team to record the update.

Results of importing security policies

When you import security policies, the import process compares the unique identifier of each selected policy to the policies that exist on the target system.

If a security policy does not exist on the target system, the import process creates it with all of the object and template level permission settings in the archive. For security policies that do exist on the target system, the import process overwrites all values for the policy, removes all user roles and associations, and then copies all user roles from the archive to the target system.

For all security policies added or updated on the target system, the import process also:

- Copies object-level function settings to the target system.
- Checks associated template-level security policy settings in the archive against the templates on the target system, and copies template-level security policy settings for any project or component templates that exist.
- Checks user data in the archive for users that exist on the target system, and copies user role assignments for users that exist.
- Checks group data in the archive for groups that exist on the target system, and copies group visibility for roles for groups that exist.
- Checks team data in the archive for teams that exist on the target system, and copies team visibility for roles for teams that exist.

Results of importing marketing object types

When you import marketing object types, the import process verifies that the archive includes files for the default locale of the target system.

For each selected marketing object type, the import process then verifies that:

- The marketing object type does not exist on the target system.
- The marketing object type in the archive passes all restrictions present on the target system.
- All states and state transitions for the marketing object type in the archive are present for the default locale on the target system.

For marketing object types that satisfy these conditions, the import process creates the marketing object type and copies all of its associated data. Alerts and notification settings for each new marketing object type are created on the target system also.

The import process does not upgrade marketing objects that do exist on the target system.

Chapter 19. Setting up libraries for digital assets

Through the optional Digital Asset Management module, Unica Plan provides centralized management, secure storage, and web-based access for digital assets.

In Unica Plan, users add digital assets to libraries, which have the following characteristics.

- Unica Plan administrators can create libraries for digital assets. Users cannot store assets in Unica Plan until at least one library is set up.
- Unica Plan users access libraries and add assets to the libraries by selecting **Operations > Assets**.
- Asset libraries themselves cannot be organized into a hierarchical structure. All of the libraries that you set up have the same organizational level.
- Users can add folders to libraries to organize the assets that they add.

For information about how users work with digital assets after you set up libraries, see the Unica Plan User's Guide.

Creating a library

Administrators create the libraries where users store digital assets.

You cannot delete a library, but you can change its state to disabled.

1. Select **Settings > Plan Settings**.
2. In the **Root-Level Object Definitions** section, click **Asset Library Definitions**.
   The **All Libraries** page displays.
3. Click **Add a Library**.
   The **New Library** page displays.
4. Enter a **Name** and **Description** for the library.
   For example, name your library "Brand Materials" and describe it as the location where all images and documents that are related to brand management are stored.
5. Select the **Security Policy** that you want the library to use.

- The library can be accessed only by users with the selected security policy.
- All of the folders and assets added to the library have the same access control rules, which are specified by this security policy.
- If you want a specific set of documents to have different access control permissions, create a different library for those documents.

6. Click **Save Changes.**

   The **All Libraries** page lists your library in the enabled state. Users can immediately begin adding assets to the library by clicking **Operations > Assets**.

To edit a library, click its name on the **All Libraries** page.

Enabled and disabled libraries

You can enable and disable libraries. By default, libraries are in the enabled state when you create them.

To disable a library, click **Settings > Plan Settings > Asset Library Definitions**, and click **Disable**. To enable it again, click the **Enable** link.

You can change the state of a library to disabled at any time; however, when you disable a library:

- Users cannot access disabled libraries or the assets they contain. They cannot view or edit assets in a disabled library, or browse to the library to add attachments to projects or to add approval items.
- When a user attaches new files to a project or an approval, disabled libraries do not display in the selection list.
- If there is a link in an alert or email message to an asset that is in a disabled library, the link does not work.
- However, if an asset in a disabled library is also an attachment to a project or approval, users can access the asset from that project or approval.
- Only administrators can change a disabled library.

Chapter 20. Setting up accounts

An account in Unica Plan represents a specific corporate General Ledger (GL) account that is established by a finance department to track and control expenditures and cash flows. Accounts are a feature of the optional Financial Management module.

Key capabilities for accounts include the following. You can:

- Define a hierarchy of accounts and subaccounts.
- Fund, or allocate money to, accounts for each month in the current fiscal year and for the next two years.
- Track the estimated (forecast) and actual withdrawals from those accounts, by time period.

Unica Plan breaks down accounts into top-level accounts and subaccounts. You can create a hierarchy by adding subaccounts to top-level accounts and to other subaccounts: Subaccounts display under the top-level and parent accounts on the **Settings > Unica Plan Settings > Account Definitions** page.

Important: Subaccounts belong to their parent accounts for organizational purposes only: subaccount financial information does not roll up to parent accounts. Functionally, top-level and subaccounts are identical.

Account administrators

The account administrators in your organization can be members of the finance or accounting department, responsible for setting up the accounting framework to track marketing budgets and expenditures. Or, account administrators can be members of the marketing department who are responsible for reporting marketing spend details in that framework to the finance/accounting department.

The responsibilities of an Unica Plan account administrator include the following.

- Defining accounts and subaccounts.
- Funding the accounts; that is, updating each account with the budgeted amount for each time period.
- Assigning account owners, who monitor and manage accounts on an ongoing basis.

An account administrator can also define the options that are included in the lists of vendors and cost categories in Unica Plan. A vendor is required when you create an invoice, and a cost category can be selected for each budget or invoice line item. For more information, see List definitions (on page 289).

Note: An account administrator must be set up as a user with the Plan Administrator security policy role in Unica Plan to complete these tasks.

Account owners

In your organization, account owners are typically the mid- to high-level marketing managers who are responsible for managing the budget for a particular business area.

In particular, they are responsible for tracking cash flows and expenditures versus budgets to ensure that their business area is not overdrawn.

The responsibilities of an Unica Plan account owner include:

- Monitoring account levels and status to ensure that they are not forecasted to be overdrawn and that balances remain positive. Account owners can click **Financials > Accounts** to review account information, and use a combination of alerts, views, and reports to monitor account activity.
- Selecting the account for the budget line items of a project or program. On the project or program Budget tab, the **Source Account** column for line items stores this information.
- Selecting the account for invoice line items. On the invoice Summary tab, the **Source Account** column for line items stores this information.
- Communicating and transferring account activity details back to corporate accounting personnel and systems.

To separate accounting functions from marketing functions, give account owners the PlanUserRole in Unica Plan. Users with this role cannot create or fund accounts, but when they are designated as account owners they can view data for their accounts and select them for budgets and invoices.

Creating an account

Unica Plan administrators can add accounts. You can add either a top-level account or a subaccount.

You add subaccounts to an existing account at any level to create an organizational hierarchy. For example, you have a top-level account that funds marketing efforts in the Northeast United States. You can add a subaccount for efforts specifically in New York, and then add subaccounts to the New York account for New York City and for other areas in New York.

1. Select **Settings > Plan Settings**.
2. In the **Root-Level Object Definitions** section, click **Account Definitions**.
   The **Account Definitions** page displays.
3. To add a top-level account, click **Add a Top-Level Account**.

   To add a subaccount to any account, click **Add** for that account.

   The **Account Properties** page displays.
4. Complete the fields in the **Basic Info** section.
   See Account Properties page *(on page 318)*.
5. Optionally, use the **Budget (Next 3 Years)** section to enter budgeted amounts for each month in the next three years.

   Note: Financial information that is entered for a subaccount does not roll up to its parent accounts.

6. Click **Save changes** to save the account.

The account displays in the disabled state on the **Account Definitions** page. See Enabling or disabling an account (on page 321). A subaccount displays in a hierarchy below the top-level account to which it, or its parent, belongs.

To edit an account, click its name on the **Account Definitions** page.

Account Properties page

When an account administrator creates or edits an account, the **Account Properties** page displays. Sections for Basic Info and Budget (Next 3 Years) display.

Basic Info section

The **Basic Info** section contains the following fields.

Field	Description
Account Name	Required. Unica Plan uses this name to identify the account in the user interface. Enter a unique name.
Description	An optional description for the account. This description displays on the **Account Definitions** page.
Team Members	Required. List of owners for the account. By default, the creator of the account is listed as an owner. See Adding or remove account owners (on page 319). Account owners who are not also account administrators can click **Financials > Accounts** to review account information, and select the accounts that they own for line items on program and project Budget tabs and invoices. They cannot access the **Account Properties** page.
Account Number	Required. A unique, alphanumeric identifier for the account. Do not include spaces.
Security Policy	Required. The security policy for the account. Only users in this security policy can access the account.

Field	Description
Source Account	Displays for subaccounts only. The read-only name of the parent account.

Budget (Next 3 Years) section

The **Budget** section provides a grid interface for you to enter the amount budgeted for each month of the current year and the next two years. Unica Plan tallies up entered amounts for each quarter, and totals the budgeted funds for the account per calendar year.

Account owners who click **Financials > Accounts** to review account information see data for the current year by default. When a different year is selected from the **Account Summary** list, a new browser window opens to display account information for that year.

Note: If budget or invoice line items draw from the account in other years, those years also appear on the **Account Summary** list.

Adding or remove account owners

You can add and remove other team members as account owners.

Only the owners of an account can view or edit that account. When you create an account, you are automatically added as an owner.

1. Navigate to the account you want to edit.
2. Click **Add/Remove Members**.
   The Select Member Access Levels dialog opens.
3. To make another team member an account owner:
   a. Select a user in the **Folders** list.
   b. Click **>>** to add the user to the list of **Selected Team Members**.
      Each team member in the **Selected Team Members** list is an owner of the account.
4. To remove an account owner:

a. Select a user in the **Selected Team Members** list.
b. Click **<<** to remove the user.

5. Click **Save changes**.

Enabled and disabled accounts

Accounts can exist in one of two states: enabled or disabled.

For example, you can set up an account for future use and leave it in the disabled state until you are ready to start using it.

When an account administrator creates an account, it is created in the disabled state. It must be enabled before it appears in the user interface for account owners to review and select. Administrators can change the state of any account at any time.

- Enabled accounts can be selected for project and program budget line items. On the project or program Budget tab, the **Source Account** column for line items stores this information.
- Enabled accounts can also be selected for invoice line items. On the invoice Summary tab, the **Source Account** column for line items stores this information.
- The names of disabled accounts display in gray, italic font on the **Settings > Plan Settings > Account Definitions** page. Administrators can edit the information and budget for disabled accounts from this page.
- Disabled accounts cannot be selected for invoice line items or for project or program budget line items.
- An account that has budget or invoice line items that are linked to it remains active for those line items even if the account is disabled. the disabled account cannot be selected for new line items, however.
- Account administrators can add a subaccount to a top-level account that is disabled. When you are ready to use the subaccount, however (at the beginning of a new fiscal year or period, for example), you must enable the top-level account.

See Enabling or disabling an account (on page 321).

Enabling or disabling an account

As an administrator, you can enable or disable an account. Account owners who are not Unica Plan administrators cannot change the state of an account.

1. Select **Settings > Plan Settings**.
2. In the **Root-Level Object Definitions** section, click **Account Definitions**.
   The **Account Definitions** page lists enabled accounts in normal font, and disabled accounts in gray italic font.
3. To enable an account, click **Enable**.

   To disable an account, click **Disable**.

Chapter 21. Advanced topics

This chapter provides information about advanced customization options for the Unica Plan interface.

The following topics are included.

- Populating fields on the Summary tab programmatically
- Writing custom data validation rules

For information about services you can use to integrate Unica Plan with other applications, see the Unica Plan Integration Module guide.

Populating fields programmatically

In Unica Plan, you can create custom fields that are populated programmatically from the values in other fields.

To specify that you want a field to populate programmatically, you add an attribute with an attribute type of External Datasource to a form and specify a program to run. The program can be a web service (located anywhere) or a Java™ program that runs on the same server as Unica Plan.

For example, a field can call a program to generate a job number that is based on values that are entered in fields for business unit and product.

An attribute with the External Datasource attribute type appears in the user interface as a read-only data entry field paired with a **Generate** button. When a user clicks **Generate**, Unica Plan accesses the specified program and displays the result in the field.

To specify the program, you must include the <servicedetails> tag within the <column> tag. The <servicedetails> tag can contain the following tags.

Tag	Description
type	Enter either javaclass or webservice as the type.

Tag	Description
classname	Enter the server-side custom Java™ class in this tag. This custom class must implement the com.unicacorp.common.template.IdGenerate interface. If you specify a value in this tag, the <methodname> tag is optional.
param	This tag has the following attributes. • parameter name • type • valuecolumn You must define all parameters in the same map file, such as projectatts.product_id. You must specify the parameters in the order that the program expects them.
wsdl	Enter the webservice definition file that is located on the Unica Plan server, or the URL to the file in this tag.
methodname	Enter the webservice method name in this tag. If you specify the <classname> tag, this tag is not required.

Examples of programmatically populating fields

There are several examples you can use to programmatically populate field.

This example uses the `<servicedetails>` tag to set up a server-side Java™ class application by implementing the `com.unicacorp.common.template.IdGenerate` interface and passing a product ID.

```
<servicedetails>
     <classname>com.unicacorp.uap.webservice.FormIdGenImpl
     </classname>
     <param name="param1" type="string"
```

```
     valuecolumn="dyn_projectatts.product_id" />
</servicedetails>
```

Similar to the previous example, the following shows how to configure the same behavior, but with a generic Java™ class that does not implement the `com.unicacorp.common.template.IdGenerate` interface.

```
<servicedetails>
     <classname>com.unicacorp.uap.webservice.FormIdGenImpl
     </classname>
     <param name="param1" type="string"
     valuecolumn="dyn_projectatts.product_id" />
     <methodname>getFormId</methodname>
</servicedetails>
```

The following is an example of how you can use the `<servicedetails>` tag to set up a web service application and pass a business unit ID.

```
<servicedetails>
     <wsdl>
          http://rd600:7004/axis/services/Service?wsdl
     </wsdl><!--wsdl>
          C:\\Product\\Plan\\webapp\\conf\\Service.wsdl
     </wsdl -->
<methodname>getFormId</methodname>
<param name="param1" type="string"
 valuecolumn="dyn_projectatts.business_unit_id" />
<param name="param2" type="string"
 valuecolumn="dyn_projectatts.prog_type_id" />
</servicedetails>
```

Server-side ID generation and project attribute validation

You can set up a template to use custom routines to generate ID values automatically, and to validate values entered on the Summary tab of an object.

The ID values that you can set up for automated generation include the identifiers for a project, plan, or program.

To define a custom ID generator, you must write a Java™ class that implements the com.unicacorp.uap.project.helper.PidGenerate interface. Within the template definition, you can then specify your Java™ class name as the value for pidGenClass attribute, and use the pidprefix attribute to include an optional prefix.

In a similar manner, you can also define custom routines to validate attribute values of a project, plan, or program. To define a custom validation routine, you must write a Java™ class that implements the following interface: com.unicacorp.uap.common.template.IdValidate.

Within the project template definition, you can then specify your Java™ class name as the value for validateClass attribute.

Example server-side ID generation

Assume that you have a custom marketing object type for offers. You want its template to generate a custom code for each offer instance that is created. The codes must have the following characteristics:

- The first number starts with 900001.
- The last number possible is 999999.
- Codes must be sequential.

To set up automated ID generation, you use the IdGenerate interface and follow these steps.

1. Create a custom Java™ implementation named `CustomComponentPidGenerateImpl.java`. See Sample Java interface *(on page 326)*.
    - This implementation uses a file, `IDRange.properties`, to hold the minimum and maximum values for the custom IDs.
    - It uses a database table, CUST_GENIDS, to hold the current value of the custom ID for each object type that uses the class.
2. Compile the class. The compiled class is named `CustomComponentPidGenerateImpl.class`.
3. Copy the class file into the following directory:

    `<MarketingOperations_Home>\unwar\WEB-INF\classes\com\unica\uap\component\helper`
4. Create a file named `IDRange.properties`, and add the following text to this file:

    ```
    mktOBJId.min=900001

    mktOBJId.max=999999
    ```
5. Copy this file to the `<Plan_Home>\unwar\WEB-INF` directory.
6. Using your database management program, create a table named CUST_GENIDS with the following columns:
    - ENTITY_NAME; string, length 50
    - ID_VALUE; integer (all in file format)
7. Restart your web server.
8. Create or edit a template for the marketing object type, and open its Properties tab.
9. In the **ID Generation Class** field, use the fully qualified class name or canonical name to specify the custom class, as shown here:

    ```
    CustomComponentPidGenerateImpl
    ```

    When you create the first offer from this template, note that its ID is 900001.

Sample Java™ interface

This section describes IdValidateInterface, IdGenerateCustom, and ID generator.

Interface - IdValidate

```
package com.unicacorp.uap.common.template;
import java.util.HashMap;
/**
This is an interface to be implemented by the end user of a Unica Plan
system for the purpose of validating system generated id values
as per business logic.
Implementations of this Interface are called by the Unica Plan Server.
*/
public interface IdValidate
{
/**
Returns true if the specified attribute values are valid.
*
* @param id - current project or program id.  This will be the
       value if it is new project/program
* @param values - This is a set of name/value pairs, referring to
         a current database connection, the appropriate
         template id and another HashMap that contains
         name/value pairs, corresponding to the fields and
         values on the screen.
* @return true       - if it is valid; otherwise returns false or throws
       exception.
* @throws      com.unicacorp.uap.user.IdValidateException
*         Should contain a message value that is meaningful
         about what went wrong.
*/
```

```
public boolean isValid(int id, HashMap values) throws
  IdValidateException;
```

```
/**
  The name of the hashkey in the HashMap passed to IdValidate.isValid(..)
  that refers to a current database connection to the Unica Plan
  system tables.
  This connection is available for use to implementations of this
  interface.
*/
  public final String PLAN_DB_CONNECTION = "dbconnection";
  /**
   * The name of the hashkey in the HashMap passed to
   idValidate.isValid(..) that refers to the id of the related
   template.
   */
 public final String OBJECT_TEMPLATE_ID = "templateid";
  /**
   * The name of the hashkey in the HashMap pass to
   * IdValidate.isValid(..) that refers to another Hashmap which
   * contains name/value pairs. The name corresponds to a field on
   * the screen for project/program and the value corresponds to the
   * user entered text or selection.
   */
 public final String OBJECT_ATTRIB_VALUES = "attributeValues";
 }
```

Interface - IdGenerate

```
package com.unicaorp.uap.common.template;
import java.util.HashMap;
/* This is an interface to be implemented by the end user
* of a Unica Plan
```

```
* system for the purpose of generating unique Project Code (PIDs). The
 intent
* is to allow users to attach to existing enterprise systems to help make
* project IDs meaningful in their enterprise.
*
* Implementations of this Interface are called by the Unica Plan Server.
* It is the responsibility of the Unica Plan Server
* to assure that there is
* only one ID being generated at a time. When implementation of this
* interface are called, they can assume that there are no other IDs
* that are being generated concurrently.
*/
```

```
public interface IdGenerate {
    /**
  * Returns a string code used to define a Project object with Unica Plan
  *
  * @param uniqueId - This is an integer value that is generated by
  *  the Unica Plan system. This is guaranteed to be unique across
  *  the system; hence, if the project ID returned is the string
  *  representation of this integer, it will be a unique
  *  Project Code (PID).
  *
  * @param values - This is a set of name/value pairs, referring to the
 current
  *  database connection, appropriate template id, code prefix,
  *  request flag, and another HashMap that contains name/value
  *  pairs, corresponding to the fields and values on the screen.
  *
  * @param uniqueChecker - An implementation used to verify the uniqueness
 of
  *  of ID's generated by this instance.
```

```
  *
  * @return    - A string that represents the ID of the project we are
         creating.
  *
  * @throws com.unicacorp.uap.user.IdGenerateException
  *  Should contain a message value that is meaningful about
  *  what went wrong
*/
```

```
    public String generateID (int uniqueId, HashMap values, IdUniqueChecker
 uniqueChecker)
 throws IdGenerateException;
 /**
  * The name of the hashkey in the HashMap passed to IdValidate.isValid(..)
  * that refers to a current database connection to the Unica Plan
  * system tables.
  * This connection is available for use to implementations of this
 interface.
  */
 public final String PLAN_DB_CONNECTION = "dbconnection";
 /**
  * The name of the hashkey in the HashMap passed to IdValidate.isValid(..)
  * that refers to the id of the related template.
  */
```

```
 public final String OBJECT_TEMPLATE_ID = "templateid";
 /**
  * The name of the hashkey in the HashMap passed to IdValidate.isValid(..)
  * that refers to the desired string prefix to prepend the generated id.
  */
 public final String OBJECT_CODE_PREFIX = "pidprefix";
```

```
/**
 * The name of the hashkey in the HashMap passed to IdValidate.isValid(..)
 * that refers that indicates whether the calling object is a request.
 */
```

```
public final String OBJECT_REQUEST_FLAG = "flagprojectrequest";
/**
 * The name of the hashkey in the HashMap pass to IdValidate.isValid(..)
 * that refers to another Hashmap which contains name/value pairs. The
name
 * corresponds to a field on the screen for project/program and the value
 * corresponds to the user entered text or selection.
 */
public final String OBJECT_ATTRIB_VALUES = "attributeValues";
/**
 * Default start plan code start number
 */
```

```
public final int PLAN_CODE_SUFFIX_START = 1000;
/**
 * Default start program code start number
 */
public final int PROGRAM_CODE_SUFFIX_START = 1000;
/**
 * Default start project code start number
 */
public final int PROJECT_CODE_SUFFIX_START = 1000;
/**
 * Default start rfq code start number
 */
public final int RFQ_CODE_SUFFIX_START = 1000;}
```

Custom ID generator

```
package com.unica.uap.component.helper;
import com.unicacorp.uap.common.db.*;
import com.unicacorp.uap.common.template.*;
import org.apache.commons.lang.StringUtils;
import java.io.File;
import java.io.FileInputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.Properties;
```

```
/**
* The Class CustomComponentPidGenerateImpl.
*/
public class CustomComponentPidGenerateImpl implements IdGenerate,
 IdUniqueChecker {
 /** The lower limit. */
 public static int LOWER_LIMIT = 0;
 /** The upper limit. */
 public static int UPPER_LIMIT = 0;
 static {
  Properties attrPro = new Properties();
  try {
   String planHome = System.getProperty("plan.home");
   System.out.println("planHome : " + planHome);
```

```
  File file = new File(planHome + "/unwar/WEB-INF/IDRange.properties");
  FileInputStream fi = new FileInputStream(file);
  if (fi != null) {
   attrPro.load(fi);
   String min = (String) attrPro.get("mktOBJId.min");
   String max = (String) attrPro.get("mktOBJId.max");
   LOWER_LIMIT = Integer.parseInt(min);
   UPPER_LIMIT = Integer.parseInt(max);
   System.out.println("Lower Limit :" + LOWER_LIMIT);
   System.out.println("Upper Limit :" + UPPER_LIMIT);
  } else {
   System.out.println("IDRange Property file can not be found");
   throw new RuntimeException("IDRange Property file can not be found");
  }
 } catch (Exception e) {
  e.printStackTrace();
  throw new RuntimeException("IDRange Property file can not be found");
 }
}
```

```
/**
 * The Constructor.
 */
public CustomComponentPidGenerateImpl() {
}
/**
 * Generate ID.
 *
 * @param uniqueChecker the unique checker
 * @param values the values
 * @param instanceId the instance id
```

```
 *
 * @return the string
 *
 * @throws IdGenerateException the id generate exception
 */
  public synchronized String generateID(int instanceId, HashMap values,
 IdUniqueChecker uniqueChecker) throws IdGenerateException {
 print("inside 'generateID' method");
 print("instanceId : " + instanceId);
 print("####################\n" + values + "####################\n");
 String prefix = (String) values.get("pidprefix");
 print("prefix : " + prefix);
 String templateid = (String) values.get("templateid");
 print("templateid : " + templateid);
 Connection con = (Connection) values.get("dbconnection");
 //int nextValue = -1;
 boolean isEmptyPrefix = false;
 try {
  if (StringUtils.isEmpty(prefix)) {
   isEmptyPrefix = true;
  }
```

```
  //GET THE CURRENT VALUE OF THE TEMPLATE ID - from CUST_GENIDS table
  String sqlString = "SELECT ID_VALUE FROM CUST_GENIDS WHERE ENTITY_NAME
= ?";
  print("sqlString : " + sqlString);
  PreparedStatement ps = null;
  ResultSet rs = null;
  int cnt = 0;
  try {
   ps = new UAPSQLPreparedStatement(con, sqlString);
```

```
 UAPSQLUtils.setupPreparedStatement(ps, 1, templateid, "string");
 rs = ps.executeQuery();
 if (rs.next()) {
  cnt = rs.getInt(1);
 }
 print("current ID vlaue :" + cnt);
 UAPSQLUtils.closeResultSet(rs, ps);
} catch (SQLException ex) {
 ex.printStackTrace();
 UAPSQLUtils.closeResultSet(rs, ps);
 throw new RuntimeException(ex);
} catch (Exception exception) {
 exception.printStackTrace();
 UAPSQLUtils.closeResultSet(rs, ps);
 throw new RuntimeException(exception);
}
if (cnt == 0) {
 //insert first new record for the template id into table
 cnt = LOWER_LIMIT;
 String sqlInsertStr = "INSERT INTO CUST_GENIDS values (?,?)";
 print("sqlInsertStr : " + sqlInsertStr);
 ps = new UAPSQLPreparedStatement(con, sqlInsertStr);
 ps.setString(1, templateid);
 ps.setInt(2, cnt);
     }
```

```
     else if ((cnt >= LOWER_LIMIT) && (cnt < UPPER_LIMIT)) {
 //increase the counter and update the row for the template id
 cnt++;
 String sqlUpdateStr =
  "UPDATE CUST_GENIDS SET ID_VALUE= ? WHERE ENTITY_NAME = ?";
```

```
    print("Update : " + sqlUpdateStr);
    ps = new UAPSQLPreparedStatement(con, sqlUpdateStr);
    ps.setInt(1, cnt);
    ps.setString(2, templateid);
   } else {
    print("Current ID is out of range, ID Range [" + LOWER_LIMIT +
     "-" + UPPER_LIMIT + "]");
     //throw exception that can not generate id, limit is over
     throw new IdGenerateException(
      "Current ID is out of range, ID Range [" + LOWER_LIMIT +
      "-" + UPPER_LIMIT + "]");
        }
   //UAPSQLUtils.beginTransaction(con);
   ps.execute();
   //UAPSQLUtils.endTransaction(con, true);
   String pid = (isEmptyPrefix ? "" : prefix) + cnt;
   print("return from 'generateID'  method with pid : " + pid);
   return pid;
  } catch (Exception ex) {
   ex.printStackTrace();
   throw new IdGenerateException(ex);
  }
}
```

```
/**
 * Checks if is unique.
 *
 * @param values the values
 * @param Id the Id
 *
 * @return true, if is unique
```

```
 */
public boolean isUnique(String Id, HashMap values) {
 print("inside 'isUnique' method");
 //provide actual implementation for uniqueness check
 return true;
 }
/**
 * Print.
 *
 * @param str the str
 */
private void print(String str) {
 System.out.println(str);
}
```

Grid validation

Unica Plan exposes a validation interface that you can use to write custom data validation rules.

A sample rule that uses the Validator interface and the com.unicacorp.uap.grid.validation.plugin.GridValidatorPluginImpl validator is delivered with Unica Plan.

When you add a form that contains an editable grid to a project template, you can specify a data validation class and rule to validate data that is entered in the grid.

When you work with validation rules, be aware that:

- The files that contain your validation rules must use a specific format. When you import a rules XML file, it is validated against the `gridrules.xsd` XML schema.
- Typically, a rule is specific to a form, as rules are tightly bound to the underlying table structure. As a result, use each rule with only one (editable grid) form.
- Unica Plan ships with several sample rules that you can use for most validation tasks. You can create and import more custom rules files as needed.

For information about working with editable grids, see Creating grids (on page 171). For information about importing data validation rules, see Data validation rules (on page 338).

Validator interface

The Validator interface exposes the following functions: init(config:GridConfig), process(rulesToExecute:Validator.RulesEnum), and destroy().

Function	Description
init(config:GridConfig)	This function initializes the validator.
process(rulesToExecute:Validator.RulesEnum)	This function executes the validation rules. The rulesToExecute parameter determines which type of rule validation plug-in executes. It is an enumerated value that can have the following values: • allRules • gridRules • rowRules
destroy()	This function is a destructor for the object, and does garbage collection.

Unica Plan provides a sample validator, `RangeCheckRule.java`. This object takes a grid as input, then iterates through all of the records in the grid, validating against rules defined in an XML file.

Data validation rules

Rules are a set of data validation functions that you define in an XML file, then import and associate with a form.

If a form has an associated set of data validation rules, Unica Plan applies them automatically when users enter data for a marketing object instance. For more information on how to set up data validation rules, see Grid validation (on page 337).

On the Rules Definitions page you can:

- Click **Add Rules Definition** to load an XML rules definition file.

  After you add a rule, you link it to tabs that collect data using a grid-style form: Edit the template and click the Tabs tab. For details, see Template Tabs tab for customizing the user interface (on page 110).
- Click **Delete** to delete a rule (if it is not in use by any templates).
- Click a rule to update its rule file, or to change the name of the rule.

Note: If you attempt to overwrite an existing rules file, the system generates a warning.

Navigating to the Rules Definitions page

1. Select **Settings > Plan Settings**.
2. In the Other Options section, click **Template Configuration**.
3. In the Templates Components section, click **Rules**.

Validation rules

The Validator interface works by invoking a series of rules, and comparing entered data values against the rules. Each rule is an executable Java™ file that implements the Rule interface.

The validation plug-in supports two types of rules.

- ROW: row-level rules are executed first.
- GRID: grid-level rules are executed after row-level rules.

When a user saves data for a grid that uses a given validation plug-in, all of the rules are applied. The row-level rules are applied first, followed by the grid-level rules. Rules are applied in the order they are declared in the rules file.

Data validation rules file structure

A validation rules file is an XML file that contains one or more rules. Each rule can contain the following tags.

Table 78. Tags for validation rules files

Tag	Description
rule	Begins the rule and sets the rule type, which can be either ROW or GRID.
name	The name of the rule.
desc	A text description of the rule.
enable	A boolean value for enabling or disabling the rule: • false: the rule is disabled • true: the rule is enabled
applies-to-tvc-id	The internal name of the grid component for which the rule applies. To apply the rule to multiple grids, use a separate applies-to-tvc-id tag for each grid component. This tag is optional; if omitted, the rule is applied to all grids on the specified form.
class	The Java™ class containing the commands for processing the rule. To use the sample range check rule, you enter: `com.unicacorp-.uap.grid.validation.rule.basic.RangeCheckRule`
set-property	The set-property tag passes parameters to the rules. Each rule can contain zero or more set-property tags.

Sample data validation rules

Unica Plan ships with the following sample rules.

Table 79. Sample data validation rules

Rule	Description

BeginsWithRule	Ensures the text column being validated begins with the specified character. You set the beginCharacter and column properties. For example:
	`<set-property property="beginCharacter" value="A"/>`
	`<set-property property="column"`
	`value="dyn_vendors.Name"/>`
	This rule checks the Name field in the dyn_vendors database table to make sure that the value begins with the letter A.
	Class name: `com.unicacorp.uap.grid.validation.rule-.basic.BeginsWithRule`

DateCheckRule	Ensures the date column being validated falls within the specified range. Set the following properties: greaterThan, lessThan, column.
	For example:
	`<set-property property="greaterThan"`
	`value="12/31/1999"/>`
	`<set-property property="lessThan" value="Today"/>`
	`<set-property property="column" value=`
	`"dyn_vendors.invoiceDate"/>`
	This rule checks the invoiceDate field in the dyn_vendors database table to make sure that the value is not before the year 2000.
	Optionally, you can also set the dateFormat property. If you add this property, dates must be entered in the specified format. You can set

Table 79. Sample data validation rules (continued)

Rule	Description
	the following format values: dd/MM/yyyy, MM/dd/yyyy, dd/MM/yy, MM/dd/yy, yyyy-MM-dd, yyyy.MM.dd Class name: `com.unicacorp.uap.grid.validation.rule-.basic.DateCheckRule`

RangeCheck-Rule	Ensures the integer column being validated falls within the specified range. You set the following properties: minValue, maxValue, column
	
	For example:
	
	`<set-property property="minValue" value="1"/>`
	`<set-property property="maxValue" value="999999"/>`
	`<set-property property="column" value=`
	`"dyn_vendors.numEmployees"/>`
	
	This rule checks the numEmployees field in the dyn_vendors database table to make sure that the value is between 1 and 999,999.
	
	Class name: `com.unicacorp.uap.grid.validation.rule-.basic.RangeCheckRule`

UniqueCheck-Rule	Ensures that the column being validated does not contain duplicate values. You set the column property. Class name: `com.unicacorp.uap.grid.validation.rule-.basic.UniqueCheckRule` **Note:** This rule is always applied across the entire grid, even if you use ROW to indicate that this is a row-level rule.

Validation rule example

The steps that follow describe how to import a validation rule file into Unica Plan, add it to a template, and test it.

The following example rule ensures that the value supplied for the `numEmps` field in a grid is 10 - 1000.

```
<?xml version="1.0" encoding="UTF-8"?>
   <validation-rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <rule validationType="ROW">
      <name>Range Check</name>
      <desc>Checks Range for numeric type. Note: for field numEmps</desc>

 <class>com.unicacorp.uap.grid.validation.rule.basic.RangeCheckRule</class>
      <enable>true</enable>
      <applies-to-tvc-id>vendors</applies-to-tvc-id>
      <set-property property="minValue" value="10"/>
      <set-property property="maxValue" value="1000"/>
      <set-property property="column" value="dyn_vendors.numEmps"/>
   </rule>
</validation-rules>
```

In this rule, the `numEmps` field is on a form with an internal name of `vendors`. The procedure that follows assumes that you have this form set up on your system.

1. Create an XML file to contain the example validation rule.
2. Upload the rules file into Unica Plan:
    a. Click **Settings > Plan Settings > Template Configuration > Rules**.
    b. Click **Add Rules Definition**.
    c. In the **Update Rule** dialog box, supply an identifying name and specify your XML file.
    d. Click **Continue** to add the rules file to Unica Plan.
3. Assign the rules file to a tab on a template.

a. Click **Settings > Plan Settings > Template Configuration > Templates**.

b. Choose a template, and then click its **Tabs** tab.

c. Click **Add Tab**. For the **Form**, select vendors. For the **Data Validation Rules**, select the identifying name of your rule.

The system supplies the **Data Validation Class** when you select a rules file.

4. Save your changes and use the template to create an object.
5. To test the rule, enter invalid data in the **empNum** field.

For example, enter `5000`. If you receive an error message, the rule is working as designed.

Chapter 22. Unica Plan REST API

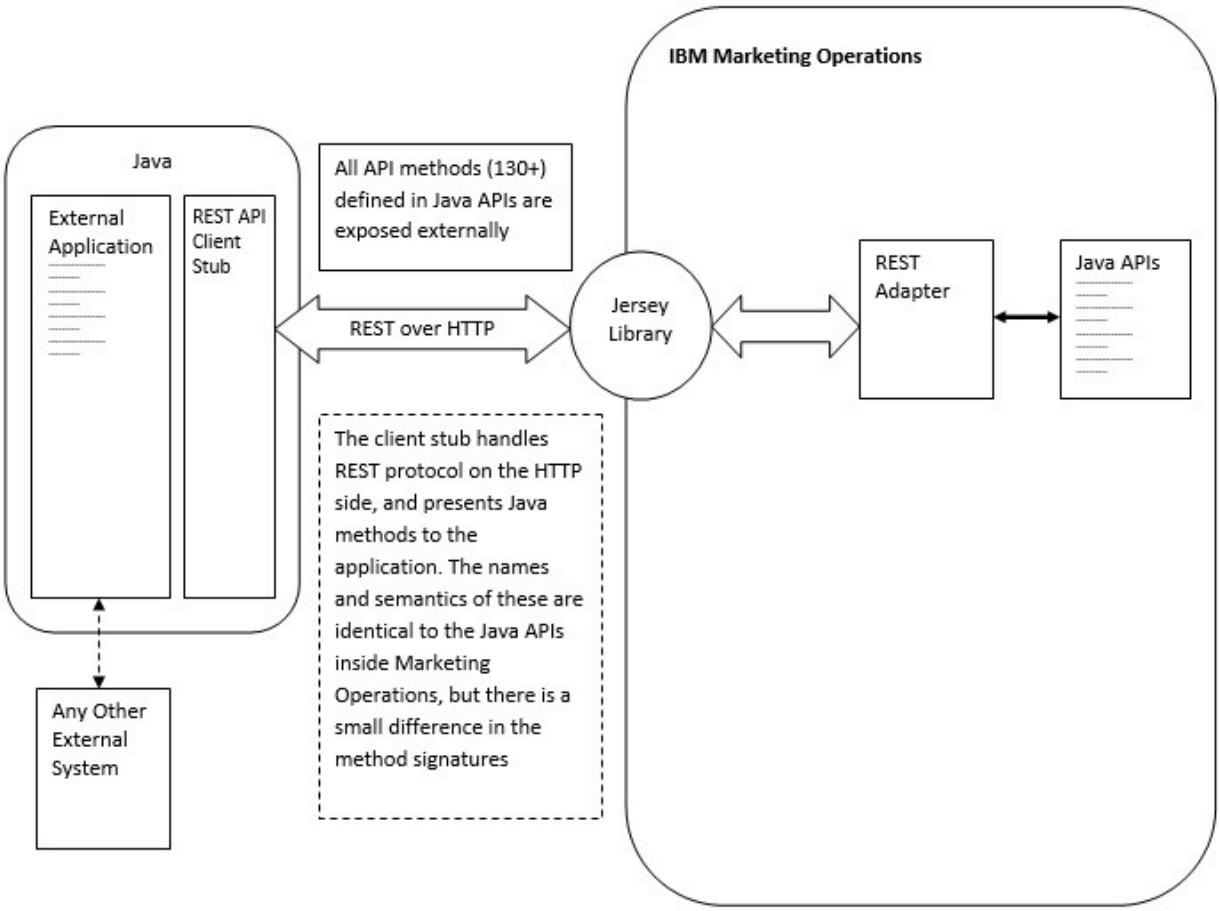
The Unica Plan REST API is a façade that provides a client view of a running Unica Plan instance.

The API supports the following types of operations.

- Component creation and deletion
- Discovery (by component type, attribute value, and more values)
- Component inspection (through its attributes, specialized links, and more values)
- Component modification

The Unica Plan REST APIs are externally exposed. Therefore, they do not require integration services the way that the Unica Plan SOAP APIs requires the integration services.

Note: Unica Plan APIs are intended for Administrator use only.



Process details

The base path for REST API client stub is `<MO Home>\devkits-rest\integration`.

The client stub and other required libraries are located at `<MO Home>\devkits-rest\integration\lib`.

Example code for usage of the client stub is located at `<MO Home>\devkits-rest\integration\examples`.

The directory `bin` under this directory has a sample batch file that can build the example code. This example code has usage details of various API methods than you can program.

Javadocs for the JAVA methods supported for the REST API are located at `<MO Home>\devkits-rest\integration\javadocs`.

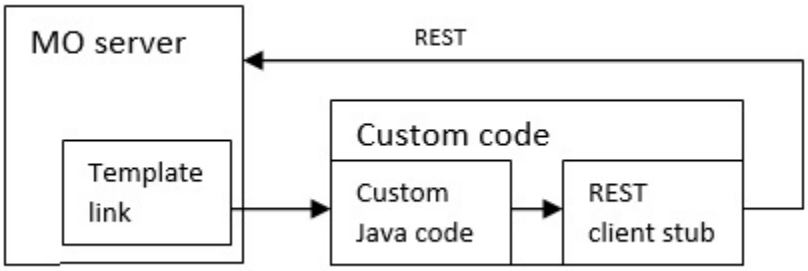
To write your own custom JAVA code to call the REST APIs:

- Write your custom code (refer to the Javadocs and the example code if needed).
- Add all `.jar` files in the lib directory to classpath when you compile and run your custom code.

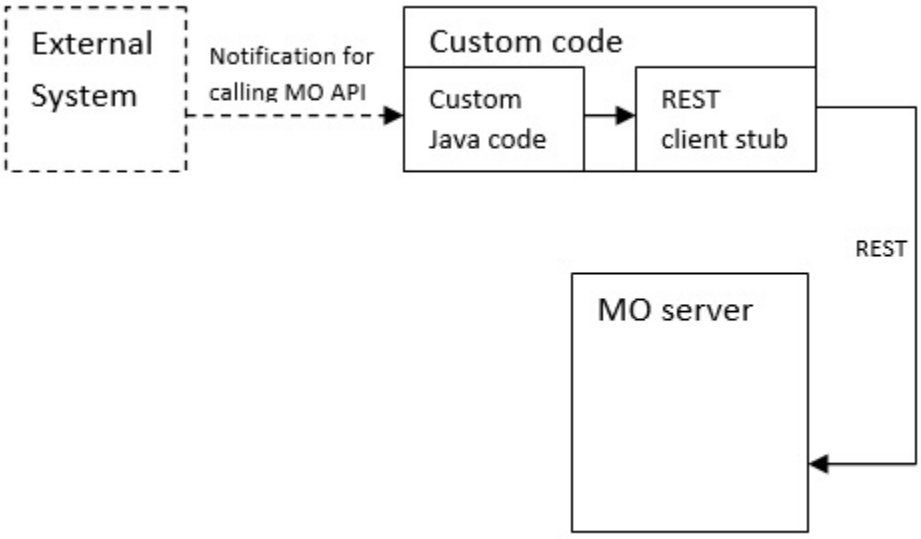
Typical usage scenarios

The following two examples illustrate usage of the REST APIs.

Usage scenario with template links:



Standalone or external trigger usage scenario



Rest API reference documentation

For more information see, REST API reference documentation.

For more information see, REST API reference documentation.

Contents of the Unica Plan REST API

The com.unica.publicapi.plan.api package delivers the Unica Plan API.

This package offers interfaces and exceptions, and contains the following types of classes:

- Enumerated data types.
- Handles to identify object and component instances.
- An AttributeMap.

Complete documentation of the API, including all methods and possible values, is available by clicking **Help > Product Documentation** in an instance of Unica Plan, then downloading the `<version>PublicAPI-RestClient.zip` file.

REST API interfaces

The Unica Plan application programming interface (API) includes IPlanAPI, IRESTAPIClient, and IPlugin interfaces.

The Unica Plan API includes the following interfaces.

IPlanAPI

Defines the public API for Unica Plan. Provides methods for creating, discovering, and modifying objects, including folders, projects, programs, workflow tasks, and team members.

For systems that have the optional integration with Unica Campaign enabled, also provides methods for creating, discovering, and modifying offers.

IRESTAPIClient

Defines the public API for Unica Plan REST client stub. This client stub implements every REST API call supported by Unica Plan REST based APIs. This client stub is allows you to quickly develop code to call the Unica Plan REST APIs with JSON. The stub also has the code to connect and

communicate with the desired Unica Plan server, given the host and port details.

IPlugin

Defines the generic interface that must be implemented by all plan plugins.

REST API methods

For specific information about the public API methods, refer to the iPlanAPI class in the API documentation files.

These files are available by logging in to Unica Plan and selecting **Help > Product Documentation** from any page, and then downloading the `<version>PublicAPI-RestClient.zip` file.

REST API exceptions

The REST API only throws one exception, RestApiException.

The RestApiException has an integer code and is a text string message.

REST API handles

A handle is special URL object that references a particular object instance in an Unica Plan instance. Handles include the component type, internal data identifier, and an instance base URL.

Handles used or generated by the API can be externalized to a full URL. You can use the resulting URL in different ways. You can use the URL to open a view of the component in the Unica Plan GUI, send it in email messages, or use it in another procedure as a parameter.

Handles are valid only for a particular Unica Plan service instance or clustered instance, but are valid for the lifetime of the deployed service. As a result, handles can be saved in a file for later reference, but they cannot be used to access components on another Unica Plan instance. This restriction also applies to instances on the same physical host server. Unica Plan does provide, however, a mechanism for mapping different base URLs to the current instance to accommodate relocating an instance to another server (for example, if the equipment malfunctions).

Handles are client-independent.

Members of the Handle class have factory methods for creating handles from various types of URLs. Examples follow for several handles.

Project handles:

```
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=projecttabs&projectid=102"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=projecttabs&projectid=103"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=projecttabs&projectid=104"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=projecttabs&projectid=105"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=projecttabs&projectid=106"}}
```

Program handles:

```
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=programtabs&programid=112"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=programtabs&programid=113"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=programtabs&programid=114"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=programtabs&programid=115"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=programtabs&programid=116"}}
```

User handles:

```
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=adminuserpermissions&func=edit&userId=2"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
```

```
.jsp?cat=adminuserpermissions&func=edit&userId=3"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=adminuserpermissions&func=edit&userId=4"}}
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=adminuserpermissions&func=edit&userId=5"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=adminuserpermissions&func=edit&userId=6"}}
```

Team handles:

```
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=teamdetails&func=edit&teamid=100001"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=teamdetails&func=edit&teamid=100002"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=teamdetails&func=edit&teamid=100003"}}
```

Task handles:

```
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=projectworkflow&projectid=102&taskid=106"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=projectworkflow&projectid=103&taskid=141"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=projectworkflow&projectid=104&taskid=176"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=projectworkflow&projectid=105&taskid=211"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=projectworkflow&projectid=106&taskid=246"}}
```

Approval handle:

```
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=approvaldetail&approvalid=149"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
```

```
.jsp?cat=approvaldetail&approvalid=150"}}
{"value":{"multiTypeValue":"http://mymachine:9021/plan/affiniumplan
.jsp?cat=approvaldetail&approvalid=151"}}
```

Marketing object handles:

```
hRestTestMO_01={"value":{"multiTypeValue":"http//mymachine:9999/plan/affini
umplan
.jsp?cat=componenttabs&componentid=resttestmo02&componentinstid=130"}}
hRestTestMO_02={"value":{"multiTypeValue":"http://mymachine:9999/plan/affin
iumplan
.jsp?cat=componenttabs&componentid=resttestmo02&componentinstid=130"}}
hRestTestMO_03={"value":{"multiTypeValue":"http://mymachine:9999/plan/affin
iumplan
.jsp?cat=componenttabs&componentid=resttestmo02&componentinstid=131"}}
hRestTestMO_04={"value":{"multiTypeValue":"http://mymachine:9999/plan/affin
iumplan
.jsp?cat=componenttabs&componentid=resttestmo02&componentinstid=132"}}
hRestTestMO_05={"value":{"multiTypeValue":"http://mymachine:9999/plan/affin
iumplan
.jsp?cat=componenttabs&componentid=resttestmo02&componentinstid=133"}}
```

Project request handle:

```
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=projecttabs&projectid=122"}}
```

Asset library handle:

```
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=library&id=101"}}
```

Asset handles:

```
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=asset&assetMode=VIEW_ASSET&assetid=123"}}
```

```
{"value":{"multiTypeValue":"http://mymachine:9999/plan/affiniumplan
.jsp?cat=asset&assetMode=VIEW_ASSET&assetid=124"}}
```

REST API AttributeMap

The AttributeMap class is a JSON map that contains only attributes. The attribute <Name> is the map entry key, and the attribute <values> array (note plural) is the map entry value.

The AttributeMap class includes the following fields.

- <*Name*>: the programmatic name of the attribute. This name serves as a unique key for accessing the attribute within the component instance in which it occurs.

  Note: <*Name*> is not necessarily the display name that is presented to a user in the GUI. For components that are created from templates (such as projects or workflow tasks), the attribute name is specified by the template element definition. The attribute name must be unique. For other components, the attribute name typically is derived programmatically from the server-side component instance.

  Note: By convention, custom attributes include the name of the form in which the editable version is defined: <*form_name*>.<*attribute_name*>.

- *Values*: a JSON object array, containing zero or more attribute values. The type of each value must be the same and agree with the type of the attribute as it is defined in Unica Plan.

The metadata of an attribute (such as translated display name and description) is defined by the template that is associated with the attribute and its parent object instance. Attributes provide a simple yet extensible mechanism for showing both required and optional object instance attributes, such as project name, code, and start date.

Attribute maps:

{"multiTypeMap":{"uapDescription":[{"valueType":"java.lang.String",
"multiTypeValue":"abcdProjDesc4"}],"uapName":[{"valueType":"java.lang.
String","multiTypeValue":"RestTestProj04"}]}}

{"multiTypeMap":{"uapDescription":[{"valueType":"java.lang.String",
"multiTypeValue":"abcdProjDesc4"}],"uapName":[{"valueType":
"java.lang.String","multiTypeValue":"RestTestProg04"}]}}

{"multiTypeMap":{"uapDescription":[{"valueType":"java.lang.String",
"multiTypeValue":"abcdProjDesc4"}],"uapName":[{"valueType":"java.
lang.String","multiTypeValue":"RestTestProg09"}]}}

{"multiTypeMap":{"uapDescription":[{"valueType":"java.lang.String",
"multiTypeValue":"ReqDesacrption"}],"uapName":[{"valueType":"java.
lang.String","multiTypeValue":"RestTestProjReq01"}]}}

{"multiTypeMap":{"uapName":[{"valueType":"java.lang.String",
"multiTypeValue":"RestTestProjReq03"}]}}

{"multiTypeMap":{"uapName":[{"valueType":"java.lang.String",
"multiTypeValue":"RestTestProjReq04"}]}}

{"multiTypeMap":{"uapName":[{"valueType":"java.lang.String",
"multiTypeValue":"RestTestProjReq05"}]}}

{"multiTypeMap":{"uapDescription":[{"valueType":"java.lang.String",
"multiTypeValue":"changedDescription"}]}}

{"multiTypeMap":{"uapStartDate":[{"valueType":"com.unica.uap.
restapi.multitype.values.GregorianCalendarValue","multiTypeValue":
"2015-06-20 00:08:00"}],"uapEndDate":[{"valueType":"com.unica.uap.

```
restapi.multitype.values.GregorianCalendarValue","multiTypeValue":
"2015-08-30 00:08:00"}]}}
```

Enumerations:

```
{"value":{"multiTypeValue":"IN_PROGRESS"}}
{"value":{"multiTypeValue":"ON_HOLD"}}
{"value":{"multiTypeValue":"SUBMITTED"}}
{"value":{"multiTypeValue":"RETURNED"}}
{"value":{"multiTypeValue":"DRAFT"}}
{"value":{"multiTypeValue":"ACCEPTED"}}
{"value":{"multiTypeValue":"PENDING"}}
{"value":{"multiTypeValue":"ACTIVE"}}
{"value":{"multiTypeValue":"FINISHED"}}
{"value":{"multiTypeValue":"SKIPPED"}}
{"value":{"multiTypeValue":"DISABLED"}}
{"value":{"multiTypeValue":"ProjectOwner"}}
{"value":{"multiTypeValue":"ProjectParticipant"}}
{"value":{"multiTypeValue":"ProjectRequester"}}
{"value":{"multiTypeValue":"ProjectRequestRecipient"}}
{"value":{"multiTypeValue":"ProjectRequestOwner"}}
{"value":{"multiTypeValue":"ApprovalOwner"}}
{"value":{"multiTypeValue":"ApprovalApprover"}}
{"value":{"multiTypeValue":"ProgramOwner"}}
{"value":{"multiTypeValue":"ProgramParticipant"}}
{"value":{"multiTypeValue":"COPY_USING_PROJECT_METRICS"}}
{"value":{"multiTypeValue":"COPY_USING_TEMMPLATE_METRICS"}}
{"value":{"multiTypeValue":"DRAFT"}}
{"value":{"multiTypeValue":"FINALIZED"}}
{"value":{"multiTypeValue":"LOCKED"}}
{"value":{"multiTypeValue":"ARCHIVED"}}
```

Array of values:

```
{"valueArray":[{"valueType":"java.lang.String","multiTypeValue":
"RestTestProj01"}]}

{"valueArray":[{"valueType":"java.lang.String","multiTypeValue":
"TRS1000"}]}

{"valueArray":[{"valueType":"java.lang.String","multiTypeValue":
"changedDescription"}]}

{"valueArray":[{"valueType":"java.lang.String","multiTypeValue":
"RestTestProj02"}]}

{"valueArray":[{"valueType":"java.lang.String","multiTypeValue":
"RestTestProjReq01"}]}

{"valueArray":[{"valueType":"java.lang.String","multiTypeValue":
"TRS1018"}]}

{"valueArray":[{"valueType":"java.lang.String","multiTypeValue":
"ReqDesacrption  "}]}

{"valueArray":[{"valueType":"com.unica.uap.restapi.multitype.values.
HandleValue","multiTypeValue":"http://mymachine:9021/plan/affiniumplan.
jsp?cat=projecttabs&projectid=103"},{"valueType":"com.unica.uap.
restapi.multitype.values.HandleValue","multiTypeValue":
"http://mymachine:9021/plan/affiniumplan.jsp?cat=
projecttabs&projectid=104"}]}

{"valueArray":[{"valueType":"com.unica.uap.restapi.multitype.values.
HandleValue","multiTypeValue":"http://mymachine:9021/plan/affiniumplan.
jsp?cat=projecttabs&projectid=107"},{"valueType":"com.unica.uap.restapi.
multitype.values.HandleValue","multiTypeValue":"http://mymachine:9021/
```

```
plan/affiniumplan.jsp?cat=projecttabs&projectid=108"}]}

{"valueArray":[{"valueType":"com.unica.uap.restapi.multitype.values.
HandleValue","multiTypeValue":"http://mymachine:9021/plan/affiniumplan.
jsp?cat=projecttabs&projectid=102"},{"valueType":"com.unica.uap.
restapi.multitype.values.HandleValue","multiTypeValue":
"http://mymachine:9021/plan/affiniumplan.jsp?cat=projecttabs
&projectid=106"}]}

{"valueArray":[{"valueType":"com.unica.uap.restapi.multitype.values.
HandleValue","multiTypeValue":"http://mymachine:9021/plan/
affiniumplan.jsp?cat=projecttabs&projectid=105"}]}

{"valueArray":[{"valueType":"com.unica.uap.restapi.multitype.
values.HandleValue","multiTypeValue":"http://mymachine:9021/plan
/affiniumplan.jsp?cat=adminuserpermissions&func=edit&userId=3"}]}

{"valueArray":[{"valueType":"com.unica.uap.restapi.multitype.
values.HandleValue","multiTypeValue":"http://mymachine:9021/plan/
affiniumplan.jsp?cat=adminuserpermissions&func=edit&userId=2"}]}

{"valueArray":[{"valueType":"com.unica.uap.restapi.multitype.
values.HandleValue","multiTypeValue":"http://mymachine:9021/plan/
affiniumplan.jsp?cat=adminuserpermissions&func=edit&userId=2"},
{"valueType":"com.unica.uap.restapi.multitype.values.
HandleValue","multiTypeValue":"http://mymachine:9021/plan/
affiniumplan.jsp?cat=adminuserpermissions&func=edit&userId=3"}]}
```


Configuring the Unica Plan REST API in Unica Platform

Use this information to configure the Unica Plan REST API in Unica Platform.

1. Under Unica Platform | General | Miscellaneous, set a `token lifetime`. This value is the number of seconds the token from the login method is valid. For example, you could set this value to `86400` so the token is valid for a day after login. By default this value is `20` seconds. This configuration does not refresh with a call to an API. If this time expires, you have to make a login call, obtain a new token, and use it in the subsequent API calls.

   Note: Reducing the value of `token lifetime` increases the number of requests to keep the Offer selector panel open. We recommend that you do not reduce the value of `token lifetime`.

2. Under Unica Platform | Security | API management, configure the following settings.

   ```
   Enable session-based API authentication: (Enabled)
   Delete security token after a single use: (Disabled)
   ```

   If **Delete security token after a single use** is enabled, a login call is required before every API call.

3. Under Unica Platform Security | API management |, configure the following settings for authentication.

   ```
   API URI: /authentication/login
   Block API access: (Disabled)
   Secure API access over HTTPS: (Disabled)
   Require authentication for API access: (Enabled)
   ```

4. Click **API configuration template**, and add a logout method that the client stub uses.

   ```
   New category name: Logout
   API URI: /authentication/logout
   Block API access:(Disabled)
   Secure API access over HTTPS: (Disabled)
   Require authentication for API access: (Enabled)
   ```

5. Under **API configuration template**, add `REST API Ver 1.0`

```
New category name: REST API Ver 1.0
API URI: /*
Block API access: (Disabled)
Secure API access over HTTPS: (Disabled)
Require authentication for API access: (Enabled)
Enable SSL if you plan to use SSL
```

6. Select **Plan V2 APIs > Edit settings** and configure the following values:

```
API URI: /v2/*
Block API access: (Disabled)
Secure API access over HTTPS: (Disabled)
Require authentication for API access: (Enabled)
```

7. Click **Save changes** to save the configurations.
8. Under Unica Platform Security | API management |, only one hard coded user can be used. This user must be a Unica Plan admin user, and the password for this user should not be blank.
9. Use the following REST API URL structure in your configurations.

   URL for authentication: `http://<Host>:<Port>/unica/api/manager/authentication`

   The host and port should point to Unica Platform for authentication.

   Unica Plan API URL: `http://<Host>:<Port>/plan/api/plan/V1`

   The host and port should point to Unica Plan.

Changes after the Unica Platform SSO integration

After you integrate the Unica Plan REST API with Unica Platform SSO, you must use specific credentials to authenticate.

`PlanAPIUser` is the only the user that can access the Unica Plan REST APIs.

This user requires a password and cannot be left blank.

Use Swagger User Interface as Rest API Client

The Rest APIs of Plan are now available with Swagger User Interface. Once the Application is up and running, The APIs are available for testing and utilization by launching the Swagger UI URL: [http://[MO_HOST_NAME]:[MO_PORT]/plan/dist/index.html

The header parameters need to be configured only once by getting a valid token ID. Each API has information about the required and optional parameters for usage along with sample expected result. Swagger UI takes care of sending the parameters as required(Either in header or in Body) and result is available using "Try it out" option. Procedure: Make the configuration changes in Unica Platform for REST API. Get a valid token from Platform using the URL below:

```
http://[HOST_NAME]:[PORT]/unica/api/manager/authentication/login
Header Parameters :
m_user_name:[USER_NAME]
m_user_password:[PASSWORD]
api_auth_mode:manager
```

```
Expected Response (example):
{
"m_tokenId": "1550222864978-2-X0V7WYew-he9r-hBaUuidI-WzLu-E6klBVba",
"m_user_name": "USER_NAME",
"createDate": "Fri Feb 15 14:57:44 IST 2019"
}
```

Use a header modifier plugn to add the following parameters:

```
m_user_name: [USER_NAME]
m_user_password: [PASSWORD]
api_auth_mode:manager
```

```
m_tokenId: [Token received via Login URL]
```