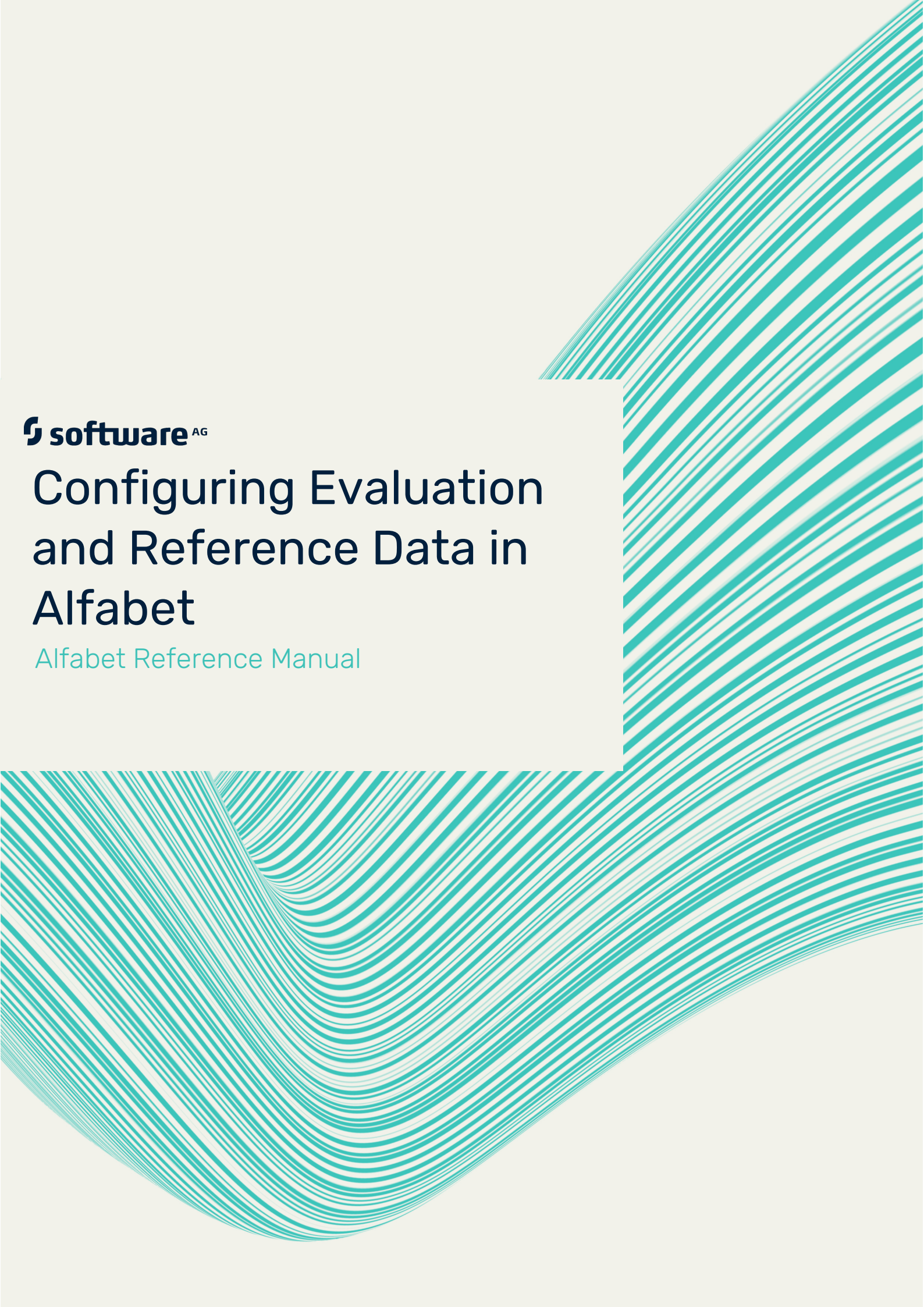


Configuring Evaluation and Reference Data in Alfabet

Alfabet Reference Manual

Documentation Version Alfabet 10.11.2

Conventions used in the documentation

Convention	Meaning
Bold	Used for all elements displayed in the Alfabet interface including, for example, menu items, tabs, buttons, dialog boxes, page view names, and commands. Example: Click **Finish** when setup is completed.
Italics	Used for emphasis, titles of chapters and manuals. this Example: see the *Administration* reference manual.
Initial Capitals	Used for attribute or property values. Example: The object state Active describes...
All Capitals	Keyboard keys Example: CTRL+SHIFT
File > Open	Used for menu actions that are to be performed by the user. Example: To exit an application, select **File > Exit**
< >	Variable user input Example: Create a new user and enter <User Name>. (Replace < > with variable data.)
	This is a note providing additional information.
	This is a note providing procedural information.
	This is a note providing an example.
	This is a note providing warning information.

Table of Contents

Chapter 1: Introduction to the Configuration of Evaluation and Reference Data in the Alfabet Interface

Various functionalities are available that allow you to configure data that are necessary to work with objects in Alfabet. The following functionalities are available via the **Configuration** menu in the user interface:

- The **Reference Data** functionality, which allows you to define currencies and currency exchange rates, cost types, income types, cost center types, connection types, connection methods, connection frequencies, connection data formats, role types, and skills that can be configured for the objects that you work with in Alfabet.
- The **Evaluations and Portfolios** functionality, which allows you to define evaluation types, indicator types, computation rules, prioritization schemes, and portfolios.
- The **Class Configuration** functionality, which allows you to assign cost types, income types, role types, default role types, diagram view items, evaluation types, prioritization schemes, portfolios, aspect evaluations, and project evaluation types to relevant object classes.
- The **Diagram Views** functionality, which allows you to define diagram views so that users can display qualitative information in diagrams.
- The **Cost Centers** functionality, which allows you to define cost centers for users working with the **Cost Management** functionality.
- The **Manage Business Documents** functionality, which you to capture business documents as enterprise assets.
- The **Color Rule Manager** functionality, which allows you to define color rules for users to use in diagram views and business support maps.
- The **Enterprise Calendars** functionality, which allows you to capture calendars for the enterprise.
- The **Integration Solutions Configuration** functionality, which allows you to create Alfabet database connection objects representing all relevant database connections configured for various integration solutions.

For security reasons, bookmarks can not be set to any views available in the **Configuration** module.

For detailed information about the configuration of data in the Alfabet interface, see the following chapters:

- Introduction to the Configuration of Evaluation and Reference Data in the Alfabet Interface
- Configuring Evaluations, Prioritization Schemes, and Portfolios
- Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities
- Configuring Cost Types and Income Types for Cost Management Capabilities
- Configuring Cost Centers for Cost Management Capabilities
- Configuring Objects for the Object Class Generic Reference Data

- Configuring the Connection Data for Information Flows
- Configuring Role Types to Define Roles in the Responsibilities Page View
- Configuring Reference and Evaluation Data Required for Project Management
- Configuring Data Retention Policies
- Configuring Third-Party Data Security Warnings
- Configuring Color Rules for Map Views and Diagram Views
- Creating Generic Attributes for an Object Class/Object Class Stereotype
- Configuring Diagram Views for Diagrams
- Configuring Enterprise Calendars for the User Community
- Managing Business Documents
- Configuring Business Roles
- Configuring Business Questions
- Configuring Semantic Connections for Integration Solutions
- Appendix: Class Configuration for Object Classes

Please note that this reference manual describes the evaluation and reference data listed above that can be configured in the Alfabet interface. Your Alfabet solution environment is largely configured by your solution designer using the configuration tool Alfabet Expand. For more information about configuration of the Alfabet solution, see the reference manual *Configuring Alfabet with Alfabet Expand*.

Chapter 2: Configuring Evaluations, Prioritization Schemes, and Portfolios

Alfabet provides many possibilities to evaluate the objects in your enterprise's IT landscape. In order to use these evaluation and reporting capabilities, evaluation types and indicator types must be configured.

An indicator type defines a dimension of measurement regarding the performance of an object in the IT landscape. The indicator is the value defined for an indicator type in the context of an object in the IT landscape.

An indicator type defines a dimension of measurement regarding the performance of an object in the IT landscape. The indicator is the value defined for an indicator type in the context of an object in the IT landscape.

A single indicator type or multiple indicator types can be bundled in an evaluation type that is used to evaluate an object. For example, the indicator types Number of Interfaces, Number of Modules, and State of the Art could be assigned to the evaluation type Complexity, and the indicator type Standardization Status could be assigned to the evaluation type also named Standardization Status.

Each evaluation type may have one indicator or more assigned to it. An indicator can be either:

- manually entered by a user
- selected from a predefined set of values by a user
- calculated by Alfabet according to defined computation rules
- calculated based on customer-specific code provided by Alfabet
- imported via an interfacing system such as ADIF

Indicator types having a predefined set of values may be associated with an icon gallery, allowing for indicators to be visualized in diagram views as well as other configured reports by means of an icon rather than a numeric value.

In a few contexts in Alfabet, indicator types are not assigned to evaluation types. For example, indicators based on a defined indicator type may be displayed in object cockpits to provide a value relevant for an object.

Finally, evaluation types can be assigned to prioritization schemes, which constitute a weighted composite of a set of evaluation types and are used to prioritize a ranking of objects or to represent the axes of a portfolio. Evaluation types as well as indicator types or prioritization schemes can be assigned to the axes of the portfolios you configure for your enterprise.

Access to evaluation types and portfolios can be controlled via user profiles.

The evaluation types you configure can be used in a number of different contexts. For example:

- in the *Evaluation Page View*. the **Evaluation** page view allows users with access permissions to an object to enter indicator values for customer-defined indicator types. The indicator values assigned to an object can then be further analyzed in the context of standard or configured reports including a wide variety of portfolio reports available in Alfabet. For more information about configuring evaluation types, see the section Configuring Evaluation Types. For an overview of the object classes for which evaluation types can be configured, see Appendix: Class Configuration for Object Classes.

- in configured prioritization schemes, which allow users to determine a prioritized ranking for a set of objects. For more information about configuring prioritization schemes, see the section Configuring Prioritization Schemes. For an overview of the object classes for which prioritization schemes can be configured, see Appendix: Class Configuration for Object Classes.
- in configured portfolios, which allow users to understand the relative performance of a set of objects that belong to the same object class in two or three independent dimensions of performance measurement. For more information about configuring portfolios, see the section Configuring Portfolios. For an overview of the object classes for which portfolios can be configured, see Appendix: Class Configuration for Object Classes.
- in aspect evaluation types, which allow users to evaluate a particular aspect of applications and components. Aspect evaluation types can only be used to evaluate applications and components. Aspect evaluation types can also be implemented in aspect prioritization schemes and aspect portfolios in order to further analyze applications and components. For more information about configuring aspect evaluation types, see the section Assigning Aspect Evaluations Types to the Object Classes Application and Component.
- in project evaluation types, which allow projects to be evaluated in the context of the project monitoring functionalities. Project evaluation types can only be used to evaluate projects. For more information about configuring project evaluation types, see the section Configuring Project Evaluation Types in the chapter Configuring Reference and Evaluation Data Required for Project Management.
- in time series evaluation types, which allow target evaluation values to be defined for objects in the Alfabet database. For more information about configuring time series evaluation types, see the section Creating a Time Series Period
- in risk management templates and indicator lookup tables which are used to assess in risk in the context of the risk management capability. For more information about the configuration required to implement the risk management capability, see the section *Risk Management Functionality* in the reference manual *User and Solution Administration*.
- in capability maps in order to evaluate the capability map according to various capability aspects. Any configured evaluation type can be assigned to a capability map. It is not necessary to assign the evaluation type to an object class to make it available for the capability map. For more information about working with business capabilities, see the section *Working with Business Capability Maps* in the reference manual *Portfolio Management Basic*.
- in configured reports. For more information about configuring reports displaying evaluation types, see the chapter *Configuring Reports* in the reference manual *Configuring Alfabet with Alfabet Expand*.

Indicator types can be implemented in the following contexts:

- in diagram views, which allow indicators to be displayed for objects in business support maps and diagrams. In this case, the indicator type must be assigned to an evaluation type, which must be assigned to the same object class that the diagram view is assigned to. For more information, see the section Configuring Diagram Views for Diagrams.
- in compliance controls, whereby the indicator type serves as a metric to evaluate compliance issues for architecture elements. In this case, the indicator type must be assigned to an evaluation type, which must be assigned to the object class Compliance Control. The indicator types can then be assigned to the relevant compliance control in the **Compliance Control Set** editor. For more information about configuring the compliance management capability, see the section *Configuring*

the Compliance Management Capability in the reference manual *Configuring Alfabet with Alfabet Expand*.

- in business appraisals, which allow an evaluation of the as-is and to-be support of the business to be assessed. In this case, the indicator type must be assigned to an evaluation type, which must be assigned to the object class Business Appraisal. The indicator types must also be specified in the XML object ***ITMapDef***. For more information about configuring indicator types for business appraisals, see the section *Configuring Standard Business Support Matrices* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- in a configured object cockpit. For more information about configuring object cockpits, see the section *Configuring Object Cockpits for a Custom Object View* in the chapter *Configuring Object Views* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- in configured reports. Indicator types can be used in Alfabet queries and native SQL queries written for configured reports. For more information about configuring reports displaying indicator types, see the chapter *Configuring Reports* in the reference manual *Configuring Alfabet with Alfabet Expand*.

The **Evaluation & Portfolios** functionality allows you to configure the evaluation data necessary to use the evaluation functionalities available in Alfabet. In this functionality, you can create indicator types and evaluation types for the objects that will be evaluated. You can further specify computation rules that can be used to automatically calculate one or more indicator types. Additionally, you can create and configured prioritization schemes and portfolios.

Once the evaluation data has been configured, it must be assigned to the object classes for which they are relevant in the **Class Configuration** functionality.

The following information is available:

Configuring Evaluation Types

An evaluation type bundles one or more indicator types for use in the evaluation of a specific dimension of an object's performance. Users define an indicator for each indicator type bundled in the evaluation. For example, typical evaluation types could be Complexity and Standardization Status. The evaluation type Complexity might have the indicator types Number of Interfaces, Number of Modules, and State of the Art, and the evaluation type Standardization Status might have one indicator type also named Standardization Status.

An evaluation type may be configured once and reused for multiple object classes. In this way, all system-related classes such as Application, Component, and Standard Platform, for example, could be evaluated for their complexity, standardization, etc.

Evaluation types can also be grouped into prioritization schemes that, in turn, may be implemented as axes in portfolios.

Access to the evaluation types used in Alfabet evaluations can be controlled via user profiles. A user can view any evaluation type that has either no user profile defined or the same user profile as that which he/she is currently logged in with.

For information about the object classes for which evaluation types can be configured, see Appendix: Class Configuration for Object Classes. For an overview of the contexts in which evaluation types are implemented, see the section Configuring Evaluations, Prioritization Schemes, and Portfolios.

The following provides an overview of the configuration steps necessary so that evaluation types can be defined by users in the *Evaluation Page View*. For other possibilities to implement evaluation types and indicator types, define the evaluation data as described below and refer to the section Configuring Evaluations, Prioritization Schemes, and Portfolios to find the information required to implement evaluation types and indicator types in other contexts in Alfabet.

- Create a new evaluation type in the **Evaluation Types** page view in the **Evaluations and Portfolios** functionality.
- Create the indicator types that are relevant for the evaluation type in the **Indicator Types** page view in the **Evaluations and Portfolios** functionality.
- If necessary, define computation rules for indicator types in the **Computation Rules** page view for the relevant indicator type in the **Evaluations and Portfolios** functionality. For more information, see the section Specifying Computation Rules for Indicator Types.
- Assign the evaluation type in the **Evaluation Types** page view of the relevant object class(es) or object class stereotype(s) to that object class/object class stereotype in the **Class Configuration** functionality. Please note the following:
  - If an evaluation type is assigned to an object class, any object class stereotypes based on that object class will inherit the evaluation types if no evaluation types are explicitly assigned to the object class stereotype in the **Evaluation Types** page view in the **Class Configuration** functionality.
  - If the evaluation type is to be implemented as an aspect evaluation type, it should instead be assigned to the classes Application and/or Component in the **Aspect Evaluation Types** page view in the **Class Configuration** functionality. For more information, see the section Assigning Aspect Evaluations Types to the Object Classes Application and Component.
  - If the evaluation type is to be implemented as a project evaluation type, it should instead be assigned to the class Project in the **Project Evaluation Types** page view in the **Class Configuration** functionality. For more information, see the section Configuring Project Evaluation Types.
  - If the evaluation type is to be implemented as a time series evaluation type, it should instead be assigned to the relevant object class in the **Time Series Evaluation Types** page view in the **Class Configuration** functionality. For more information, see the section Creating a Time Series Period.

The following information is available:

- Creating an Evaluation Type
- Deleting an Evaluation Type
- Configuring Indicator Types for an Evaluation Type
  - Creating New Indicator Types for the Selected Evaluation Type

Creating an Evaluation Type

An evaluation type is a grouping of one or more indicator types that are used to evaluate objects (like applications, business processes, etc.) in Alfabet. You must first create the evaluation type before you can assign indicator types to it.

Once objects have been evaluated based on an evaluation type, the evaluation type name should NOT be changed.

Please note that a service is offered by Software AG that allows erroneous indicator type names to be corrected for objects in the database. For more information about renaming such indicator types, please contact Software AG Support.

1) Go to the **Evaluations and Portfolios** functionality and click **Evaluation Types**.
2) In the **Evaluation Types** page view, click **New** > **Create New Evaluation Type**. The **Evaluation Type** editor opens.
3) Enter the data, as needed. Each field is defined below:

   Basic Data tab:

   - **Name**: Enter a unique name for the evaluation type.
   - **Short Name**: Enter an abbreviated name for the evaluation type.
   - **Technical Name**: Enter a unique technical name for the evaluation type.

- **Description**: If necessary, enter a meaningful description that will clarify the purpose of the evaluation type.

Visualization tab:

- **Icon Gallery**: Select an icon gallery to visualize the evaluation type. The values of the evaluation type will be displayed as icons in the relevant view. If you do not define an icon gallery, numeric values will be displayed.

Custom icons are added to an icon gallery in the configuration tool Alfabet Expand. For more information, see the section *Adding and Maintaining Icons for the Alfabet Interface* in the reference manual *Configuring Alfabet with Alfabet Expand*.

For more information about additional configuration required to display evaluation types in configured reports, see the chapter *Configuring Reports* reference manual *Configuring Alfabet with Alfabet Expand*.

User Profiles tab:

- **User Profiles**: Select none, one, or more user profiles that may have access to this evaluation type. A user can view any evaluation type with either no user profile or the same user profile as that which he/she is currently logged in with.

4) Click **OK** to save the evaluation type or **Cancel** if you do not want to save it. The new evaluation type appears in the table.

Deleting an Evaluation Type

If you delete an object in the object class Evaluation Type, it will be irrevocably deleted from the Alfabet database. If any of the following dependent objects or data is defined for the deleted object, these will also be deleted: Indicator Type, Evaluation.

To delete an evaluation type:

1) In the table, click the evaluation type that you want to delete.
2) In the toolbar, click the **Delete**  button.
3) Confirm the warning by clicking **Yes,** or click **No** to exit without deleting the selected object(s).

Configuring Indicator Types for an Evaluation Type

For an overview of how indicator types can be implemented in Alfabet, see the section Configuring Evaluations, Prioritization Schemes, and Portfolios.

When you define an indicator type, you must decide if users will manually enter values for each indicator type or select a value from a preconfigured set of values, or if the values are to be computed by means of a computation rule. If the indicators are to be explicitly entered by a user, then you only need to define a name and description for the indicator type. If indicators are to be selected by the user from a

preconfigured set of values, then you should define the permissible values in the **Value Range** attribute for the indicator type. One of the values defined in the value range should be specified as a default value.

Please note that users may define a negative indicator value for an indicator type for indicators based on real values. This is not the case for indicators that are defined for an indicator type for which the **Range** attribute has been defined. Only non-negative integers (greater than or equal zero) may be specified for the **Range** attribute. It is possible to use any characters or strings in the semantic value part of the **Range** definition and to set the **Hide Numbers** attribute to `True` in the editor for the indicator type.

If the indicator rule is to be calculated via a computation rule, you must first create the indicator type as explained below and then define the relevant computation rule(s) that apply to the indicator type. Computation rules may range from simple calculation based the calculation of object references or aggregation of indicator values to more complex computation rules in which multiple indicators are computed by means of queries. All indicator types required for the computation rule must first be defined in order to create the computation rule. For more information about the configuration of computation rules, see the section Specifying Computation Rules for Indicator Types.

Once objects have been evaluated based on an indicator type, the indicator type definition should NOT be changed.

Please note that a service is offered by Software AG that allows erroneous indicator type names to be corrected for objects in the database. For more information about renaming such indicator types, please contact Software AG Support.

The following information is available:

- Creating New Indicator Types for the Selected Evaluation Type
- Moving Existing Indicator Types to the Selected Evaluation Type

Creating New Indicator Types for the Selected Evaluation Type

To create an indicator type:

1) In the **Evaluation Types** page view, click the evaluation type that you want to define an indicator type for and click the **Navigate**  button. The **Indicators** page view opens. The table lists indicator types that are already assigned to the selected evaluation type.
2) In the toolbar, click **New** > **Create New Indicator Type**. The **Indicator Type** editor opens.
3) Enter the information in the fields, as needed. Each field is defined below:

   Basic Data tab:

   - **Name**: Enter a unique name for the indicator type. If only one indicator type is assigned to the evaluation type, it is recommended that the indicator type name is the same as the evaluation type name. If the name has more than 35 characters and will be assigned to diagram view, the name will be truncated in the diagram legend.
   - **Short Name**: Enter up to 5 characters for a meaningful short name for the indicator type. The short name may be displayed in some Alfabet reports as a column header for the indicator.

- **Range**: If a predefined set of values is to be available for the user to select, enter the valid values for manually-entered indicators according to the following convention: <NumericValue>-<SemanticValue> The default displayed in the **Range** field is: 1 - low, 2 - medium, 3 - high. This can be changed as needed.

  In the following range definition, the numeric values are 1, 2, 3, and 4 and the semantic values are Number 0 - 1, Number 2 - 4, Number 5-10, and Number < 10.

  ```
  1 - Number 0 - 1
  2 - Number 2 - 4
  3 - Number 5 - 10
  4 -Number < 10
  ```

  Please keep the following in mind:

  - Each value must be on a separate line.
  - Ensure that an empty space exists before and after the dash.
  - Only non-negative integers (greater than or equal zero) may be specified.
  - The text you enter for each range value in the **Range** field may not exceed 128 characters.
  - Leave the **Range** field empty if the indicator is to be explicitly entered by the user or computed via a computation rule.
  - The tooltip displayed in a portfolio will display the semantic value of an indicator if this has been configured. If no semantic value is available, the numeric value will be displayed.
  - Translation is supported for the range definitions for indicator types. For more information about the translation of the range definition, see the section *Understanding the Configuration Required to Translate Semantic Indicators* in the chapter *Localization and Multi-Language Support for the Alfabet Interface*in the reference manual *Configuring Alfabet with Alfabet Expand*.
  - It is highly recommended that you do **NOT** edit the values defined for the **Range** attribute once existing Alfabet objects have already been evaluated. If you determine that this is necessary, the existing indicators must be migrated to align with the new values defined for the **Range** attribute. Please contact Software AG Support about the required script required for such a migration.

- **Default Value**: Enter a default value to set for the indicator for new objects. If the **Range** field is defined, then you should enter a valid value that is defined in the **Range** field. Leave the **Default Value** field empty if the indicator is to be computed via a computation rule.

  Please note that the default value will not be applied automatically to objects but rather must be set explicitly by a user for a selected object via the **Set Undefined Values to Default** option in the *Evaluation Page View*.

- **Hide Numbers**: Set a checkmark to hide the numeric values for the indicators based on indicator types defined in the **Range** field. The semantic value defined for the values in the **Range** field will be displayed in standard Alfabet views, editors, and configured reports instead of the numeric value. The following convention is used to define values in the **Range** field: <NumericValue>-<SemanticValue> (for example, 1-low, 2-medium, 3-high).

- **Generation**: This attribute is used to determine the calculation order of multiple computation rules defined for an indicator type. This attribute allows you to define the level of depth and dependency that the indicator type has on other indicator computations in the computation hierarchy. Defining the **Generation** attribute will improve the performance of complex computations that are dependent on the computation of other indicators.

  Enter 0 if this indicator type does not require other indicators to be computed. Enter 1 if this indicator type is dependent on the generation of the next level of indicator types in the computation hierarchy. Enter n+1 for indicators that require computed indicators of generation n.

  Please note that a computational rule for an object class cannot reference an indicator for the same object class if that indicator is the result of a computation rule. All indicators based on computation rules are created simultaneously and therefore one indicator cannot be based on the result of the calculation of another indicator for the same object. If you require this type of computation, it is recommended that you define a computational rule using a native SQL query.

- **Description**: Enter a meaningful description that will clarify the purpose of the indicator type. The indicator description will be displayed as a read-only field in the **Indicator** editor available in the *Evaluation Page View*.
- **Technical Name**: Enter a unique technical name for the indicator type.

Visualization tab:

- **Icon Gallery**: Select an icon gallery to visualize indicators by means of an icon. The icon will be displayed in relevant Alfabet views instead of the numeric value.

  Once you have selected an icon gallery, the table below will display additional cells that must be defined.

  Custom icons are added to an icon gallery in the configuration tool Alfabet Expand. For more information, see the section *Adding and Maintaining Icons for the Alfabet Interface* in the reference manual *Configuring Alfabet with Alfabet Expand*.

  In the **Min. Value** and **Max. Value** cells, enter a number for the minimum and maximum value that the icon represents. You can leave either the **Min. Value** or **Max. Value** cells empty in order to include everything below or above a certain value.

  The minimum and maximum values that you enter should correspond to the numeric values defined in the **Range** field in the **Basic Data** tab. If there is overlap in the definition of the value ranges, then the first range that matches an indicator value will be used to determine the icon to display.

  In the **Caption** cell, enter a caption as short tooltip text that explains the numeric value represented by the icon when the user moves the mouse over the indicator (for example, Very Critical, Critical, Not Critical). If you do not enter a caption, the numeric values defined in the **Min. Value** and **Max. Value** will be displayed.

  The minimum and maximum values that you enter should correspond to the values defined in the **Range** field in the **Basic Data** tab. If there is overlap in the definition of the value ranges, then the first range that matches an indicator value will be used to determine the icon to display.

4) Click **OK** to save the indicator type or **Cancel** if you do not want to save it.

Moving Existing Indicator Types to the Selected Evaluation Type

To move an indicator type:

1) In the **Evaluation Types** page view, click the evaluation type that you want to define an indicator type for and click the **Navigate**  button. The **Indicators** page view opens. The table lists indicator types that are already assigned to the selected evaluation type.
2) In the toolbar, click **New** > **Move Exising Indicator Type Here**. The selector opens.
3) Select the indicator type to move to the selected evaluation type. The **Browse** tab displays the indicator types structured according to the evaluation types they are assigned to.
4) Click **OK** to move the indicator type to the selected evaluation type. The indicator type will be removed from the original evaluation type.

Specifying Computation Rules for Indicator Types

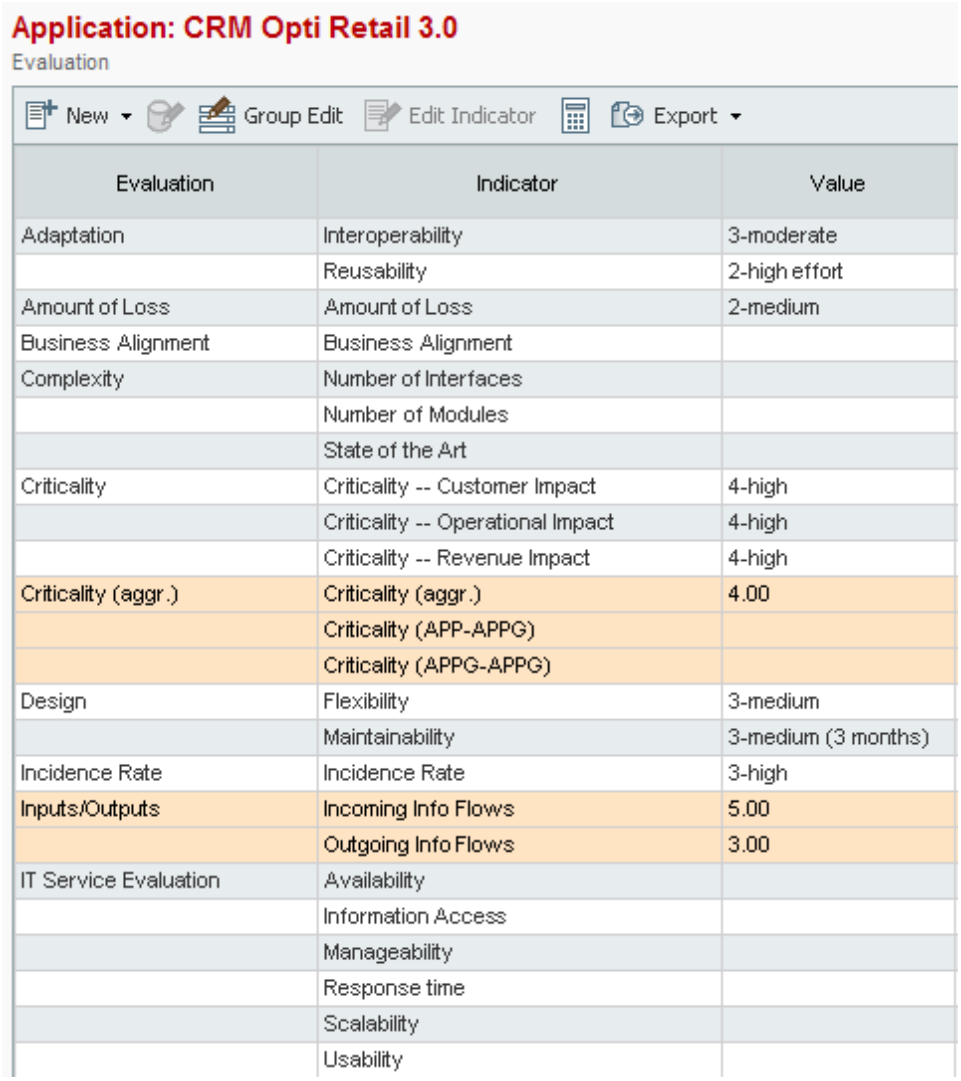
A computation rule allows numeric indicators to be automatically calculated for objects in a specific object class. In order to define the computation logic, the user must define a base class and an associated attribute of the type `Integer`, `Real`, `Reference`, or `ReferenceArray` that is to be computed. The indicator type to be calculated must be assigned to the computation rule. A scaling scheme can be applied to the computation rule, if needed.

The following can be defined by means of a computation rule:

An indicator rule may have multiple computation rules defined. Each computation rule must have a different base class defined. For example, an indicator type Adaptability might have a computation rule that assesses the adaptability of applications, another that assesses the adaptability of components, another that assesses local components, and one that assesses solution applications.

In order for the indicator to be calculated and displayed in the *Evaluation Page View* for an application, component, local component, and solution application, the evaluation type that the adaptability indicator is assigned to must be assigned to the classes Application, Component, Local Component, and Solution Application in the **Class Configuration** functionality.

Computation rules can be defined for object classes that represent artifacts in Alfabet as well as the classes Business Case and Platform Element.



Application: CRM Opti Retail 3.0

Evaluation

New | Group Edit | Edit Indicator | Export

Evaluation	Indicator	Value
Adaptation	Interoperability	3-moderate
	Reusability	2-high effort
Amount of Loss	Amount of Loss	2-medium
Business Alignment	Business Alignment	
Complexity	Number of Interfaces	
	Number of Modules	
	State of the Art	
Criticality	Criticality -- Customer Impact	4-high
	Criticality -- Operational Impact	4-high
	Criticality -- Revenue Impact	4-high
Criticality (aggr.)	Criticality (aggr.)	4.00
	Criticality (APP-APPG)	
	Criticality (APPG-APPG)	
Design	Flexibility	3-medium
	Maintainability	3-medium (3 months)
Incidence Rate	Incidence Rate	3-high
Inputs/Outputs	Incoming Info Flows	5.00
	Outgoing Info Flows	3.00
IT Service Evaluation	Availability	
	Information Access	
	Manageability	
	Response time	
	Scalability	
	Usability	

FIGURE: Example of indicators generated via computation rules (highlighted) in the Evaluation page view

Indicators based on computation rules are highlighted for users in the *Evaluation Page View*. The computation rules can be executed and updated by a user by clicking the **Calculate Indicators** button in this view. All indicators based on computation rules will be updated in the *Evaluation Page View* as well as in all other page views with evaluations, prioritization schemes, portfolios, and diagram views containing the indicator in the context of the selected object.

It is recommended that your enterprise compute such indicators at regular intervals via a batch process. For more information about configuring a batch process for evaluations, see the section *Batch-Calculation of Indicators with RescanIndicatorsConsole.exe* in the reference manual *System Administration*.

Please note the following issues regarding the implementation of computation rules.

- The **Range** attribute for the indicator type should **NOT** be defined if a computation rule is to be defined for the indicator type. If default values exist in the **Range** field in the **Indicator Type** editor, you must remove them.
- When defining a computation rule for an indicator type, the associated evaluation type must be assigned to same object class that is specified as the base class of the

computation rule. For more information, see Assigning Evaluations Types to an Object Class in the **Class Configuration** functionality.

- The **Generation** attribute in the **Indicator Type** editor is used to determine the calculation order of computation rules defined for indicator types. This attribute allows you to define the level of depth and dependency on other indicator computations for an indicator type in the computation hierarchy. Defining the **Generation** attribute will improve the performance of complex computations that are dependent on the computation of other indicators.

  Please note, however, that a computational rule for an object class cannot reference an indicator for the same object class if that indicator is the result of a computation rule. All indicators based on computation rules are created simultaneously and therefore one indicator cannot be based on the result of the calculation of another indicator for the same object. If you require this type of computation, it is recommended that you define a computational rule using native SQL.

- Indicators based on computation rules are evaluated by Alfabet in real time if the **Calculate Indicators** button is used or if the indicators are displayed in standard portfolio views. Whereas this has the benefit of evaluating the most recent data, performance may be compromised, particularly when aggregations are performed across multiple hierarchy levels (for example, from a component to an application to the application group and further up the application group hierarchy). Depending on the depth of an aggregation hierarchy, the recalculation of aggregated indicators may take up to several minutes.

- The definition of evaluation types, indicator types, computation rules, and the assignment of icon galleries to indicator types are cached in Alfabet to provide better performance. Consequently, any changes to these definitions will result in a few minutes delay before they are available to the user community.

- The reference manual *Alfabet Meta-Model* provides an overview of all relevant object classes in the meta-model, their object class properties, as well as the object classes that reference the object class for which you are seeking results.

If you edit an existing computation rule, you must restart the Alfabet Windows Client if you want to immediately implement the modified computation rule.

The following information is available regarding the definition of computation rules:

- Defining a Computation Rule to Calculate an Object's Attribute of the Type Reference or ReferenceArray
- Defining a Computation Rule to Compute an Object Attribute of the Type Real or Integer
- Defining a Computation Rule to Calculate Indicators
  - Calculating Indicator Types Having a Range Definition
  - Defining a Computation Rule to Aggregate Indicators
  - Defining a Computation Rule Using a Query to Calculate an Indicator Type
- Defining a Computation Rule That Uses the Object and Indicator Computation Methods
- Defining a Code-Based Computation Rule

- Scaling the Indicators Generated by a Computation Rule
- Detaching an Indicator Type from a Computation Rule
- Deleting a Computation Rule

Defining a Computation Rule to Calculate an Object's Attribute of the Type Reference or ReferenceArray

A computation rule can be configured that computes a numeric indicator for any attribute that is not a numeric attribute. In this way, for example, you can calculate the sum or average of the number of objects assigned via an attribute of the type `Reference` or `ReferenceArray`. By using the **Object Computation Method** available for computation rules, you can configure a computation rule that calculates the sum, minimum, maximum, or average value of the computed objects or that is based on an Alfabet query or native SQL query that specifies the computation.

The following examples use computation rules based on the **Object Computation Method** in order to calculate the number of object references:

- Example 1: The enterprise wants to know the number of business services that are assigned to an application's local components and reused by an application. The evaluation type Routed Business Services has been created and the indicator Routed Business Services has been created for the evaluation type. The indicator type requires the following computation rule:
  - **Base Class**: Application (this is the class for which you are seeking the result)
  - **Property**: Routed Services (a property of the type `ReferenceArray`)
  - **Object Computation Method**: `Count`

    The evaluation type must be assigned to the class Application. The resulting indicator will be an integer determined by the number of all business services assigned to the local components of the application that are reused by the application.
- Example 2: The enterprise wants to know the total number of incoming and outgoing information flows for an application. The evaluation type Information Flows has been created and the indicator Sum of Information Flows has been created for the evaluation type. Because the class Application does not have an information flow attribute, a query must be defined to find an application's incoming and outgoing information flows by means of the relevant attributes on the class Information Flow.
  - **Base Class**: Application
  - **Property**: leave this field empty
  - **Object Computation Method**: `Count`
  - **Queries** tab. Because there is no attribute to define, you must define an Alfabet query or SQL query: The query parameter:`BASE` references the base class defined in the **Base Class** field. The following query finds the information flows for which a selected application is the incoming or outgoing application.

```
<Queries>
    <Query Query="ALFABET_QUERY_500
```



```
FIND Application
InnerJoin InformationFlow ON
    (OR
    InformationFlow.From = Application.REFSTR
    InformationFlow.To = Application.REFSTR
    InformationFlow.ToOwner = Application.REFSTR
    InformationFlow.FromOwner = Application.REFSTR
    )
WHERE Application.RefStr =:BASE"/>
</Queries>
```

The evaluation type must be assigned to the class Application. The resulting indicator will be the number of all incoming and outgoing information flows for which the application is a source or target.

To create a computation rule that calculates object references:

1) To create a computation rule, click **New** > **Create New Computation Rule**. The **Computation Rule** editor opens.
2) In the **Name** field, enter a unique name for the computation rule.
3) In the **Base Class** field, select the class for which you are seeking the result.

For a list of all object classes that can be defined as the base class of a computation rule, see Appendix: Class Configuration for Object Classes in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

4) Next, you must define the referenced object class that is to be evaluated by means of the computation rule:

- To specify the referenced objects via an attribute of the base class, select a attribute of the type `Reference` or `ReferenceArray` in the **Property** field.

Please note that the **Property** field displays all custom attributes of the types `Integer`, `Real`, `Reference`, `ReferenceArray` and `String` that have been configured for the base class.

- To specify the referenced objects via a query, enter an Alfabet query or SQL query in the **Queries** tab.

Please keep the following in mind regarding the definition of queries:

- The query replaces the definition specified in the **Property** field. Therefore, the **Property** field should not be defined in the **Computation Rule** editor. The query parameter:`BASE` must reference the base class defined in the **Base Class** field.
- `SHOW` properties cannot be defined for an Alfabet query used in a computation rule.
- A computation rule may consist of multiple queries.

- Alfabet queries may first be written in the tool Alfabet Expand using the Query Builder and then copied and pasted into the **Queries** tab in the **Computation Rule** editor.
- For more information about defining Alfabet queries and the special rules that apply to the use of native SQL queries for Alfabet configurations, see *Defining Queries* in the reference manual *Configuring Alfabet with Alfabet Expand*.

5) In the **Object Computation Method** field, select the relevant means to calculate the referenced objects:

- `Sum`: Returns the sum value of all indicator values.
- `Min`: Returns the minimum value of all referenced objects.
- `Max`: Returns the maximum value of all referenced objects.
- `Avg`: Returns the average value of all referenced objects.
- `Count`: Returns the number of objects referenced by the attribute.
- `PropertyValue`: Returns a value that is computed based on the numerical value defined for the attribute for the objects in the base class. The attribute defined in the **Property** field must be a numerical type. (For example, the standard compliance of applications can be evaluated by means of the standard compliance of the components assigned to the applications.) This is described in more detail in the section Defining a Computation Rule to Compute an Object Attribute of the Type Real or Integer.

6) In the **Description** field, enter a meaningful description that will clarify the purpose of the computation rule.

7) After you have defined the computation rule, click **OK** to save the computation rule.

You must assign the evaluation type that the computation rule is associated with to the object class that is defined in the **Base Class** field in the **Computation Rule** editor. This is done in the **Class Configuration** functionality. For more information about assigning an evaluation type to an object class, see Assigning Evaluations Types to an Object Class.

If necessary, you can define scaling values for the computation rule. For more information, see Scaling the Indicators Generated by a Computation Rule.

Defining a Computation Rule to Compute an Object Attribute of the Type Real or Integer

A computation rule can be configured to display the numeric value for a standard or custom attribute of the type `Real` or `Integer`. The number defined for the attribute will be displayed as an indicator.

The enterprise wants to display an indicator representing the number of the maximum allowed time for recovery in hours for an application. A custom attribute of the type `Real` named RecoveryTimeObjective has been created for the class Application. The evaluation type Application Statistics has been created and the indicator Application Statistics has been created for the evaluation type. The indicator type requires the following computation rule:

- **Base Class**: Application (this is the class for which you are seeking the result)
- **Object Computation Method**: `PropertyValue`

- **Property**. RecoveryTimeObjective

  The evaluation type must be assigned to the class Application. The resulting indicator will be the number of hours allowed for recovery of an application in case of a disruption or break down.

To create a computation rule that displays the numeric value of a standard or custom attribute:

1) To create a computation rule, click **New** > **Create New Computation Rule**. The **Computation Rule** editor opens.
2) In the **Name** field, enter a unique name for the computation rule.
3) In the **Base Class** field, select the class for which you are seeking the result.

   For a list of all object classes that can be defined as the base class of a computation rule, see Appendix: Class Configuration for Object Classes in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

4) In the **Property** field, specify the attribute of the type `Real` or `Integer`.

   Please note that the **Property** field displays all custom attributes of the types `Integer`, `Real`, `Reference`, `ReferenceArray` and `String` that have been configured for the base class.

5) In the **Object Computation Method** field, select `PropertyValue`.
6) In the **Description** field, enter a meaningful description that will clarify the purpose of the computation rule.
7) After you have defined the computation rule, click **OK** to save the computation rule.

You must assign the evaluation type that the computation rule is associated with to the object class that is defined in the **Base Class** field in the **Computation Rule** editor. This is done in the **Class Configuration** functionality. For more information about assigning an evaluation type to an object class, see Assigning Evaluations Types to an Object Class.

If necessary, you can define scaling values for the computation rule. For more information, see Scaling the Indicators Generated by a Computation Rule.

Defining a Computation Rule to Calculate Indicators

A computation rule can be configured to calculate multiple indicator types to produce an overall indicator. By using the **Indicator Computation Method** available for computation rules, you can configure a computation rule that calculates the sum, minimum, maximum, or average value of the computed indicators or that is based on an Alfabet query or native SQL query that specifies the computation.

A computation rule must be defined for each object class that is to be evaluated in order to produce the overall indicator. The indicator types that are calculated via the computation must be assigned to the computation rule. These indicator types must also be assigned to evaluation type that is associated with the computation rule.

The general definition of a computation rule is described below. Detailed examples about configuring various kinds of computation rules using indicator types is also provided:

- Calculating Indicator Types Having a Range Definition
- Defining a Computation Rule to Aggregate Indicators
- Defining a Computation Rule Using a Query to Calculate an Indicator Type

1) To create a computation rule, click **New** > **Create New Computation Rule**. The **Computation Rule** editor opens.
2) In the **Name** field, enter a unique name for the computation rule.
3) In the **Base Class** field, select the class for which you are seeking the result.

For a list of all object classes that can be defined as the base class of a computation rule, see Appendix: Class Configuration for Object Classes in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

4) Select the method to compute the indicators generated for the indicator type for the base class in the **Indicator Computation Method** field. This setting applies if multiple indicator types are to be attached to the rule. Your choices include:

If indicator types are attached to a rule and no object computation method is defined, the method `Avg` is used by default. For more information about defining a computation rule that aggregates indicators, see the section Defining a Computation Rule to Aggregate Indicators.

- `Sum`: Returns the sum value of all indicator values.
- `Min`: Returns the minimum value of all indicators.
- `Max`: Returns the maximum value of all indicators.
- `Avg`: Returns the average value of all indicators.
- `ValueByQuery`: Select to define an Alfabet query or SQL query with a result of exactly one row and one column. The query is specified in the **Queries** tab. The use of a query in the context of the **Indicator Computation Method** is explained in more detail in the section Defining a Computation Rule Using a Query to Calculate an Indicator Type.

5) In the **Description** field, enter a meaningful description that will clarify the purpose of the computation rule.
6) After you have defined the computation rule, click **OK** to save the computation rule.
7) Next, you must assign the indicator types that are to be calculated to the computation rule. In the **Computation Rules** view, click **New** > **Add Indicator Type**. The object selector opens.
8) Select the relevant indicator type(s) and click **OK**. The indicator types are displayed in the **Indicator Type** column in the **Computation Rules** view.

You must ensure that the indicator types that you want to assign to the computation rule are associated with the object class defined as the base class. Therefore you must first assign the evaluation type that the computation rule is associated with to the object class that is defined in the **Base Class** field in the **Computation Rule** editor. This is done in the **Class Configuration** functionality. For more information about assigning an evaluation type and subordinated indicator types to a class, see Assigning Evaluations Types to an Object Class..

If necessary, you can define scaling values for the computation rule. For more information, see Scaling the Indicators Generated by a Computation Rule.

Calculating Indicator Types Having a Range Definition

A computation rule can be configured that calculates an overall indicator for multiple indicator types that have the **Range** attribute defined.

Example 1: The enterprise wants to assess the adaptability of their applications and components. The evaluation type Adaptability has been created and requires two computation rules, one to assess adaptability for the base class Application and one for the base class Component. The computation rule will compute the average value for 3 indicators in order to calculate one overall indicator. The 3 indicators that must be manually defined by the relevant users based on a predefined range of values. The three indicator types and their defined values ranges are:

Exposed and Reusable Services	Function Points	Level of Standardization
• 1-no functionality exposed (0%) • 2-some parts of functionality exposed (1%-30%) • 3-major parts of functionality exposed (31%-80%) • 4-nearly all functionality exposed (>80%)	• 1-more than 300 • 2-100 to 299 • 3-30 to 99 • 4-less than 30	• 1-in house developed • 2-standard, but customized • 3-standard, but configured • 4-standard, as is

A fourth indicator type must be created for the evaluation type. This fourth indicator type will have the computation rule definition. The computation rule for the class Application will be defined as follows:

- **Base Class**: Application (this is the class for which you are seeking the result)
- **Indicator Computation Method**: `Avg`

  Additionally, the three indicators Exposed and Reusable Services, Function Points, and Level of Standardization must be assigned to the computation rule. The resulting indicator will be an integer determined by the average value defined for the three indicators Exposed and Reusable Services, Function Points, and Level of Standardization.

  In order to also assess the adaptability of components, another computation rule must be created with the same configuration except that Component must be defined for its **Base Class** attribute. The evaluation type must be assigned to both the object class Application and the object class Component.

Defining a Computation Rule to Aggregate Indicators

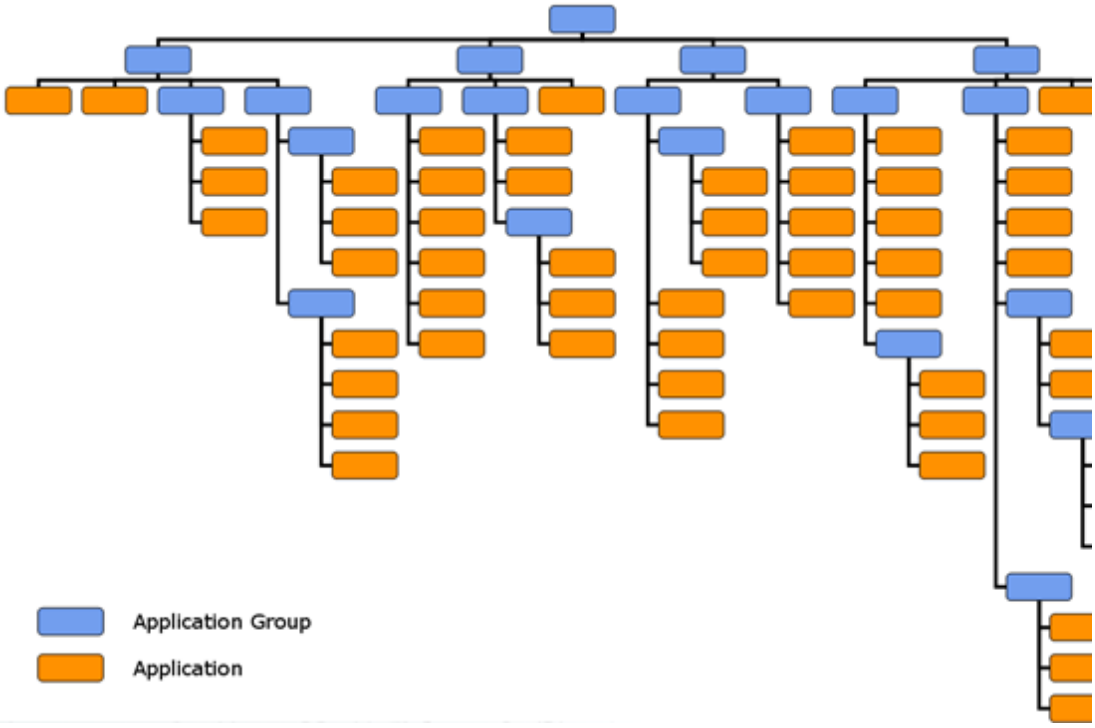
A computation rule may aggregate indicators defined for objects in an object hierarchy. A computation rule based on aggregation logic states how to aggregate the results of multiple computation rules for multiple objects in the object hierarchy in order to produce an overall indicator.

The aggregation will skip indicators that do not have a value. When computing an average, for example, the result is based on the number of objects that have an indicator.

Please note that indicators based on computation rules are evaluated by Alfabet in real time if the **Calculate Indicators** button is used or if the indicators are displayed in standard portfolio views. Whereas this has the benefit of evaluating the most recent data, performance may be compromised, particularly when aggregations are performed across multiple hierarchy levels (for example, from a component to an application to the application group and further up the application group hierarchy). Depending on the depth of an aggregation hierarchy, the recalculation of aggregated indicators may take up to several minutes.

For example, a user wants to understand the criticality of applications in an all application groups subordinate to a parent application group as well as the criticality of the application groups. This would require two indicators; one indicator provides the criticality of applications and the other provides the criticality of application groups.

To configure the computation rules, you must first understand the application group hierarchy.



As the figure above demonstrates, the hierarchy has two different types of relationships:

- applications in application groups
- subordinate application groups in parent application groups

Therefore, to produce the indicators necessary to understand the criticality of applications and the criticality of application groups, the following is required:

1) A Criticality indicator for applications. For this purpose, a computation rule should be configured that calculates the values for relevant indicator types (such as Customer Impact, Operational Impact, and Revenue Impact). In our example, these indicators have a **Range** attribute with values that users define. The computation rule would be defined as follows:

   - **Name**: Criticality Apps
   - **Base Class**: Application. The class Application is the class for which you are seeking the result.
   - **Property**: Leave this field empty.
   - **Indicator Computation**: Leave this field empty.
   - **Object Computation Method**: Leave this field empty.

     Add the relevant indicator types (Customer Impact, Operational Impact, and Revenue Impact) that have a range definition to the computation rule.

2) A Criticality indicator for application groups. For this purpose, a computation rule should be configured that computes the values for two indicators: Criticality App<AppGroup and Criticality AppGroup<AppGroup. The indicators Criticality App<AppGroup and Criticality AppGroup<AppGroup each require a computation rule of their own to calculate the overall criticality indicator.

   - The first indicator Criticality App<AppGroup requires a computation rule that calculates an indicator of each application group containing applications. The computation rule for the indicator (Criticality App<AppGroup) would be defined as follows:
     - **Name**: Criticality App<AppGroup
     - **Base Class**: Application Group. The class Application Group is the class for which you are seeking the result.
     - **Property**: Applications. (The applications in an application group)
     - **Indicator Computation Method**: `Max`
     - **Object Computation Method**: Leave this field empty.
   - The second indicator Criticality AppGroup<AppGroup requires a computation rule that calculates an indicator for each all application group containing subordinate application groups. The computation rule for the indicator (Criticality AppGroup<AppGroup) would be defined as follows:
     - **Name**: Criticality AppGroup<AppGroup
     - **Base Class**: Application Group. The class Application Group is the class for which you are seeking the result.
     - **Property**: Subordinate Groups. (The subordinate application groups in a parent application group.)
     - **Indicator Computation Method**: `Max`
     - **Object Computation Method**: Leave this field empty.

3) The evaluation type associated with the computation rules must be assigned to the object classes Application and Application Group.

Defining a Computation Rule Using a Query to Calculate an Indicator Type

Whenever the needed computation logic exceeds the build-in functionality of Alfabet for computation rules, you can use an Alfabet query or native SQL query in the computation rule to implement your logic.

Set the following in the **Computation Rule** editor to define a computation rule using a query:

1) In the **General** tab, set the following attributes:
   - **Name**: Enter a unique name for the computation rule.
   - **Base Class**: Select the class for which the indicators based on the indicator type shall be calculated.
   - **Property**: Leave this field empty.
   - **Object Computation Method**: Leave this field empty.
   - **Indicator Computation Method**: Select `ValueByQuery`.
2) In the `Queries` tab, enter an XML element `Query` with the following structure:

   ```
   <Query Name="Unique Name"
   Query="Native SQL or Alfabet Query"/>
   ```
3) Set the attributes of the XML element `Query` to the following:
   - `Name`: Enter a unique name for the query that describes the meaning of the query.
   - `Query`: Enter a native SQL or Alfabet query that returns a single numeric value in a dataset with one row and one column. The standard Alfabet query language parameter `BASE` can be used in the query to refer to the `REFSTR` of the current base object. The base class is the object class specified with the attribute **Base Class** in the **General** tab.

     If the Alfabet database is hosted on an Oracle® database server, SQL queries that shall return a real or float value must contain an according CAST statement in the `SELECT` clause. Otherwise, values will be returned as integer values.

     To test a query prior to using it in a computation rule, you can create a configured report based on the query in the configuration tool Alfabet Expand.

     For basic information about the definition of Alfabet queries and native SQL queries in the context of Alfabet configurations, see *Defining Queries* in the reference manual *Configuring Alfabet with Alfabet Expand*.

The enterprise wants to know the number of projects that impact an ICT object. The evaluation type ICT Projects has been created. Because there are no direct references between the classes ICT Objects and Projects, a query must be defined by means of a computation rule. The query must search the project architecture of all projects to find the ones where the ICT object is specified as affected by the project in the *Affected Architecture Page View* page view.

```
<Query Name="ICT Projects"
```



```
Query="SELECT COUNT(*)
FROM PROJECT p, PROJECT_ARCH pa, ICTOBJECT icto
WHERE p.STEREOTYPE = 'Project'
    AND p.REFSTR = pa.PROJECT
    AND icto.REFSTR = pa.OBJECT
    AND icto.REFSTR = @BASE
"
/>
```

The evaluation type must be assigned to the object class ICT Object. The resulting indicator will be the number of projects that an ICT object is involved in.

Defining a Computation Rule That Uses the Object and Indicator Computation Methods

It is possible to create a complex computation rule that uses both the Object Computation Method and the Indicator Computation Method.

For example, you want to generate a computation rule on objects of the class Application Group that calculates the average availability of the applications in the application group, while considering only the lowest of the three indicators Reusable Services, Function Points, and Standardization Level per application.

- **Name**: Average Adaptability over Minimum of Application Indicators
- **Base Class**: Select **Application Group**.
- **Object Computation Method**: Select `Max`. This will take the highest value of the application indicators Reusable Services, Function Points, and Standardization Level.
- **Indicator Computation Method**: Select `Avg`. This will average the resulting values computed via the **Object Computation Method** attribute to produce an overall indicator.

Defining a Code-Based Computation Rule

For more complex rules that cannot be expressed in an aggregation or Alfabet query, Software AG can support you with customer-specific code as well as an API that allows the code to be linked. In this case, please contact Software AG Support

A computation rule defined in the **Code Handler** tab has precedence over any other computation rule definition defined in the **Computation Rule** editor.

1) To create a computation rule, click **New** > **Create New Computation Rule**. The **Computation Rule** editor opens.
2) In the **Name** field, enter a unique name for the computation rule.
3) n the **Description** field, enter a meaningful description about the computation rule.
4) In the **Code Handler** tab, enter data in the following fields:

- **Assembly**: Enter the DLL provided by Software AG.
- **Assembly Class**: Enter the assembly class identified in the DLL.
- **Assembly Class Method**: Enter the method to link the DLL.

5) After you have defined the computation rule, click **OK** to save the computation rule or **Cancel** if you do not want to save it.

Scaling the Indicators Generated by a Computation Rule

The values generated in a computation rule can be scaled in order to make them easier for users to interpret. In certain situations, it may not be appropriate to display to users the indicator that is the result of a computation rule because the numbers are, for example, not comparable to other indicators or not meaningful to the user. Instead, you may want to scale such indicators to better fit with other evaluations performed with the same or related objects. In this case, the original indicators generated by the computation rule are adjusted based on a scaling scheme that you defined. The adjusted values are then displayed to users in the *Evaluation Page View* as well as other relevant views.

Scaling is available for all types of computation rules.

A user would like to assess the complexity of an application based on the number of local components associated with the application. For example, one application has 5 local components, one has 15 local components, and another has 25 local components. These numeric values should be expressed in a scaling scheme that is easier for users to interpret. Therefore, a scaling scheme should be configured whereby applications with more than 20 local components would be assigned a scaled value of 3 (highly risky), applications with between 5 and 20 local components would be assigned a value of 2 (moderately risky), and applications with less than 5 local components would be assigned a value of 0 (less risky). The scaling scheme would be defined as follows:

Result	Min Value	Max Value
3.00	20.00	50.00
2.00	5.00	19.00
1.00	0.00	4.00

The definitions in the **Base Class** and **Property** fields in the **General** tab determine the value for a specific object in the base class. The value used to compute the scaling entry is: MinValue<=Value<=MaxValue.

To define the scaling for a computation rule:

1) In the **Scaling** tab in the **Computation Rule** editor, enter the relevant scaling values in the cells in the **Result** column. To do so, click in the cell to make it active and enter the data.
2) For each value in the **Result** column, enter the minimum value in the **Min. Value** column and the maximum value in the **Max. Value** column.

   If the fields **Min. Value** and **Max. Value** are left blank for a value in the **Result** column, this will be used as the default value and will be assigned to objects that otherwise don't

meet any of the scaling conditions. The value in the **Result** column will be the indicator for the specific object that has been assessed.

3) After you have defined the scaling values, click **OK** to save the computation rule or **Cancel** if you do not want to save it.

Detaching an Indicator Type from a Computation Rule

To detach an indicator type from a computation rule:

1) Click the indicator type you want to remove from the computation rule.
2) In the toolbar, click the **Detach** button.
3) Confirm the warning by clicking **Yes or** click **No** to exit without saving your changes.

The indicator type is removed from the computation rule but will remain in the Alfabet database.

Deleting a Computation Rule

If you delete an object in the object class Computation Rule, it will be irrevocably deleted from the Alfabet database.

1) In the table, select the computation rule you want to delete.
2) In the toolbar, click the **Delete** button.
3) Confirm the warning by clicking **Yes,** or click **No** to exit without deleting the selected object(s).

Assigning Evaluation Types to an Object Class

Evaluation types must be assigned to the object class or object class stereotype that they are relevant for in order to be displayed in the *Evaluation Page View* as well as other relevant page views available for the object class. The **Evaluation** page view will display the evaluation types along with their indicator types. Indicators can be defined for the indicator types that have been configured to be manually defined.

Please note the following:

- If an evaluation type is assigned to an object class, any object class stereotypes based on that object class will inherit the evaluation types if no evaluation types are explicitly assigned to the object class stereotype in the **Evaluation Types** page view in the **Class Configuration** functionality.
- If computation rules are defined for the indicator types assigned to an evaluation type, then the evaluation type must be assigned to all object classes/object class stereotypes that have been specified in the **Base Class** attribute of the computation rule. For more information about the configuration of computation rules, see the section Specifying Computation Rules for Indicator Types.

1) Go to the **Class Configuration** functionality in the **Configuration** module and click the relevant object class or object class stereotype in the **Class Configuration** explorer.
2) Click the **Evaluation Types** page view to open it.
3) Click **New** > **Add Evaluation Type** to assign the evaluation types that are relevant for the selected class.
4) In the object selector, select the evaluation type(s) and click **OK**. The evaluation type will be available in the *Evaluation Page View* as well as other relevant page views available for the object class.

Configuring Prioritization Schemes

A prioritization scheme is a weighted composite of a set of evaluation types. A prioritization scheme is typically used to determine a prioritized ranking for a set of objects, for instance during budgeting and prioritization of projects or to define the axis of a portfolio.

Prioritization schemes aim at aggregating evaluations along complementary evaluation dimensions. Whereas evaluation types are aggregates of indicators for semantically related performance measurements of objects, prioritization schemes are typically aggregates of semantically-unrelated (or loosely related) performance measurements for objects. However, prioritization schemes are assigned to a portfolio usually describe performance along dependent dimensions.

After a prioritization scheme is created and evaluation types are assigned to it, you must specify the weighting of the evaluation types that are assigned to the prioritization scheme. The value for a prioritization scheme is calculated as the sum of the indicators divided by the sum of the weight. The sum of the weights must equal 100 for each prioritization scheme.

The formula is displayed below:

$$W = \frac{\sum_{i=1}^{n} k_i g_i}{\sum_{i=1}^{n} g_i}$$

- i=index of the evaluation
- k=evaluations (the average of the underlying indicators)
- g=weighting

Prioritization schemes may be used in any of the portfolios listed in Appendix: Class Configuration for Object Classes. Prioritization schemes are also relevant for the following page view:

- *Application Evaluation Report* (available on the root node of the *Applications by Group Explorer*)

- *Application Evaluation Report Page View* in the object profile for application groups
- *Application Evaluation Overview Page View* in the object profile for business functions
- *Application Evaluation Report Page View* in the object profile for domains
- *Application Evaluation Overview Page View* in the object profile for organizations
- *Prioritization Page View* in the object profile for project groups

The following provides an overview of the configuration steps necessary so that prioritization schemes can be displayed in the relevant views as well as configured for portfolios.

- Create the relevant evaluation types and indicator types in the **Evaluations and Portfolios** functionality. For more information, see the section Configuring Evaluation Types.
- Create the prioritization scheme in the **Prioritization Schemes** page view in the **Evaluations and Portfolios** functionality.
- Assign the evaluation types to the prioritization scheme in the **Evaluation Types** page view of the relevant prioritization scheme. Define the weighting of each evaluation type for the selected prioritization scheme.
- Assign the prioritization scheme to the relevant object class.
- If the prioritization scheme is to be implemented as an aspect prioritization scheme, it should instead be assigned to the classes Application and/or Component in the **Aspect Prioritization Schemes** page view in the **Class Configuration** functionality. For more information, see the section Assigning Aspect Prioritization Schemes to the Object Classes Application and Component.
- If the prioritization scheme is to be used as an axis in a portfolio, see the section Configuring Portfolios. Please note that you can define captions for the value range displayed on the X- and Y-axis. Each prioritization scheme may have a different set of value range captions in the context of the portfolio in which it is implemented.

The following information is available:

- Creating a Prioritization Scheme
- Deleting a Prioritization Scheme
- Assigning Evaluations Types to the Prioritization Scheme

Creating a Prioritization Scheme

To create a prioritization scheme:

1) Go to the **Evaluations and Portfolios** functionality and click **Prioritization Schemes**.
2) In the toolbar, click **New** > **Create New Prioritization Scheme**. The **Prioritization Scheme** editor opens.
3) Enter the information into each field, as required.

- **Name**: Enter a unique name for the prioritization scheme.
- **Description**: Enter a meaningful description about the prioritization scheme.

4) Click **OK** to save the prioritization scheme or **Cancel** if you do not want to save it. The new prioritization scheme appears in the table and the explorer tree.

You must now assign the evaluation types to the prioritization scheme that you have created. For more information, see

Deleting a Prioritization Scheme

If you delete an object in the object class Prioritization Scheme, it will be irrevocably deleted from the Alfabet database. However, the evaluation types assigned to it will remain in the database.

To delete a prioritization scheme:

1) In the **Prioritization Schemes** page view, click the prioritization scheme that you want to delete.
2) In the toolbar, click the **Delete**  button.
3) Confirm the warning by clicking **Yes,** or click **No** to exit without deleting the selected object(s).

Assigning Evaluation Types to the Prioritization Scheme

You can add multiple evaluation types to the selected prioritization scheme. You must specify the weighting of each evaluation type in relation to the other evaluation types. The total weighting of all evaluation types assigned to the prioritization scheme must equal 100.

To assign evaluation types to the prioritization scheme:

1) In the **Prioritization Schemes** page view, double-click the prioritization scheme that you want to assign evaluation types to. The **Evaluation Types** page view will open.
2) In the toolbar, click **New** > **Add New Evaluation Type**.
3) In the object selector that opens, select the evaluation types that you want to assign to the prioritization scheme and click **OK**.
4) Once all evaluation types are assigned to the prioritization scheme, click the **Edit Weighting** button in the toolbar. The **Evaluation Weighting Scheme** editor opens.
5) For each evaluation type, drag the slider to adjust the weight value. Once you have specified a value for an evaluation type, click the lock symbol to the left of the evaluation type. Once the value is locked, it will not change when the weighting is being specified for another evaluation type. The weighting values for all evaluation types assigned to the prioritization scheme must equal 100. Click **OK** when all evaluation types have been weighted.

Configuring Portfolios

A portfolio is a bubble chart that represents the relative performance of a set of objects that belong to the same object class in two or three independent dimensions of measurement. Typical portfolios include a group of applications that are owned by a specific organization or used by a particular process.

The portfolio consists of an X-, Y-, and if necessary, power-axis. The power-axis is the Z-dimension and is reflected via bubble size. Either an indicator type, evaluation type, or prioritization scheme can be assigned to a portfolio axis. In Alfabet, a portfolio may also be displayed as a BCG quadrant. Access to portfolios can be controlled via user profiles. A user can view any portfolio that has either no user profile defined or the same user profile as that which he/she is currently logged in with.

The tooltip displayed in a portfolio will display the semantic value of an indicator if this has been configured. If no semantic value is available, the numeric value will be displayed.

Please note that you can define captions for the value range displayed on the X- and Y-axis. Each prioritization scheme may have a different set of value range captions in the context of the portfolio in which it is implemented.

For information about the object classes for which portfolios can be configured, see Appendix: Class Configuration for Object Classes.

The following provides an overview of the configuration steps necessary so that portfolios can be displayed in the relevant views:

- Create the relevant evaluation types and indicator types in the **Evaluations and Portfolios** functionality. For more information, see the section Configuring Evaluation Types.
- Create the relevant prioritization schemes in the **Evaluations and Portfolios** functionality. For more information, see the section Configuring Prioritization Schemes.
- Create a portfolio and assign the relevant indicator types, evaluation types, or prioritization schemes to the portfolio.
- Assign the portfolio to the relevant object class.
- If the portfolio is to be implemented as an aspect portfolio, it should instead be assigned to the classes Application and/or Component in the **Aspect Portfolios** page view in the **Class Configuration** functionality. For more information, see the section Assigning Aspect Portfolios to the Object Classes Application and Component.
- If the prioritization scheme is to be used as an axis in a portfolio, see the section Configuring Portfolios. Please note that you can define captions for the value range displayed on the X- and Y-axis. Each prioritization scheme may have a different set of value range captions in the context of the portfolio in which it is implemented.

The following information is available:

- Creating a Portfolio
- Assigning the Portfolio to an Object Class

Creating a Portfolio

To create a portfolio and define its layout:

1) Go to the **Evaluations and Portfolios** functionality and click **Portfolios**.
2) In the toolbar, click **New** > **Create Portfolio**. The **Portfolio** editor opens.
3) Enter the information into each field, as required. Each field is defined below:

   Properties tab:

- **Name**: Enter a unique name for the portfolio. The name will be displayed as an option in the **Show Portfolio** field available in the relevant portfolio reports in the Alfabet interface.
- **X-Axis**: In the **Search For** field, select either Indicator Type, Evaluation Type, or Prioritization Scheme to search for the relevant measure to place on the X-axis. The name defined for the indicator type, evaluation type, or prioritization scheme will be displayed on the axis as a caption. The caption for a prioritization scheme can be defined differently in each portfolio that it is assigned to.
- **Y-Axis**: In the **Search For** field, select either Indicator Type, Evaluation Type, or Prioritization Scheme to search for the relevant measure to place on the Y-axis. The name defined for the indicator type, evaluation type, or prioritization scheme will be displayed on the axis as a caption. The caption for a prioritization scheme can be defined differently in each portfolio that it is assigned to.
- **Power-Axis**: The power-axis does not appear as a conventional axis in the portfolio. Instead, the value of the power-axis is expressed by the size of the bubble that represents an object. Users will be able to read the object's value by placing the mouse pointer on the bubble. In the **Search For** field, select either Indicator Type, Evaluation Type, or Prioritization Scheme to search for the relevant measure to place on the power-axis (Z-dimension).
- **Quadrant Layout**: Select the checkbox if you want to display the portfolio in a quadrant layout. If you choose to display a quadrant layout, no ticks will be displayed on the axes. You can name the quadrant captions in the **Quadrants** tab.
- **Transparency**: Enter an integer to specify the percentage of transparency of the portfolio bubbles. Small numbers will result in a plain color while larger numbers will result inthe bubbles becoming nearly invisible. The default value is set to 20%.
- **Boundary Width**: Enter an integer to specify the width in pixel of the boundary of the portfolio bubbles. The default value is set to 5 pixels.

X-Ranges and Y-Ranges tabs: These tabs are only relevant if you have NOT set a checkmark for the **Quadrant Layout** attribute in the **Properties** tab.

- **Tick Name <1-4>**: If necessary, define up to four captions per axis. If less than four caption names are defined for an axis, the axis range will be scaled accordingly. If no captions names are defined, the default captions Low, Medium, High, and Very High are applied to the axes.
- **Min. Value and Max. Value**: If necessary, define a minimum value and a maximum value for the X-axis and Y-axis. Objects with a lower/higher value than the defined min./max. value will be excluded from the graphic. Please note that portfolios without an explicit **Min. Value and Max. Value** definition will not be displayed if all values along an axis are identical.
- **Ignore Undefined Values**:Select the checkbox if objects with an undefined value for the corresponding axis shall not be displayed in the portfolio.

Power-Axis Legend: Enter short line-separated texts for the power-axis (Z-dimension) legend. The texts should indicate the meaning of the potential bubble sizes from smallest to largest. The legend will be displayed below the X-axis. Depending on the screen size, the information may be truncated in the portfolio view, although the last text will be displayed if information is truncated.

- **Ignore Undefined Values**: Select the checkbox if objects with an undefined value for the power-axis shall not be displayed in the portfolio.

Quadrants tab: This tab is only relevant if you have set a checkmark for the **Quadrant Layout** attribute in the **Properties** tab.

- **Quadrant <1-4>**: If necessary, define captions for each quadrant.
- **<X>Min. Value and Max. Value**: If necessary, define a minimum value and a maximum value for the X-axis. Bubbles with a lower/higher value than the defined minimum/maximum value will be excluded from the graphic.
- **<Y>Min. Value and Max. Value**: If necessary, define a minimum value and a maximum value for the Y-axis. Bubbles with a lower/higher value than the defined minimum/maximum value will be excluded from the graphic.

User Profiles tab:

- **User Profiles**: Select none, one, or more user profiles that have access permissions to this portfolio. A user will be able to view any portfolio that has either no user profile assigned and any portfolio that has the same user profile assigned as that which her/she is currently logged in with.

4) Click **OK** to save the portfolio or **Cancel** if you do not want to save it. The new portfolio is displayed in the table.

Assigning the Portfolio to an Object Class

You can assign multiple portfolios to a selected object class. The portfolio must be assigned to the object class that represents the objects that are to be displayed in the portfolio diagram.

For example, if you assign a portfolio to the object class Application, the portfolio could then be selected in the **Application Portfolio** page view for application groups because the page view displays applications in an application group. If you assign a portfolio to the object class Application Group, the portfolio could then be selected in the **Sub-Groups Portfolio** page view for application groups because the page view displays subordinate applications groups in an application group.

Portfolios assigned to an object class stereotype in the **Class Configuration** functionality will be ignored. Portfolios may only be assigned to the object class that the stereotype is based on. For example, if application stereotypes have been defined, the portfolios must be assigned to the class Application.

To assign a portfolio to a selected object class:

1) Go to the **Class Configuration** functionality in the **Configuration** module and click the relevant object class or object class stereotype in the **Class Configuration** explorer.
2) Click **Portfolios** to open the view.

3) In the toolbar, click **New** > **Add Portfolio**.
4) The selector opens. Enter search criteria, as needed, and click **Search**.
5) Click a portfolio and click **OK** to add the portfolio to the selected object class or click **Cancel** to exit without saving your changes. The portfolio is added to the table.

Configuring Aspect Evaluations for the Classes Application and Component

Aspect evaluations allow you to assess the performance of applications or components assigned to one or more respective application groups or component groups according to defined evaluation criteria. To this purpose, aspect evaluation types can be configured that can then also be analyzed in the context of aspect prioritization schemes as well as aspect portfolios.

Please note that the concept of aspect evaluation has no association with the concept of aspects, which are implemented in the context of applications or components. The configuration of aspects and aspect groups in business support maps is defined in the tool Alfabet Expand. For more information, see the section *Configuring Standard Business Support Matrices* in the reference manual *Configuring Alfabet with Alfabet Expand*.

The following information is available:

- Assigning Aspect Evaluations Types to the Object Classes Application and Component
- Assigning Aspect Prioritization Schemes to the Object Classes Application and Component
- Assigning Aspect Portfolios to the Object Classes Application and Component

Assigning Aspect Evaluations Types to the Object Classes Application and Component

An aspect evaluation is a qualitative assessment of the performance of applications or components assigned to one or more respective application groups or component groups according to defined evaluation criteria. Because an application, for example, may be a member of several different application groups, different qualitative assessments of the same application using the same evaluation criteria are possible for each application group that the application is assigned to.

For example, the component TradeNet may be relevant for different purposes in the enterprise architecture. The component may be suitable to varying degrees both as a business intelligence solution and as an OLAP database. An aspect evaluation would allow you to examine components in a component group based on specific aspect evaluation types, such as business importance or usage. Like conventional evaluation types, aspect evaluation type comprises one or more indicator types that are used to evaluate objects.

An aspect evaluation is based on one or more evaluation types, which are then assigned as aspect evaluation types to the object classes Application or Component. Like conventional evaluation types, aspect evaluation types comprise one or more indicator types that are used to evaluate objects.

Aspect evaluations are currently only available for the object classes Application or Component.

Before you can assign an aspect evaluation type to a selected object class, you must first create the evaluation type in the **Evaluation Types** page view in the **Evaluations & Portfolios** functionality and configure the relevant indicator type for the evaluation type. For more information, see the sections Creating an Evaluation Type and Configuring Indicator Types for an Evaluation Type.

Aspect evaluation types assigned to an object class stereotype in the **Class Configuration** functionality will be ignored. Aspect evaluation types may only be assigned to the object class that the stereotype is based on. For example, if application stereotypes have been defined, the aspect evaluation types must be assigned to the class Application.

To assign an aspect evaluation type to a selected object class:

1) Go to the **Class Configuration** functionality in the **Configuration** module and select either the class Application or Component in the **Class Configuration** explorer.
2) Click **Aspect Evaluation Type** to open the view
3) In the toolbar, click **New** > **Add Evaluation Type**.
4) The selector opens. Enter search criteria, as needed, and click **Search**.
5) Click an evaluation type and click **OK**. The evaluation type is added as an aspect evaluation type to the selected object class.

Assigning Aspect Prioritization Schemes to the Object Classes Application and Component

An aspect prioritization scheme is a prioritization scheme grouping evaluation types in an aspect evaluation in order to assess the performance of applications or components assigned to one or more respective application groups or component groups. An aspect prioritization scheme may also be implemented as an axis or dimension that is used in an aspect portfolio. Aspect prioritization schemes are currently only available for the classes Application and Component.

Aspect prioritization schemes are based on conventional prioritization schemes that are then assigned to an object class as an aspect prioritization schemes. Before you can assign an aspect prioritization scheme to a selected object class, you must first create the prioritization scheme in the **Prioritization Schemes** page view in the **Evaluations & Portfolios** functionality. For more information, see the section Configuring Prioritization Schemes.

Aspect prioritization schemes assigned to an object class stereotype in the **Class Configuration** functionality will be ignored. Aspect prioritization schemes may only be assigned to the object class that the stereotype is based on. For example, if application stereotypes have been defined, the aspect prioritization schemes must be assigned to the class Application.

To assign an aspect prioritization scheme to a selected object class:

1) Go to the **Class Configuration** functionality in the **Configuration** module and select either the class Application or Component in the **Class Configuration** explorer.
2) Click **Aspect Prioritization Scheme** to open the view
3) In the toolbar, click **New** > **Add Prioritization Scheme**.

4) The selector opens. Enter search criteria, as needed, and click **Search**.
5) Click a prioritization scheme and click **OK**. The prioritization scheme is added as an aspect prioritization scheme to the selected object class.

Assigning Aspect Portfolios to the Object Classes Application and Component

An aspect portfolio is a portfolio displaying either a three-dimensional graphic or a BCG quadrant layout that reports qualitative information about applications or components assigned to one or more respective application groups or component groups according to the aspect evaluation and aspect prioritization schemes that have been defined for the aspect portfolio. Aspect portfolios are only available for the classes Application and Component.

Aspect portfolios are based on conventional portfolios that are then assigned to an object class as an aspect portfolio. Before you can assign an aspect portfolio to a selected object class, you must first create the portfolio in the **Portfolios** page view in the **Evaluations & Portfolios** functionality. For more information, see the section Configuring Portfolios.

Aspect portfolios assigned to an object class stereotype in the **Class Configuration** functionality will be ignored. Aspect portfolios may only be assigned to the object class that the stereotype is based on. For example, if application stereotypes have been defined, the aspect portfolios must be assigned to the class Application.

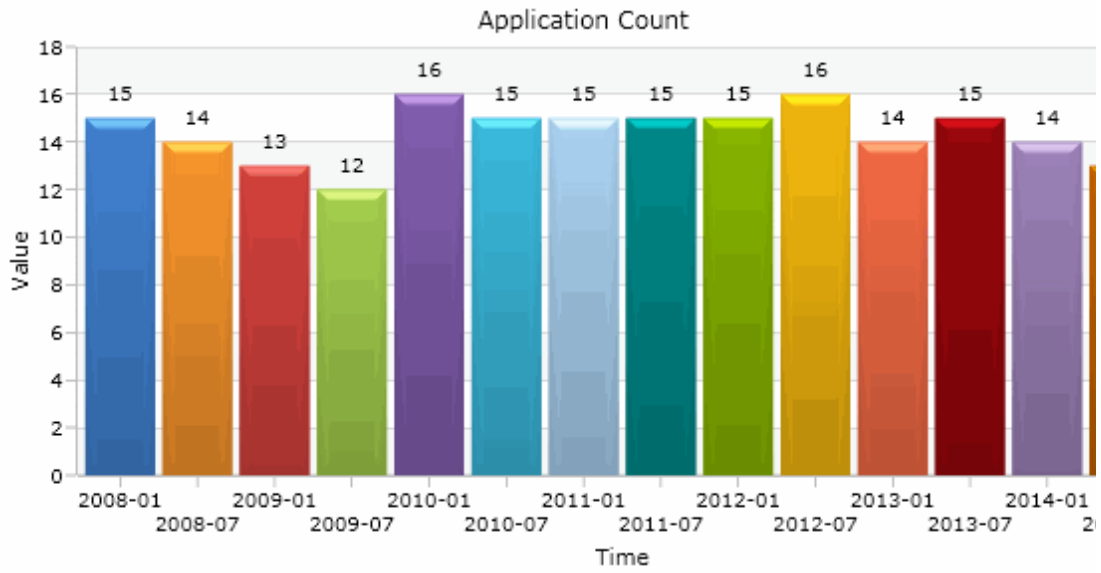
To assign an aspect portfolio to a selected object class:

1) Go to the **Class Configuration** functionality in the **Configuration** module and select either the class Application or Component in the **Class Configuration** explorer.
2) Click **Aspect Portfolios** to open the view
3) In the toolbar, click **New** > **Add Portfolio**.
4) The selector opens. Enter search criteria, as needed, and click **Search**.
5) Click a portfolio and click **OK**. The portfolio is added as an aspect portfolio to the selected object class.

Configuring Time Series Evaluations

Alfabet provides a capability that allows target evaluation values to be defined for objects in the Alfabet database. This capability enables the user community to define and track the evolution of target values for specified indicator types for a specific object over time.

For example, an enterprise wants to provide target values for the number of applications owned by organizations for the next two years. Target values would be defined for each organization every 6 months in the *Time Series Evaluation Page View*. The target values can be viewed as a business graphic in the *Time Series Analysis Page View*. The image below displays bi-annual target values of the number of applications for a selected organization.



The example above displays the historic target values defined for the indicator type Application Count for the selected organization. In this case, the periods 2008-01, 2008-07, 2009-01, 2009-07, etc. have been configured as the time series periods for the enterprise. The evaluation type Application Count has been assigned as a time series evaluation type to the object class Organization.

A configured report could additionally be created by your enterprise to provide a graphic visualization of the target evaluations in relation to the actual values defined for the organization in the *Evaluation Page View*. For more information about creating a configured report, see the section *Configuring Reports* in the reference manual *Configuring Alfabet with Alfabet Expand*.

In order to implement time series evaluations, you must configure time series periods and time series evaluation types.

Time series periods may be used in the context of time series evaluations as well as to measure value nodes via measure types in the context of the **Strategy Deduction** capability. If your enterprise plans to implement time series periods for measure types, you must specify time series periods in the Time Series Period Managemen Functionality.

1) In the **Time Series Period Management** functionality in the **Configuration** module, you can create the times series periods for which target evaluations should be defined in your enterprise (for example, 2020-01, 2020-7, 2022-01, 2022-07, etc. or Q1/2020, Q2/2020, Q3/2020, Q4/2020).

   The time series periods that you create can later be activated and deactivated. Any time series period (for example, 2020-01) that is activated will be visible in the *Time Series Evaluation Page View*

and can thus be evaluated. Once the time series period is deactivated it can no longer be evaluated in the *Time Series Evaluation Page View* nor will it be visible in the *Time Series Analysis Page View*.

2) Once you have created the time series periods that represent your enterprise's time series periods, you must assign relevant evaluation types to the object classes for which target evaluations are to be defined. This is carried out in the *Time Series Evaluation Page View* for the relevant object class in the **Class Configuration** functionality.

   Please note that time series definition is uniform for all measure types and an editor allows the target and rated values to be maintained for the measures based on a measure type. Time series periods may be skipped for those measure types where values are not available.

3) For each time series period that is currently active, users will be able to specify target values for the indicator types associated with the time series evaluation types for the objects in the relevant object class. Users can define the target indicators for an object in the *Time Series Evaluation Page View* in the object profile of the relevant object class.

- If your enterprise plans to compare an object's target indicators with the actual indicators defined in the *Evaluation Page View*, then you should ensure that the time series evaluation types assigned to an object class are the same as the conventional evaluation types associated with that object class.

  To check which evaluation types have been assigned to the selected object class, view the definition of evaluation types in the **Evaluation Types** page view for the selected object class in the **Class Configuration** functionality. For general information about the creation and definition of evaluation types, see the section Configuring Evaluation Types.

- Please note that computational rules are strictly available for ordinary indicators only. Computation rules cannot be defined for time series indicators. If a computation rule is necessary, the computation could be realized using ADIF. This would be realized via a data manipulating job that reads from the table `ASPECTINDICATOR` and updates/creates records in the same table. For more information, see the reference manual *Alfabet Data Integration Framework*.

The following information is available:

- Creating a Time Series Period
- Assigning Time Series Evaluation Types to an Object Class
  - Assigning a Time Series Evaluation Type to an Object Class
  - Detaching a Time Series Evaluation Type from an Object Class
- Creating a Time Series Group

Creating a Time Series Period

Time series periods may be used in the context of time series evaluations as well as to measure value nodes via measure types in the context of the **Strategy Deduction** capability.

For more information about the configuration of time series evaluations, see the section Configuring Time Series Evaluations. For more information about working with the **Strategy Deduction** capability, see the section *Strategy Deduction Functionality*.

Each time series period should represent one specific event or period for which target values or measures should be defined. A time series period could be, for example, 2020-01 or Q1/2020. You can create as many time series periods as required. If the time series periods shall be implemented in the context of time series evaluations, the time series period must be activated for users in the user community to define target values for that period. Only the time series periods that you activate will be displayed in the *Time Series Evaluation Page View*. Time series periods can be bundled in time series groups for the **Strategy Deduction** capability in the.Time Series Group Management Functionality

To create a times series period:

1) In the **Time Series Management** functionality, click **New** > **Create New Time Series Period**. The **Time Series Period** editor opens.
2) Define the following fields, as needed.
   - **Time Series Period Name**: Enter a unique name for the time series period. It is recommended that the name indicate the time unit (for example, Q1) that is relevant to your enterprise and that users will understand.
   - **Target Date**: Select a target date to indicate when the evaluation should be executed. The target date assigned to the time series period determines its sequence in the series of active time series periods displayed in the *Time Series Evaluation Page View*. Please note that the target date is not displayed in the *Time Series Evaluation Page View* and no control mechanism is associated with the target date.
   - **Is Active for Evaluation**: Select the checkbox to activate the time series period. Target indicators can be defined by the user community if the time series period is active. Users with relevant access permissions can define an object's target indicators for the relevant time series period in the *Time Series Evaluation Page View*. If the time series period is inactive, the time series period will not be displayed in the *Time Series Evaluation Page View* and thus indicators cannot be defined for the period.
   - **Description**: Enter a meaningful description about the time series period.
   - **Time Series Group**: Select the times series group that the time series period is assigned to. The time series group bundles a set of time series periods and is relevant for implementation in the **Strategy Network** capability.

     If the time series group is to be implemented in the context of the **Strategy Deduction** capability, it must be configured in the XML object ***ValueManager*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Strategy Deduction Capability* in the reference manual *Configuring Alfabet with Alfabet Expand*.
3) Click **OK** to save the time series period.

Assigning Time Series Evaluation Types to an Object Class

A time series evaluation type is an evaluation type for which target values can be defined for active time series periods for objects in the object class that the evaluation type is assigned to. Users can specify target values for the indicator types associated with the evaluation type. Multiple time series evaluation types can be assigned to an object class.

Once a time series evaluation type has been assigned to a selected object class, users will be able to define the associated indicators for the active periods in the *Time Series Evaluation Page View*. The configured time series periods will only be displayed in the *Time Series Evaluation Page View* for object classes that you have assigned at least one time series evaluation type. If no time series evaluation types are assigned to an object class, then the *Time Series Evaluation Page View* will be empty for that object class.

- If your enterprise plans to compare an object's target indicators with the actual indicators defined in the *Evaluation Page View*, then you should ensure that the time series evaluation types assigned to an object class are the same as the conventional evaluation types associated with that object class.

  To check which evaluation types have been assigned to the selected object class, view the definition of evaluation types in the **Evaluation Types** page view for the selected object class in the **Class Configuration** functionality. For general information about the creation and definition of evaluation types, see the section Configuring Evaluation Types.

- Please note that computational rules are strictly available for ordinary indicators only. Computation rules cannot be defined for time series indicators. If a computation rule is necessary, the computation could be realized using ADIF. This would be realized via a data manipulating job that reads from the table `ASPECTINDICATOR` and updates/creates records in the same table. For more information, see the reference manual *Alfabet Data Integration Framework*.

The following information is available:

- Assigning a Time Series Evaluation Type to an Object Class
- Detaching a Time Series Evaluation Type from an Object Class

Assigning a Time Series Evaluation Type to an Object Class

You can assign an existing evaluation type as a time series evaluation type to an object class.

To assign a time series evaluation type:

1) In the **Class Configuration** explorer, select the relevant object class folder and click **Time Series Evaluation Type**.
2) In the toolbar, click **New** > **Add Evaluation Type**.
3) The selector opens. Enter search criteria, as needed, and click **Search**.
4) Click an evaluation type and click **OK** to add the evaluation type as a time series evaluation type to an object class or click **Cancel** to exit without saving your changes. The evaluation type is displayed in the table.

Detaching a Time Series Evaluation Type from an Object Class

To detach a time series evaluation type from an object class:

1) In the table on the right, click the time series evaluation type that you want to detach.
2) In the toolbar, click the **Detach** button.
3) Confirm the warning by clicking **Yes or** click **No** to exit without saving your changes.

The time series evaluation type is removed from an object class.

Creating a Time Series Group

The **Time Series Group Management** functionality allows you to create various time series groups that shall be implemented in the **Strategy Deduction** capability. The time series group bundles a set of time series periods In the context of the **Strategy Deduction** capability, the time series group or individual time series periods may be assigned to a measure type in the *Measure Types Page View*. For more information about working with the **Strategy Deduction** capability, see the *Strategy Deduction Functionality*.

If the time series group is to be implemented in the context of the **Strategy Deduction** capability, it must be configured in the XML object ***ValueManager*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Strategy Deduction Capability* in the reference manual *Configuring Alfabet with Alfabet Expand*.

To create a time series group:

1) In the **Time Series Management** functionality, click **New** > **Create New Time Series Group**. The **Time Series Group** editor opens.
2) Define the following as needed:
   - **Time Series Group Name**: Enter the name of the time series group.
   - **Description**: Enter a description of the time series group.
   - **Time Series Periods**: Set a checkmark next to each time series period to be included in the time series group.
3) Click **OK** to save the time series group.

Chapter 3: Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities

Alfabet allows multiple currencies to be defined in the **Currencies** view in order to allow architecture and project costs to be captured across the enterprise.

For each currency and currency unit that is defined, a currency name, currency code, and currency symbol must be defined. Only the currency symbol will be displayed in editors, so you must ensure that the currency symbol is easy for users to interpret in the drop-down list (for example, T€).

One currency must be defined as the enterprise base currency (EBC), which is the principal currency used by the company. For each of the currencies defined, you can define currency units that allow a multiplier or factor to be used in order to simplify capturing currency values for that currency. For example, if the currency unit T€ = 1000€ is configured for the currency EUR with a factor = 1000, users could define 30 T€ to represent 30000€.

One currency or currency unit must be specified as the default currency. In some costs views in Alfabet, the currency/currency unit may be specified via a **Currency** filter and in other views the default currency will be automatically displayed for the entire report. When defining cost values, the user can specify the currency/currency unit as needed for the cost type selected in the cost editor.

For example, the company First Direct has its headquarters in Europe, but has corporate locations in India, Japan, the United States, and the United Kingdom. In this case, five currencies would be created, whereby the currency EUR would be specified as the enterprise base currency (EBC).

Name	Code	Symbol	EBC	Is Default	Factor
Euro Member Countries	EUR	€	✔		
Thousand Euro	T EUR	T€		✔	1000
India Rupee	INR	₹			
Lakh	INR - Lakh	L₹			100000
Japan Yen	JPY	¥			
United Kingdom Pound	GBP	£			
United States Dollar	USD	$			
T$	T$	T$			1000

For three of the currencies in the example, currency factors have been defined to simplify capturing currency values.

- The currency EUR is defined as the EBC. The currency EUR has a currency unit T EUR that allows the T EUR value entered to be factored by 1000 to compute the EUR value. Instead of entering a value of 30,000 for the € currency, users could select the T€ currency unit value and enter 30. The currency unit T EUR is defined as the default currency.
- Select the **Currency** and move it to the **Currency**.
- The currency INR has a currency unit INR - Lakh that allows the INR - Lakh value entered to be multiplied by 100000 to compute the INR value.
- The currency USD has a currency unit T$ that allows the T$ value entered to be multiplied by 1000 to compute the USD value.

All currencies and currency units defined in the **Currencies** view will be available in the relevant editors in which cost values can be captured. Users can specify the currency/currency unit for each cost type, thus allowing costs to be defined in the currency of the organization in which they are being captured. The currency values are stored in the Alfabet database in the currency that they have been captured in.

In order to aggregate and compute the cost data so that it can be analyzed in reports, a currency exchange rate must be defined for each currency. The currency exchange rate converts the price of that currency to the EBC currency. The period of the currency exchange rate starts with a defined start date and ends with the start date of the next defined currency exchange rate, thus allowing your enterprise to determine the period of time for which a currency exchange rate is valid. Currency exchange rates can be defined manually or via an ADIF job. For more information about importing data via the ADIF capability, see the reference manual *Alfabet Data Integration Framework*.

The following columns are displayed in the **Currencies** page view:

- **Name**: Displays the name of the currency.
- **Currency Code**: Displays the currency code.
- **Currency Symbol**: Displays the currency symbol.
- **EBC**: Displays a checkmark if the currency is the enterprise base currency.
- **Is Default**: Displays a checkmark if the currency is the default currency.
- **Factor**: Displays the multiplier defined to factor the currency unit in order to produce the value for the primary currency.
- **Currency Exchange Rate**: Displays the currency exchange rate defined. All currencies other than the EBC currency and the currency units require a currency exchange rate. If no exchange rate has been specified, the **Currency Exchange Rate** field will be highlighted in the report. Please note that currency values will be converted with the default **Currency Exchange Rate** = 1 if no currency exchange rate has been defined.

The following information is available:

- Creating Currencies and Their Currency Units
- Migrating Currencies
- Configuring Currency Exchange Rates for Currencies

Creating Currencies and Their Currency Units

You can create multiple currencies. One currency must be defined as the enterprise base currency (EBC). For each currency you create, you can create multiple currency units that allow a multiplier to be used in order to simplify the definition of cost values. A currency unit can only be assigned to one currency. One currency or currency unit must also be defined as the default currency.

If a user changes a currency unit to its currency in a cost view, then the last two digits of the costs after the separator (period, comma, etc.) will be used to round the currency up or down. For example, if a currency unit is specified as T$ to the currency $ and the factor is 1000, the value 399,999,99 in $ would be rounded to 400.00 in the currency T$.

The hierarchy of currencies and their currency units should not be changed after cost values have been defined.

To create a currency:

1) Go to the **Reference Data** functionality and click **Currencies**.
2) In the toolbar, click **New** > **Create New Currency**. You will see the **Currency** editor opens.
3) In the new window, enter information into each field, as required:
   - **Standard Currencies**: In order to fill the **Currency Name**, **Currency Code**, or **Currency Symbol** fields with a standard currency, select a currency in the field and click the **Copy** button. The values copied to the **Currency Name**, **Currency Code**, or **Currency Symbol** fields can be modified, if needed. If you do not select a standard currency, you must manually define the **Currency Name**, **Currency Code**, or **Currency Symbol** fields.
   - **Currency Name**: Enter a name for the currency.
   - **Currency Code**: Enter a code for the currency. A maximum of 10 characters is possible.
   - **Currency Symbol**: Enter a symbol for the currency. The currency symbol will be available in the **Currency** fields available in the various page views. Users should be able to easily understand the relation of the currency unit to its currency by means of the given symbol (for example, T€ = 1000E).
   - **Is Enterprise Base Currency**: Select the checkbox if the currency is the enterprise's principal currency (EBC). The EBC is used to calculate currency exchange rates for all other currencies defined in this view. The EBC is set per default in editors if no currency is defined for the specific context (for example, for a project). The currency can be changed in the editor by the user as needed. Only one of the available currencies can be defined as the EBC. A currency unit cannot be defined as an EBC.
   - **Is Default Currency**: The default currency is automatically used for cost reports that do not provide the possibility to define currencies. Select the checkbox if the currency is to be displayed per default in the **Currency** fields available in the various cost reports. Only one of the available currencies or currency units can be defined as a default currency.
4) Click **OK** to save the currency or **Cancel** if you do not want to save it.
5) To define a currency unit that allows users to capture a factor of the currency, select the currency in the table for which you want to create the currency unit and click **New** > **Create Currency Unit**.

   You can also specify an existing currency to be the currency unit of the selected currency. In this case, the existing currency will become a subordinate currency unit of another currency. Please note however that a currency unit cannot be assigned to two different currencies. To specify an existing currency as the currency unit of another currency, select the currency that you want to reassign as a currency unit and click **New** > **Define Existing Currency as Currency Unit**. In the selector, click the currency that the currency unit should be subordinate to and click **OK**. To define a currency unit to be a primary currency, click **New** > **Set Selected Currency as Primary Currency**. The currency unit will become a primary currency.

6) In the **Currency** editor, modify the **Currency Name**, **Currency Code**, and **Currency Symbol**. Please note the following:

- The **Currency Code** may have a maximum of 10 characters. It is recommended that it is derived from the currency code of the parent currency with additional prefixes and/or suffixes that explicate the meaning of currency unit.
- The **Currency Symbol** should be defined so that users will understand the meaning of the currency unit.

7) In the **Factor to Parent Currency** field, enter a real number that is the multiplier to apply to the value entered by the user in order to calculate the amount. For example, if the currency is EUR, then the currency unit might be T€ with a factor of 1000.
8) If the currency unit should be the default currency, select the **Is Default Currency** checkbox.
9) Click **OK** to save the currency unit or **Cancel** if you do not want to save it.

Migrating Currencies

The currency concept described in the chapter Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities was introduced with Alfabet release 10.0.

Currency configurations in prior releases that were configured in the XML object ***CustomUnitDef*** in Alfabet Expand will be automatically migrated upon update to the Alfabet release. The migrated currency will be automatically specified as the enterprise base currency (EBC) as well as the default currency. This can be modified as needed. For most customers, the migrated currency will actually be considered to be a currency unit in the new currency concept. If this is the case, a new currency should be created and defined as the enterprise base currency (EBC), and the migrated currency should be assigned to the new currency as its currency unit. For more information, see the section Creating Currencies and Their Currency Units.

If currencies have been defined for contracts in an Alfabet release prior to release 10.0, you will need to migrate the contract currencies and manually revise them based on the currency concept introduced with Alfabet release 10.0. Please note that the contract currencies configured prior to Alfabet release 10.0 were based on a property of type `String` and configured via the protected enumeration `Currency` available in Alfabet Expand.

To migrate contract currencies defined in Alfabet releases prior to release 10.0:

1) Go to the **Reference Data** functionality and click **Currencies**.
2) Create a currency for each enumeration item in the protected enumeration `Currency` in Alfabet Expand that you want to migrate. To do so, click **Create New Currency** and define the following:
   - **Currency Code**: Enter the string defined in the **Value** attribute of the enumeration item that you want to migrate. You must write the value of the enumeration item exactly as it is written in the **Value** attribute.
   - **Currency Name**: Enter a name for the new currency.
   - **Currency Symbol**: Enter a symbol for the new currency.
3) In the toolbar, click **New** > **Migrate Currencies Defined for Contracts**.
4) Confirm the informational message by clicking **Close**.
5) The contract currencies have been added to the **Currencies** page view. You can edit the migrated currency or reassign it as a currency unit to a new currency as described in the section Creating Currencies and Their Currency Units.

Configuring Currency Exchange Rates for Currencies

In order to analyze the cost data in reports, a currency exchange rate to the EBC must be defined for each currency other than the EBC in order to enable conversions to different currencies. A currency exchange rate definition must be specified for each currency implemented in your enterprise. Currency exchange rate definitions can be defined manually or via an ADIF job. For information about importing data via the ADIF capability, see the reference manual *Alfabet Data Integration Framework*.

The currency exchange rate can be defined in the **Currencies** page view as well as the **Currency Exchange Rates** page view, both available in the **Reference Data** functionality. The process to define both is similar, whereby the **Currency Exchange Rates** page view displays the currency exchange rates in a time schedule so that you can see the currency exchange rate definitions over time.

The **Currency Exchange Rates** page view automatically displays all currencies for which a currency exchange rate must be defined. The syntax of the currency exchange rate definition is as follows: <Currency code of EBC> -> <Currency code of non-EBC>. For example:

- EUR -> INR
- EUR -> USD
- EUR -> GBP
- EUR -> JPY

Each currency exchange rate has only a start date. The period of the currency exchange rate starts with a defined start date and ends with the start date of the next currency exchange rate definition, thus allowing your enterprise to determine the period of time for which a currency exchange rate is valid. The currency exchange rate will be applied for recalculations based on the date that the cost value was defined. If the cost value was defined before the start date of the first currency exchange rate definition, then the first currency exchange rate definition will be used for the conversion.

For the costs that are captured on a yearly basis, the currency exchange rate on January 1 of the year is expected to be defined. If not, the previous available currency exchange rate will be applied for the calculations. If no previous data is available, only then will the next available currency exchange rate be considered.

If no currency exchange rate is defined for Jan 1, 2018, the last entry for 2017 was for November, and a currency exchange rate for January 2, 2018 is defined, the currency exchange rate defined for November 2017 will be used for calculations of business case values for 2018.

If a currency exchange rate is defined to begin on Jan 1, 2018 and the next currency exchange rate is defined to begin on April 1, 2018, the first currency exchange rate will be considered for calculation for March 30 and 31, and the second currency exchange rate will be calculated starting with April 1, 2018.

The conversion of currencies based on the valid currency exchange rate is applied at runtime. If a cost value is captured in a currency unit that is not defined as the default currency, the conversion of cost values would be computed in the following order in order to convert the cost value to the default currency unit displayed in a report: Currency Unit of Defined Value > Currency Owning Currency Unit of Defined Value > Enterprise Base Currency (EBC) > Currency Owning Default Currency Unit > Default Currency Unit.

To specify a currency exchange rate definition:

1) In the **Currency Exchange Rates** page view, select the currency in the **Currency Exchange** column and click the **Edit** button to open the **Currency Exchange Rate** editor.

In the **Currencies** page view, select the currency and click **New** > **Edit Currency Exchange Rates**.

2) In the **Currency Exchange Rate** editor, the first column displays the EBC currency symbol as 1<currency symbol> and the last column displays the currency symbol of the currency that the currency exchange rate is being defined for. In the **Valid From** field, select the start date of the first currency exchange rate definition and in the **Rate** field, enter a real number that represents the currency exchange rate.
3) Repeat this for subsequent currency exchange rate definitions. The first date of each period is the last date of the previous period.
4) Click **OK** to save the currency exchange rate definitions and close the editor.

Chapter 4: Configuring Cost Types and Income Types for Cost Management Capabilities

Alfabet provides support to budget and analyze the costs and benefits of your IT landscape and project portfolio in order to understand and optimize IT costs as well as prioritize and maximize the effectiveness of proposed IT investments. Cost types allow investment and operational costs to be captured and tracked whereas income types allow benefits in the context of project business cases to be documented.

For an overview of various methodologies to manage costs for architecture objects in Alfabet, see the sections *OPEX Optimization* and *Cost Driver Analysis* in the reference manual *Portfolio Management Advanced*.

The following provides an overview of the configuration steps necessary to work with the various cost management capabilities in Alfabet:

- In order to work with the cost management capabilities in Alfabet, you must configure the necessary cost types and income types required by your enterprise. Please note the following regarding the various cost types implemented in Alfabet:
  - Maintenance cost types (typically for OPEX costs) and can be specified. These cost types are used to capture and analyze the following:
    - Operational costs of applications, deployments, ICT objects, projects, and service products
    - Business service and business support costs for business processes
    - ICT costs relevant for business capabilities

    The following page views in Alfabet display operational cost types:

Object Class	Page View
Application	*Consolidated Operational Expenses Page View* *Operational Expenses Page View*
Deployment	*Operational Expenses Page View*
Service Product	*Operational Expenses Page View*
ICT Object	*Operational Expenses Page View* *Lifecycle Costs Page View* *Lifecycle Costs Chart Page View*
ICT Object Group	*Operational Expenses Page View* *Cost Aggregation Page View*

Object Class	Page View
ICT Object Category	*Operational Expenses Page View* *Cost Aggregation Page View*
Organization	*Operating Expenses Page View* *Business Support Costs Hierarchy Page View*
Business Process	*Business Support Costs Hierarchy Page View* *Business Support Costs Page View* *Business Service Costs Page View* *Cost Benchmarking Page View*
Business Support	*Cost Benchmarking Page View*
Capability Map	*Capability ICT Cost Report Page View*

- Additional non-operational cost types (typically for CAPEX costs) as well as income types (benefits) can be specified for projects. Non-operational cost types and income types are used to define business cases as well as to capture skill requests and resource requests for projects.

  The following page views in Alfabet display operational and non-operational cost types and income types:

Object Class	Page View
Project	*Cost Report Page View* *Cost Accrual Page View* *Business Case Page View* *Business Case Comparison Page View* *Project, Skill Request and Resource Request Time Schedule Page View* *Cash Out Planning Page View*
Bucket	*Cost Accrual Page View*

Object Class	Page View
ICT Object	*Lifecycle Costs Page View* and *Lifecycle Costs Chart Page View* (CAPEX costs will be displayed in the views if the ICT object or its associated applications are defined in the **Primary Architecture Element** field in the **Project** editor)

- Cost types associated with cost centers that represent the costs of applications, deployments, ICT objects, projects, and service products. The following page views in Alfabet display these cost types:

Object Class	Page View
Cost Center	Cost Accrual Page View
Cost Center Group	*Cost Aggregation Page View*

- All cost types and income types for any cost management capability must be configured in the **Reference Data** functionality. This is explained below in the sections Creating Cost Types and Cost Type Hierarchies and Creating Income Types.
- If you plan to track OPEX costs for architecture objects and projects, you must assign the operational cost types to the class **Project** in the **Maintenance Cost Types** page view in the **Class Configuration** functionality. This is explained below in the section Specifying Cost Types as Maintenance (Operational) Costs.
- If you plan to track CAPEX costs or other non-operational costs for projects, you must assign the relevant cost types to the class **Project** in the **Cost Types** page view in the **Class Configuration** functionality. This is explained below in the section Configuring Cost Management Capabilities for Project Management.
- If you plan to track business support costs for business processes (which are based on the operational costs defined for applications), your solution designer must configure the XML object ***CostManagerDef*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Calculation of Business Support Costs* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- If you plan to specify business cases and resource costs (via skill requests and resource requests) for projects:
  - You must assign the project cost types to the class **Project** in the **Cost Types** page view and you must assign income types to the class `Project` in the **Income Types** page view in the **Class Configuration** functionality. This is explained below in the section Assigning Cost Types to the Object Class Project.
  - Your solution designer must configure the XML object ***CostManagerDef*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Business Case Definition for Projects* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- If you plan to track costs for cost centers, you must configure cost center types in the **Reference Data** functionality and create cost centers in the **Cost Centers** functionality (both functionalities

are available in the **Configuration** module). The creation of cost centers and cost center types as well as the assignment of cost types to the cost center types is explained in the section Configuring Cost Centers for Cost Management Capabilities. Furthermore, the cost definition categories (`Request`, `Budget`, `Current`, `Obligation`, `Forecast`) available for cost centers and their associated applications, ICT objects, deployments, and projects must be configured by your solution designer in the XML object ***CostManagerDef*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Editability of Costs in Cost Centers* and *Configuring the Editability of Costs for Architecture Objects* in the reference manual *Configuring Alfabet with Alfabet Expand*.

- To use any cost management capability available in Alfabet:
  - The fiscal year for your enterprise must be configured by your solution designer in the XML object ***CostManagerDef*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Fiscal Year for Cost Reporting in Your Enterprise* in the reference manual *Configuring Alfabet with Alfabet Expand*.
  - The currency and currency units displayed on all page views and editors in which costs are captured and visualized is configured in the **Reference Data** functionality. For more information about the configuration of currencies, see the chapter Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities.

The following information is available:

Specifying Cost Types for Cost Management

A cost type is a classification of costs. Cost types are defined by the enterprise to ensure comparability in the definition and evaluation of the investment and operation costs associated with ICT objects, applications, deployments, projects, and service products. Cost types are critical for the definition of business cases as well as for cost planning for projects. An unlimited number of subordinate cost types may be defined for any cost type in order to allow for bottom-up cost estimation and analysis.

All cost types are created in the **Reference Data** functionality. For each cost type created, you can define its subordinate cost types. Once the cost types are created, they must be further configured so that they are available in the relevant functionality in Alfabet.

The following cost and budget issues can be addressed in Alfabet:

- Operational costs of applications, ICT objects, deployments, projects, and service products
- Business service and business support costs for business processes
- ICT costs relevant for business capabilities
- Other non-operational costs relevant for business case (such as capital expenditures) and skill requests and resource requests for projects
- Cost centers that represent the costs of applications, deployments, ICT objects, and projects

The following information is available:

Creating Cost Types and Cost Type Hierarchies

The first requirement to implement the cost management capabilities is to define the cost types required. This includes all potential operating cost types for applications, ICT objects, deployments, projects, and service products, other (non-operational) cost types relevant for documenting the business cases and skill requests(/resource requests for projects. Each cost type may have subordinate cost types defined.

Once the cost types have been created, they will require further configuration, depending on the purpose of the cost type:

- **For operational cost types:** All cost types that are relevant for specifying the OPEX costs of applications, ICT objects, deployments, projects, and service products must be assigned to the class **Project** via the **Maintenance Cost Types** page view in the **Class Configuration** functionality. These cost types are also used to understand business support and business service costs for business processes and organizations. For more information, see the section Making Maintenance Cost Types Available to Architecture Objects.
- **For non-operational cost types:** Cost types that are relevant for capturing non-operational costs such as CAPEX costs for projects must be assigned to the object class **Project** via the **Cost Types** page view. For more information, see the section Assigning Cost Types to the Object Class Project.
- **For cost center cost types:** Cost types that are relevant for cost centers must be assigned directly to the cost center. This is explained in the context of cost center

configuration in the chapter Configuring Cost Centers for Cost Management Capabilities.

Please consider the following about capturing costs for architecture objects:

- If costs are assigned to ICT objects, then costs should not be assigned to applications. Please note that any costs assigned to an application will be aggregated to the ICT object that the application is associated with. Therefore, as a rule, costs should not be assigned to both applications and ICT objects.
- Costs can be assigned to deployments and applications. Please note the following:
  - Application-related costs such as Development, Maintenance, Help Desk, License Fees, etc. should be captured on the application.
  - Deployment-related costs such as Operations, Hardware, Energy, Data Center Fees, etc. should be captured on the deployment.

    If costs are assigned to deployments, the deployment costs will be aggregated to the associated application. The deployment and application costs are aggregated to the associated ICT object.

To create cost types on the root level as well as their subordinate cost types:

1) Go to the **Reference Data** functionality click **Cost Types**.
2) In the toolbar, click **New** > **Create New Cost Type**. You will see the **Cost Type** editor opens.
3) In the new window, enter information into each field, as required:
   - **Name**: Enter a unique name for the cost type.
   - **Description**: Enter a meaningful description that will clarify the purpose of the cost type.
   - **Show for Skill Requests/Resource Requests**: Select the checkbox if the cost type may be defined for skill requests and resource requests. You must select this checkbox to make the cost type available in the **Cost Type** field in the **Skill Request** editor and **Resource Request** editor.
4) Click **OK** to save the cost type or **Cancel** if you do not want to save it. The new cost type  is displayed on the root level in the explorer tree.
5) To define cost types that are subordinate to the root cost type or any other cost type that should have subordinate cost types, select the cost type in the explorer tree and click the **Cost Types** page view to open it. Define the subordinate cost type as described in steps 4-5.

Once the cost types are created you can do one of the following:

- Moving a Cost Type to the Selected Cost Type
- Detaching a Cost Type from a Parent Cost Type
- Deleting a Cost Type

Moving a Cost Type to the Selected Cost Type

A cost type can only have one parent cost type. If you move a cost type to the selected cost type, it will be removed from the original parent cost type.

To make an existing cost type subordinate to the selected cost type:

1) n the **Cost Types** page view, select the cost type that you want to assign to a different parent cost type.
2) In the toolbar, click **New** > **Move Existing Cost Type Here**. The object selector opens.
3) Click the cost type that you want to assign as subordinate to the selected cost type.
4) Click **OK** to save the cost type or **Cancel** if you do not want to save it.

Detaching a Cost Type from a Parent Cost Type

To detach a cost type from a parent cost type:

1) In the **Cost Types** page view, select the cost type that you want to remove from the parent cost type.
2) In the toolbar, click the **Detach**  button.
3) Confirm the warning by clicking **Yes or** click **No** to exit without removing the selected object(s).

Deleting a Cost Type

If you delete an object in the object class Cost Type, it will be irrevocably deleted from the Alfabet database. If you delete a cost type that has subordinate cost types assigned to it, all subordinate cost types will also be deleted from the database.

1) In the **Cost Types** page view, select the cost type that you want to delete.
2) In the toolbar, click the **Delete**  button.
3) Confirm the warning by clicking **Yes,** or click **No** to exit without deleting the selected object(s).

Making Maintenance Cost Types Available to Architecture Objects

If your user community will be capturing or analyzing operational and maintenance costs for applications, ICT objects, deployments, service products, and projects, you must assign the relevant cost types as maintenance cost types to the object class **Project**.

The cost types must be assigned to the class **Project** regardless of the object class or object class stereotype that will be the target of cost management.

The maintenance cost types are available in the following contexts:

- Operational costs of applications, ICT objects, deployments, projects, and service products

- Business service and business support costs for business processes
- ICT costs relevant for business capabilities

In some cases, your enterprise may decide that the values specified for some cost types should not be editable. This may be the case, for example, if costs from an external system are imported to Alfabet via ADIF and thus should not be changed by Alfabet users. If this is the case, you can specify some or all of the maintenance cost types for the object classes **Application**, **Deployment**, **ICT Object**, and **Service Product** as read-only cost types. The specified read-only cost types cannot be edited by users in the context of the Alfabet interface. For more information, see the section Specifying Cost Types as Read-Only Cost Types.

1) Go to the **Class Configuration** functionality in the **Configuration** module and click the **Project** class in the **Class Configuration** explorer.
2) Click the **Maintenance Cost Types** page view to open it.
3) Click **New** > **Add Cost Type** to assign the cost types that are relevant for capturing operational costs for applications, deployments, and ICT objects.
4) In the object selector, select the cost type(s) and click **OK**. The cost type and its subordinate cost types will be available in the relevant page views that track and analyze operational costs for architecture objects and projects.

Assigning Cost Types to the Object Class Project

If your user community will be capturing or analyzing business cases for projects or capturing CAPEX costs for projects, you must assign the relevant non-operational cost types to the object class **Project**. The cost types will not be available in the relevant page views if they are not assigned to the class **Project**.

Operational cost types (OPEX) relevant for applications, deployments, ICT objects, projects, and service products must be assigned to the class **Project** in the **Maintenance Cost Types** page view in the **Class Configuration** functionality. For more information, see the section Assigning Cost Types to the Object Class Project.

1) Go to the **Class Configuration** functionality in the **Configuration** module and click the **Project** class in the **Class Configuration** explorer. Please note that you must configure the class **Project** and NOT **Project** (**ProjectStereotype**).
2) Click **Cost Types** page view to open it.
3) Click **New** > **Add Cost Type** to assign the cost types that are relevant for capturing the costs in business cases.
4) In the object selector, select the cost type(s) and click **OK**. The cost type and its subordinate cost types will be available in the relevant page views

Specifying Cost Types as Read-Only Cost Types

In some cases, your enterprise may decide that the values specified for some cost types should not be editable. This may be the case, for example, if costs from an external system are imported to Alfabet via ADIF and thus should not be changed by Alfabet users. In this case, you can specify cost types to be read-only. The costs defined for read-only cost types will be visible in the relevant page views and editors but they cannot be edited in the Alfabet interface.

The following cost types can be defined as read-only:

- Maintenance cost types that are available for the object classes **Application**, **Deployment**, **ICT Object**, and **Service Product** or their object class stereotypes. Please note the following:
  - The cost types must first be assigned to the **Maintenance Cost Types** page view **for the object classProject** in the **Class Configuration** functionality. These cost types are then available in the **Read-Only Cost Types** page view available for the relevant object class (**Application**, **Deployment**, **ICT Object**, and **Service Product**) in the **Class Configuration** functionality.
- Maintenance cost types as well as non-investment cost types that are available for the object class **Project**. The cost types must first be assigned to the **Maintenance Cost Types** page view and the **Cost Types** page view **for the object classProject** in the **Class Configuration** functionality. These cost types are then available in the **Read-Only Cost Types** page view available for the object class **Project** as well as the **Read-Only Cost Types** page view available for the object class stereotypes specified for the class **Project**.

Please note the following:

- The read-only setting specified for a cost type will NOT be inherited by its subordinate cost types. Each cost type in the cost type hierarchy must be explicitly specified as a read-only cost type, as needed.
- Read-only cost types specified for an object class stereotype will overwrite the read-only cost type definition of the base object class.
- For the object class **Project**: Please note that if a cost type is specified as a read-only cost type, costs may nevertheless be transferred to the project via the **Transfer from Skill Requests/Resource Requests** functionality in the *Business Case Page View*. If the **Transfer from Skill Requests/Resource Requests** functionality is executed, any costs defined for skill requests will overwrite the existing project's costs including all costs specified for a read-only cost type. If costs should not be transferred from skill requests, the **Transfer from Skill Requests/Resource Requests** functionality should be hidden by your solution designer. For more information, see the section *Hiding Functionalities in a Page View or Configured Report* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- Read-only cost types may also be defined by authorized users for an individual application, deployment, ICT object, service product or project via the *Read-Only Cost Types Page View* available in the object profile. If a cost type is set as read-only for an object class or object class stereotype, it will not be possible to change the read-only definition for that cost type for a specific object in the *Read-Only Cost Types Page View* available in the object profile.

To specify a cost type as a read-only cost type:

1) Go to the **Class Configuration** functionality in the **Configuration** module and click the relevant object class or object class stereotype in the **Class Configuration** explorer.
2) Click **Read-Only Cost Types** page view to open it.
3) In the table, select the cost type that you want to define as read-only and click **Set/Clear Read-Only Setting**. A checkmark is set in the **Is Read-Only** column.

To remove the read-only definition from the cost type, select the read-only cost type and click **Set/Clear Read-Only Setting**. The checkmark will be removed in the **Is Read-Only** column.

4) Repeat this step for all cost types assigned to the object class, as needed.

Creating Income Types

An income type is a classification of income. Income types typically represent benefits in a business case and are defined by the enterprise to ensure comparability in the monetary evaluation of activities and services for projects. Income types are critical for the definition of business cases in Alfabet.

To create an income type:

1) Go to the **Reference Data** functionality and click **Income Types**.
2) In the toolbar, click **New** > **Create New Income Type**. The **Income Type** editor opens.
3) Enter information into each field, as required.
   - **Name**: Enter a unique name for the income type.
   - **Description**: Enter a meaningful description that will clarify the purpose of the income type.
4) Click **OK** to save the income type or **Cancel** if you do not want to save it.

Once the income types are assigned, you can assign the income types to the class **Project**. For more information, see the section Assigning Income Types to the Object Class Project.

Assigning Income Types to the Object Class Project

All income types that should be documented in business cases must be assigned to the class **Project**. Income types will not be available in the relevant page views if they are not assigned to the class **Project**. Income can be defined for projects in the *Business Case Page View* and *Benefit Tracking Page View*.

1) Go to the **Class Configuration** functionality and click the **Project** class in the **Class Configuration** explorer. Please note that you must configure the class **Project** and NOT **Project** (**ProjectStereotype**).
2) Click **Income Types**.
3) Click **New** > **Add Income Type** to assign the income types that are relevant for capturing benefits in business cases.
4) In the object selector, select the income type(s) and click **OK**. The income type will be available in the relevant page views

Chapter 5: Configuring Cost Centers for Cost Management Capabilities

The cost management capabilities support users to manage costs based on cost centers.

A cost center is a means to centrally define costs for a specified period of time and allocate them to a group of architecture objects (applications, deployments, ICT objects, projects, or service products) according to a defined allocation scheme.

A cost center is based on a cost center type, which serves as a template to define cost centers. The costs assigned to a cost center are based on the cost types configured for the associated cost center type. A cost center can be assigned to an unlimited number of cost center groups.

For example, several cost centers for application-related costs such as Development, Maintenance, Help Desk, License Fees are created. The cost center development and Maintenance are both based on the cost center type Operations. These cost centers will have a standardized set of cost types available that are relevant for application-related operations. Each cost center may have additional cost types explicitly added that are not included in the cost center type Operations in order to capture, for example, development-specific costs (for example, Third-Party Licenses) or maintenance-specific costs (On-Site Consultation Fees).

For an overview of various methodologies to manage costs via cost centers in Alfabet, see the section *OPEX Optimization* in the reference manual *Portfolio Management Advanced*.

The **Cost Centers** functionality allows you to create and define the cost types that are relevant for the cost center. In the **Cost Center** editor available in the **Cost Centers** functionality, you must specify whether the cost center's costs are either automatically and equally distributed among the architecture objects or whether the allocation of the costs must be manually defined for the architecture objects. Once cost centers are created, you (or users with access permissions to the **Cost Center Groups** explorer) may further refine the cost center. In the **Objects** page view, you can assign the relevant applications, deployments, networks, ICT objects, service products or projects to the cost centers.

In the **Cost Accrual** page view, you can define the cost center's budget for the available cost types. Once the cost center budget has been defined, the costs specified for the cost center can be allocated to the architecture objects via the **Reallocate Costs** button in the **Objects** page view and thus included in the *Operational Expenses Page View* for applications, deployments, networks, ICT objects, and service products and the *Business Case Page View* for projects.

For example, the cost center Development has several cost types assigned, one of which is the cost type Hardware Replacement Costs. The cost center is defined to have equal allocation of its costs assigned to the 5 applications that have been assigned to it. If the cost for the cost type Hardware Replacement Costs has been budgeted for the current year as $20,000 for the cost center, then each of the 5 applications will have $4,000 allocated to the cost type Hardware Replacement Costs. This will then be displayed column for the cost type Hardware Replacement Costs in the *Operational Expenses Page View* of each application.

Once cost centers have been created, users with relevant access permissions may access the **Cost Center Groups** explorer and structure the cost centers in cost center groups. In the **Cost Center Groups** explorer, users may also assign applications, deployments, networks, ICT objects, service products, and projects to the cost centers and define the allocation of the cost center budget to those architecture objects, specify the cost distribution for the architecture objects and reallocate the cost center costs to the architecture objects. The **Cost Center Groups** explorer is available in the **Portfolio** module.

The following provides an overview of the configuration steps necessary to work with the cost center capability in Alfabet:

- Cost types that are relevant to the cost centers in your enterprise must be configured in the **Reference Data** functionality. The creation of cost types is explained in the section Specifying Cost Types for Cost Management in the chapter Configuring Cost Types and Income Types for Cost Management Capabilities.
- Cost center types that are relevant to the cost centers in your enterprise must be configured in the **Reference Data** functionality. The cost types you have created in the previous step will be assigned to the cost center types. This is described below in the section Configuring Cost Center Types.
- Cost centers created in the **Cost Centers** functionality that are relevant to the cost centers in your enterprise must already be configured in the **Reference Data** functionality. Each cost center will be based on a cost center type. This is described below in the section Creating Cost Centers and Specifying Cost Allocation.
- The cost definition categories (categories (**Request**, **Budget**, and **Current**) available for cost centers and their associated applications, deployments, networks, ICT objects, service products, and projects must be configured by your solution designer. For more information, see the section *Configuring the Editability of Costs in Cost Centers* and *Configuring the Editability of Costs for Architecture Objects* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- To use any cost management capability available in Alfabet:
  - The fiscal year for your enterprise must be configured by your solution designer in the XML object ***CostManagerDef*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Fiscal Year for Cost Reporting in Your Enterprise* in the reference manual *Configuring Alfabet with Alfabet Expand*.
  - Currencies must be configured in the **Reference Data** functionality. For more information, see the section Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

Cost reallocation may be configured by your enterprise to be executed regularly via a batch process. This batch execution is configured via an ADIF import scheme in the Alfabet Data Integration Framework capability. If this is the case, you must explicitly update the imported cost allocation to the objects assigned to the cost center in the **Objects** page view in the user interface. For more information, see the section Updating the Cost Allocation After a Batch Update via ADIF. For more information about the configuration of an ADIF import scheme for cost allocation, see the section *Import Scheme to Automatically Reallocate Cost Center Costs* in the reference manual *Alfabet Data Integration Framework*.

The following information is available:

Configuring Cost Center Types

Before cost centers can be created, cost center types must first be defined. The **Cost Center Types** page view in the **Reference Data** functionality allows you to create cost center types that are needed in order to create cost centers.

A cost center type serves as a template for creating cost centers. One or more cost types can be assigned as standard cost types to a cost center type. When a cost center is created, it is based on a cost center type. The cost types assigned to the cost center type will be automatically assigned to the cost center.

You can create multiple cost center types.

To create a cost center type:

1) Go to the **Reference Data** functionality in the **Configuration** module and click **Cost Center Types**.
2) In the toolbar, click **New** > **Create New Cost Center Type**. The **Cost Center Type** editor opens.
3) Enter information into each field, as required.
   - **Name**: Enter a unique name for the cost center type.
   - **Description**: Enter a meaningful description that will clarify the purpose of the cost center type.
4) Click **OK** to save the cost center type or **Cancel** if you do not want to save it.
5) Select the new cost center type in the table and click the **Navigate**  button to open its object profile.
6) Click **Cost Types**.
7) Click **New** > **Add Cost Type** to assign the cost types that are relevant for cost centers.
8) In the object selector, select the cost type(s) and click **OK**. The cost type and its subordinate cost types will be added to the cost centers based on the selected cost center type.

Deleting a Cost Center Type

If you delete an object in the object class Cost Center Type, it will be irrevocably deleted from the Alfabet database.

1) Select the cost center type that you want to delete.
2) In the toolbar, click the **Delete**  button.
3) Confirm the warning by clicking **Yes,** or click **No** to exit without deleting the selected object(s).

Creating Cost Centers and Specifying Cost Allocation

The **Cost Centers** functionality also allows you to create new cost centers and define the cost allocation for the architecture objects that will be assigned to the cost center.

A cost center is a means to centrally define costs for a specified period of time and allocate them to a group of architecture objects (applications, deployments, ICT objects, projects, or service products) according to a defined allocation scheme.

A cost center is based on a cost center type, which serves as a template to define cost centers. The costs assigned to a cost center are based on the cost types configured for the associated cost center type. A cost center can be assigned to an unlimited number of cost center groups.

The **Cost Centers** functionality displays all existing cost centers. To limit the set of cost centers displayed, define the following filters and click the **Search** button:

- **Name / ID**: Enter the name or identification number of the cost center that you want to view.
- **First Date**: Enter the date that defines the start of the displayed that you want to view. Cost centers with an end date after this date will be displayed
- **Last Date**: Enter the date for the end of the timespan that you want to view. Cost centers with a start date before this date will be displayed
- **Cost Type**: Select a cost type. All cost centers based on this cost type will be displayed.
- **Object**: Click the **Search** icon and in the **Search For** field, select the relevant object class to find the object(s) whose cost centers you want to display. All cost centers that the selected object is assigned to will be displayed in the table.

To create a new cost center:

1) Go to the **Cost Centers** functionality and click **New** > **Create New Cost Center**. The object selector opens.
2) Select the cost center type that the new cost center should be based on and click **OK**.
3) In the **Cost Center** editor, enter the data, as needed. Each field is defined below:

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each object in the inventory. This number cannot be edited.
   - **Name**: Enter a unique name for the cost center.

- **Start Date**: Enter the start date of the cost center in the format appropriate to your culture settings or select the start date in the calendar.
- **End Date**: Enter the end date of the cost center in the format appropriate to your culture settings or select the end date in the calendar.
- **Equal Allocation**: Select the checkbox if the distribution of costs across architecture objects assigned to the cost center is equal. Clear the checkbox if the cost allocation will be manually defined for the cost center. If you select the checkbox, the **Edit Allocation** functionality will be disabled in the **Objects** page view. Clear the checkbox if the cost allocation should be manually defined for the project. In this case, the **Edit Allocation** functionality will be available in the **Objects** page view.
- **Owner**: Click the **Search** icon to select the organization that owns the cost center. The costs accrued on this cost center will be associated with the owner organization.
- **Cost Center Type**: Click the **Search** icon to select a cost center type.
- **Description**: If necessary, enter a meaningful description that will clarify the purpose of the cost center.
- **ERP Cost Center ID**: If you plan to interface with an ERP system, enter the ID of the cost center in the ERP system. This will allow you to communicate and map costs in the ERP system.
- **ERP Instance ID**: If you plan to interface with an ERP system, enter the ID of the instance of the ERP system that manages this cost center. This will allow you to capture multi-instance or multi-mandate ERP installations in your corporation.
- **Currency**: Select the default currency or currency unit to capture and analyze cost center costs. The currency/currency unit may be changed in the relevant views and reports, as needed. For more information, see the section Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected object. The authorized user will have Read/Write access permissions to the object and is authorized to maintain the object in Alfabet.
- **Authorized User Groups**: Select the checkbox to assign Read/Write access permissions to all users in the selected user group. For more detailed information about the concept of access permissions, see the section *Understanding Access Permissions in Alfabet* in the reference manual *Getting Started with Alfabet*.

4) Click **OK** to save the cost center or **Cancel** if you do not want to save it.

If you delete an object in the object class Cost Center, it will be irrevocably deleted from the Alfabet database. If any of the following dependent objects or data is defined for the deleted object, these will also be deleted: Attachment, Deputy, Role.

Creating Cost Centers and Specifying Cost Allocation

Cost types are typically added to a cost center via the cost center type that you assign to the cost center. In this case, the cost types are available as default cost types for the selected cost center. If necessary, you can remove any of the default cost types from a cost center or add additional cost types to that are relevant for the cost center that weren't included in the cost center type.

The cost types that are relevant to the cost centers in your enterprise must be configured in the **Reference Data** functionality. The creation of cost types is explained in the section Specifying Cost Types for Cost Management in the chapter Configuring Cost Types and Income Types for Cost Management Capabilities.

You can add an unlimited number of cost types to the selected cost center type. It is highly recommended that you only select cost types on the leaf level of the cost type hierarchy in order to avoid an excessively large hierarchy in the **Cost Accrual** page view.

To add an existing cost type to a cost center type:

1) Open the **Cost Types** page view in the object profile of the relevant cost center.
2) In the toolbar, click **New** > **Add Existing Cost Types**.
3) The selector opens. Enter search criteria, as needed, and click **Search**.
4) Click the cost type and click **OK**. The cost type appears in the table.

To remove a cost type from the cost center, select the cost type in the table and click the **Detach** button.

Defining the Cost Center's Budget

The **Cost Accrual** page view allows you to define and view the current costs for a cost center for a specified period of time. Once the cost center budget has been defined, you can distribute the maintenance costs of the cost center to the architecture objects assigned to it in the **Objects** page view. For more information, see the sections Creating Cost Centers and Specifying Cost Allocation and Assigning Objects to a Cost Center and Allocating Costs to the Objects.

Please note the following regarding the configuration of the cost information in this view:

- Cost types must be preconfigured for the class **Cost Center** in order to see results in this report. For more information about the configuration of cost types in the **Reference Data** functionality in the **Configuration** module, see the section Specifying Cost Types for Cost Management in the chapter Configuring Cost Types and Income Types for Cost Management Capabilities..
- The relevant cost definition categories (**Request**, **Budget**, and **Current**) available for cost centers and their associated applications, ICT objects, deployments, networks, service products, and projects must be configured by your solution designer in the XML object ***CostManagerDef*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Editability of Costs in Cost Centers* and *Configuring the Editability of Costs for Architecture Objects* in the reference manual *Configuring Alfabet with Alfabet Expand*.

- Currencies must be configured in the **Reference Data** functionality. For more information, see the section Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

In the **First Year** field, enter the start year of the report and in the **Last Year** field, enter the end year of the report, and in the **Total Sum Currency** field, select the currency or currency unit to use to calculate the costs and click **Update**.

Each column in the report is defined below:

- **Cost Type**: Displays the name of the cost type assigned to the cost center
- **[Years]**: Displays the yearly costs defined for the cost type.
- The **Total** row at the bottom of the view displays the sum of the accrued costs for each year in the selected currency/currency unit.

To edit the cost center budget:

1) In the toolbar of the **Cost Accrual** page view, click the **Edit** button. The **Cost Center Costs** editor opens.
2) Enter your changes in a matrix cell in the < **Request** / **Budget** / **Current** > column for the relevant cost type and click again anywhere outside of the cell matrix. The value for the cost type will be displayed in the < **Request** / **Budget** / **Current** > column.
3) Click **OK** to save your changes or **Cancel** to exit without saving your changes.
4) To redistribute the maintenance costs of the cost center to the architecture objects assigned to it, see the section Allocating the Cost Center Costs to the Objects.

Assigning Objects to a Cost Center and Allocating Costs to the Objects

The **Objects** page view allows you to add applications, deployments, ICT objects, networks, projects, and service products can be assigned to a cost center. A cost center may have an unlimited number of objects assigned to it and each object may be assigned to multiple cost centers, if needed.

If the cost center has been defined to have its costs equally distributed across the architecture objects, you will see the automatically generated percentage for the costs that will be allocated from the cost center to each object in the **Allocation** column. If the cost center has not been specified for its costs to be distributed equally across the architecture objects, you must manually specify the cost allocation.

The costs must be explicitly allocated to the architecture objects via the **Reallocate Costs** button available in the **Objects** page view. Only then will they be added to the object and displayed in the *Operational Expenses Page View* for applications, deployments, networks, ICT objects, and service products, and the *Business Case Page View* for projects.

Please note the following if an object is added or removed from the cost center:

- The values for an equally distributed allocation will be automatically adjusted in the **Cost Allocation Percentage** column in the **Objects** page view.

- The values for a manually defined allocation will not be automatically updated for the remaining objects in the **Cost Allocation Percentage** column in the **Objects** page view. Therefore, the total sum of the allocation values will be less than 100. In this case you must explicitly redefine the allocation of the costs. For more information, see Manually Allocating Costs to the Objects Assigned to a Cost Center.
- Regardless of whether the allocation is specified to be equal allocation or not, the allocated costs will not be automatically updated in the **Operational Expenses** page views for applications, deployments, networks, ICT objects, and service products. and the **Business Case** page view for projects. In this case, you must explicitly redistribute the costs via the **Reallocate Costs** button in the **Objects** page view. For more information, see the section Allocating the Cost Center Costs to the Objects.

The authorized user of the cost center may add objects to the cost center to which he/she does not necessarily have Read/Write access permissions to. If the objects that can be added to the cost center should be limited in some way, a custom selector should be configured by your solution designer for this purpose. For more information about the configuration of custom selectors, see the section *Configuring a Custom Selector for Search Functionalities*.

The following information is available:

- Adding Applications, Deployments, ICT Objects, Projects, and Service Products to the Cost Center
- Manually Allocating Costs to the Objects Assigned to a Cost Center
- Removing Objects from a Cost Center
- Allocating the Cost Center Costs to the Objects
- Updating the Cost Allocation After a Batch Update via ADIF

Adding Applications, Deployments, ICT Objects, Projects, and Service Products to the Cost Center

To add an object to a cost center:

1) Go to the **Cost Centers** functionality in the **Configuration** module, navigate to the object profile of the relevant cost center and click **Objects**.
2) In the toolbar, click **New** > **Add<ObjectClass>**.
3) The object selector opens. Enter search criteria, as needed, and click **Search**.
4) Select the object(s) that you want to add to the cost center and click **OK**. The object is added to the cost center. You will see the following information about the object that has been added to the cost center:
   - **Name**: The name of the object assigned to the cost center.
   - **Type**: The object class of the object assigned to the cost center.
   - **Cost Allocation Percentage**: The percentage of costs allocated to the architecture object for the cost center. The total of the allocation for all objects should be 100. Please note the following:

- If the cost center is specified to equally distribute the costs among the architecture objects, the **Cost Allocation Percentage** column will be automatically updated to represent the equal distribution of costs. In this case, the cost proportions will always equal 100.
- If the cost center is not specified to equally distribute the costs among the architecture objects, the **Cost Allocation Percentage** column displays 0 for the cost proportion for any new object add to the cost center and the cost proportions may not equal 100. In this case, the cost allocation must be manually defined. For more information, see Manually Allocating Costs to the Objects Assigned to a Cost Center.

Manually Allocating Costs to the Objects Assigned to a Cost Center

In some cases, your company may decide that the costs should not be equally distributed among the objects assigned to a cost center. If this is the case, you must manually define the allocation of the costs for the objects assigned to the selected cost center. The sum of all cost proportions for a cost center must equal 100.

Please note the following:

- The **Edit Cost Allocation** button available in the toolbar opens an editor that allows you to manually define the distribution of the cost of the objects to the cost center. If the **Equal Distribution** checkbox in the **Cost Center** editor is selected, the **Edit Cost Allocation** button will not be available. In this case, the costs will be automatically distributed equally among the objects assigned to the cost center. If costs should not be equally distributed to the cost center, the **Equal Distribution** checkbox in the **Cost Center** editor should not be selected.
- If an object is added or removed from the cost center, the values for a manually defined allocation will not be automatically updated for the remaining objects and the total sum of the allocation values will be less than 100. In this case you must explicitly redefine the allocation of the costs via the **Edit Cost Allocation** button as described below.

To manually allocate costs to the objects assigned to the selected cost center:

1) In the toolbar of the **Objects** page view, click the **Edit Cost Allocation** button. The **Cost Allocation** editor opens.
2) Adjust the values by moving a selector to the left or right. To lock a value for an object, click the lock symbol on the left. When the lock is activated, it turns red and the value will no longer be automatically adjusted as you change the values of other objects. The sum of all costs for a cost center must equal 100.
3) When you have the correct distribution of costs across the objects assigned to the selected cost center, click **OK**. The values are automatically updated in the table.
4) To allocate the costs of the cost center to the architecture objects, click the **Reallocate Costs** button. Only then will the costs be distributed and displayed in the *Operational Expenses Page View* for applications, deployments, networks, ICT objects, and service products and the *Business Case Page View* for projects. For more information, see the section Allocating the Cost Center Costs to the Objects.

Removing Objects from a Cost Center

If you detach an application, deployment, network, ICT object, and service product, or project from a cost center, it will be removed from the cost center but remain in the Alfabet database.

Please note the following if an object is added or removed from the cost center:

- The values for an equally distributed allocation will be automatically adjusted in the **Cost Allocation Percentage** column in the **Objects** page view.
- The values for a manually defined allocation will not be automatically updated for the remaining objects in the **Cost Allocation Percentage** column in the **Objects** page view. Therefore, the total sum of the allocation values will be less than 100. In this case you must explicitly redefine the allocation of the costs. For more information, see Manually Allocating Costs to the Objects Assigned to a Cost Center.
- Regardless of whether the allocation is specified to be equal allocation or not, the allocated costs will not be automatically updated in the **Operational Expenses** page views for applications, deployments, networks, ICT objects, and service products. and the **Business Case** page view for projects. In this case, you must explicitly redistribute the costs via the **Reallocate Costs** button in the **Objects** page view. For more information, see the section Allocating the Cost Center Costs to the Objects.

To detach an object from a cost center:

1) In the **Objects** page view, click the object that you want to remove from the cost center.
2) In the toolbar, click the **Detach** button.
3) Confirm the warning by clicking **Yes,** or click **No** to exit without saving your changes.

Allocating the Cost Center Costs to the Objects

Once the cost center budget has been defined in the **Cost Accrual** page view and the objects have been assigned to the cost center, you can distribute the costs of the cost center to the architecture objects assigned to the cost center. Regardless of whether the allocation is specified to be equal allocation or not, changes to the cost proportions displayed in the **Cost Allocation Percentage** column in the **Objects** page view (for example, by adding an object to the cost center) will not be automatically updated in the *Operational Expenses Page View* for the architecture objects. In this case, you must explicitly redistribute the costs via the **Reallocate Costs** button in the **Objects** page view.

To do so, click the **Reallocate Costs** button in the **Objects** page view for the relevant cost center. The costs will be updated in the *Operational Expenses Page View* for applications, deployments, networks, ICT objects, and service products and the *Business Case Page View* for projects.

Updating the Cost Allocation After a Batch Update via ADIF

It may be that your enterprise has configured a batch process to update the cost definition and object assignment of a cost center as well as the cost allocation proportions. Such an update may be configured to be executed regularly via a batch process. This batch execution is configured via an ADIF import scheme in the Alfabet Data Integration Framework.

However, Alfabet does not support the automatic distribution of costs from the cost center to applications, deployments, ICT objects, projects and service products after the batch update of costs. Costs must be explicitly reallocated to the objects in the user interface.

To update the imported cost allocation to the objects assigned to the cost center, click the **Reallocate Costs** button in the **Objects** page view for the relevant cost center. The costs will be updated in the *Operational Expenses Page View* for applications, deployments, networks, ICT objects, and service products and the *Business Case Page View* for projects.

For more information about the configuration of an ADIF import scheme to update cost definitions, object assignments and cost allocation proportions, see the section *Import Scheme to Automatically Reallocate Cost Center Costs* in the reference manual *Alfabet Data Integration Framework*.

Chapter 6: Configuring Objects for the Object Class Generic Reference Data

The object class `GenericReferenceData` allows objects representing classifications to be captured and provides an alternative to enumerations to capture custom data. Typical classifications that already exist in Alfabet are the reference data types **Connection Types**, **Connection Methods**, **Connection Frequencies**, and **Connection Data Formats**.

For example, the object class `GenericReferenceData` could allow accountability levels for strategic decisions (Direct), management checks (Control), and business actions (Execute) such as those in the IBM® Business Capability Model to be captured for domains/business capabilities in Alfabet. Object class stereotypes must be configured for the object class `GenericReferenceData` to capture each relevant classification system. A custom property should then be configured for the relevant object class to allow the data for the object class stereotype to be captured. In the example object, for example, the object class stereotypes Direct, Control, and Execute would be configured for the class `GenericReferenceData` and a custom property would be defined for each object class stereotype on the object class Domain (which is used to capture data for business capabilities).

The full range of configuration options such as custom properties (including properties of the type `Reference` and `ReferenceArray`), custom editors, wizards, object views, selectors, and configured reports to analyze and understand the information are available. For more information about configuring object class stereotypes and custom properties for the class `GenericReferenceData`, see the sections *Configuring Object Class Stereotypes for Object Classes* and *Configuring Custom Properties for Protected or Public Object Classes*. in the reference manual *Configuring Alfabet with Alfabet Expand*.

The objects for the object class stereotypes can be created in the **Generic Reference Data** page view in the **Reference Data** functionality.

To create a connection method:

1) Go to the **Reference Data** functionality and click **Generic Reference Data**.
2) In the toolbar, click **New** > **Create Generic Reference Data**. The **Stereotype Selector** opens.
3) Select the relevant object class stereotype that you want to create objects for and click **OK**.
4) The **Generic Reference Data** editor opens. Enter information into each field, as required.
   - **Name**: Enter a unique name for the generic reference data object.
   - **Description**: Enter a meaningful description that will clarify the purpose of the generic reference data object.
5) Click **OK** to save the generic reference data object or **Cancel** if you do not want to save it. The generic reference data object will be available in the relevant drop-down field in the relevant custom editor.

To delete a generic reference data object, select the connection method in the table and click the **Delete**  button.

If you delete an object in the object class `GenericReferenceData`, it will be irrevocably deleted from the Alfabet database.

Chapter 7: Configuring the Connection Data for Information Flows

When creating and defining information flows, the connection method, connection type, connection frequency and connection data format of the information flow can be captured in the **Information Flow** editor. The options that users can select in the drop-down fields available for these attributes may vary for an enterprise. Therefore, Alfabet supports the customization of the options that users can select.

The following information is available:

Configuring the Options Available to Define the Connection Method

A connection method describes the method of transfer used by a specific information flow to transfer business data between the two associated applications or their respective components. Examples include TCP/IP, file transfer, message queue.

You can create as many connection methods as needed.

To create a connection method:

1) Go to the **Reference Data** functionality and click **Connection Methods**.
2) In the toolbar, click **New** > **Create New Connection Method**. The **Connection Method** editor opens.
3) Enter information into each field, as required.
   - **Name**: Enter a unique name for the connection method.
   - **Description**: Enter a meaningful description that will clarify the purpose of the connection method.
4) Click **OK** to save the connection method or **Cancel** if you do not want to save it. The connection method will be available in the **Connection Method** drop-down field in the **Information Flow** editor.

To delete a connection method, select the connection method in the table and click the **Delete** button.

If you delete an object in the object class Connection Method, it will be irrevocably deleted from the Alfabet database.

Configuring the Options Available to Define the Connection Type

A connection type describes the mode of transfer used by a specific information flow to transfer business data between the two associated applications or their respective components. Examples include batch and online.

You can create as many connection types as needed.

To create a connection type:

1) Go to the **Reference Data** functionality and click **Connection Types**.
2) In the toolbar, click **New** > **Create New Connection Type**. The **Connection Type** editor opens.
3) Enter information into each field, as required.
   - **Name**: Enter a unique name for the connection type.
   - **Description**: Enter a meaningful description that will clarify the purpose of the connection type.
4) Click **OK** to save the connection type or **Cancel** if you do not want to save it. The connection type will be available in the **Connection Type** drop-down field in the **Information Flow** editor.

To delete a connection type, select the connection type in the table and click the **Delete**  button.

If you delete an object in the object class Connection Type, it will be irrevocably deleted from the Alfabet database.

Configuring the Options Available to Define the Connection Frequencies

A connection frequency describes how often a specific information flow is used to transfer business data between the two associated applications or their respective components. Examples include daily, monthly.

You can create as many connection frequencies as needed.

To create a connection frequency:

1) Go to the **Reference Data** functionality and click **Connection Frequencies**.
2) In the toolbar, click **New** > **Create New Connection Frequency**. The **Connection Frequency** editor opens.
3) Enter information into each field, as required.
   - **Name**: Enter a unique name for the connection frequency.
   - **Description**: Enter a meaningful description that will clarify the purpose of the connection frequency.
4) Click **OK** to save the connection frequency or **Cancel** if you do not want to save it. The connection frequency will be available in the **Connection Frequency** drop-down field in the **Information Flow** editor.

To delete a connection frequency, select the connection frequency in the table and click the **Delete**  button.

If you delete an object in the object class Connection Frequency, it will be irrevocably deleted from the Alfabet database.

Configuring the Options Available to Define the Connection Data Formats

A connection data format describes the data format used for the transfer of business data via a specific information flow. Examples include ASCII, XML.

You can create as many connection data formats as needed.

To create a connection data format:

1) Go to the **Reference Data** functionality in the **Configuration** module and click **Connection Date Format**.
2) In the toolbar, click **New** > **Create New Connection Data Format**. The **Connection Data Format** editor opens.
3) Enter information into each field, as required.
   - **Name**: Enter a unique name for the connection data format.
   - **Description**: Enter a meaningful description that will clarify the purpose of the connection data format.
4) Click **OK** to save the connection data format or **Cancel** if you do not want to save it. The connection data format will be available in the **Connection Data Format** drop-down field in the **Information Flow** editor.

To delete a connection data format, select the connection data format in the table and click the **Delete**  button.

If you delete an object in the object class Connection Method, it will be irrevocably deleted from the Alfabet database.

Chapter 8: Configuring Role Types to Define Roles in the Responsibilities Page View

In Alfabet, the ability to retrieve information as well as edit data about an object are controlled by access permissions that have been defined for individual objects. Access permissions determine visibility and editability of data for users. The access permission concepts implemented in Alfabet are described in detail in the section *Understanding Access Permissions in Alfabet* in the reference manual *Getting Started with Alfabet*.

In contrast to the notion of access permissions, Alfabet also allows responsibilities for objects to be defined via roles.

A role defines the functional relationship or responsibility that a user or organization has to an object. A role is based upon a configured role type that is configured for an object class. Roles are defined for informational purposes only and provide detail about users or organizations that may have information about or a stake in the object. The definition of a role for an object does not impact access permissions.

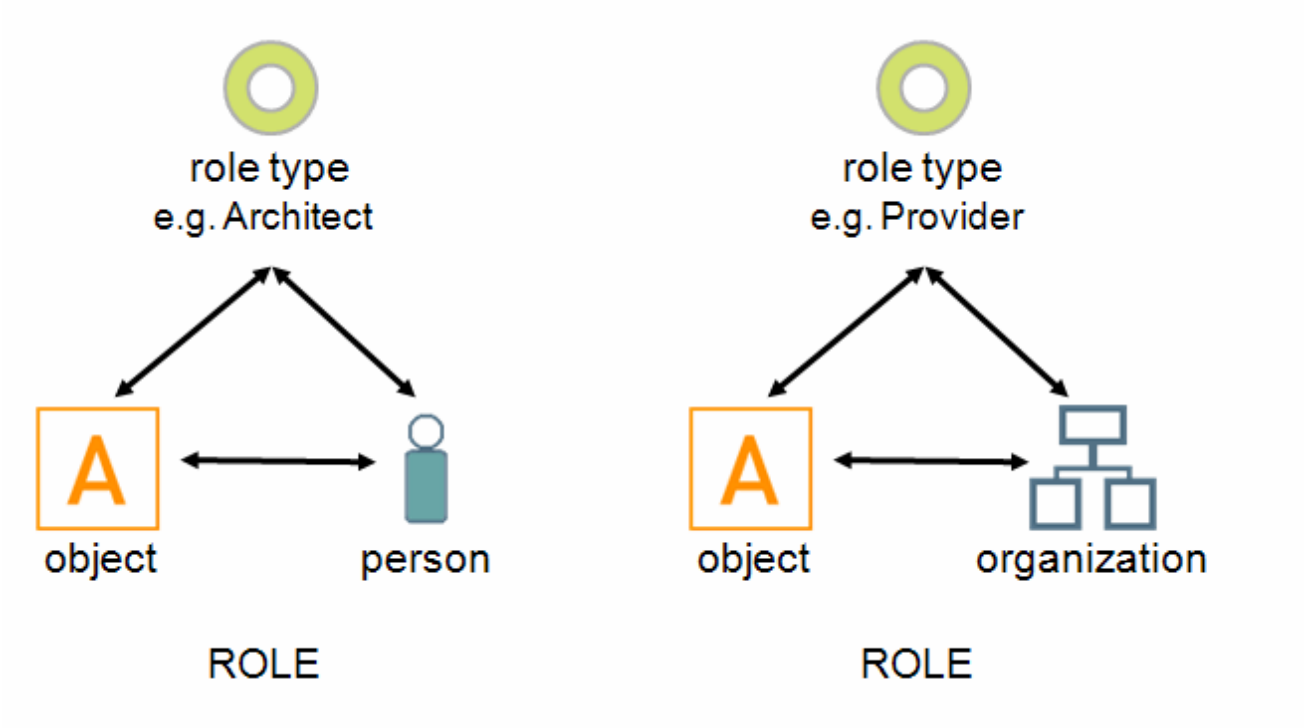


FIGURE: Example of the possible definitions of a role

A role for an object is defined as follows in Alfabet: A user with Read/Write access permissions to the object can define a role for the object in the *Responsibilities Page View*. Unlike an object's authorized user, a person or organization with a role defined for an object has no direct responsibility to maintain the data for an object and may not even have Read/Write access permissions to that object.

The *Responsibilities Page View* will display all users and organizations defined to have a role for the selected object as well as contact information such as a telephone number or email address. Users will be able to view all objects for which they have a role defined for in the *Role Objects Page View*.

In order for users to be able to create roles in the *Responsibilities Page View*, you must create role types and assign them to the relevant object class. Only the role types that have been configured for the object class of the selected object will be available for the role definition. Depending on the configuration of the role types that the roles are based on, a role may be defined for either a person and/or organization. The

configuration of the role type will also determine whether only one person or organization or multiple persons or organizations may be defined per role type for the selected object.

For example, the object class Application has 4 role types defined. The role types Architect, Business Analyst, and Project Manager can be defined as roles for persons. The role types Business Analyst and Provider can be defined as roles for organizations. In this case, both persons and organizations can be assigned a role based on the role type Business Analyst. Furthermore, the configuration can stipulate that whereas many persons can be assigned the role of Business Analyst for an application, only one organization may be assigned the role Business Analyst for an application

A default role type can be configured for an object class to streamline the definition of roles. If a default role type is defined for an object class, the default role will be automatically assigned to a new object when it is created. The role type can be changed by the user in the *Responsibilities Page View*, as needed. Only one default role type is allowed per object class.

The following provides an overview of the configuration steps necessary so that roles can be defined by users in the *Responsibilities Page View*:

- Create all role types in the **Reference Data** functionality. This is explained below in the section Configuring Role Types.
- Assign the role types to the relevant object classes in the **Class Configuration** functionality and specify whether the role type should be used to create roles for persons and/or organizations. This is explained below in the section Assigning Role Types for Persons and Organizations to Object Classes. For an overview of the object classes for which role types can be configured, see Appendix: Class Configuration for Object Classes.
- Define a default role type for the relevant object classes in the **Class Configuration** functionality. This is explained below in the section Defining Default Role Types for an Object Class.
- By default, a role type may be assigned multiple times to an object class in the **Class Configuration** functionality. If you want the role type to be unique for an object class - in other words, the role type should only be assigned once per object class, then you should configure a unique key for the object class `RoleTypeConfig` in the configuration tool Alfabet Expand. The attributes `ClassConfiguration` and `RoleType` must be defined as the class key. For more information about the configuration of class keys, see the section *Configuring Class Keys for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*.

Alfabet supports a governed process for transitioning the roles that are responsible for objects. The process to transition roles can be specified on a class-by-class basis for the role types configured for the class. The user defining the role types can specify whether the change of the person/organization responsible for the role must be governed or not, define the workflow that governs the translation, and control the deletion of roles for the objects of the class.

The following information is available:

- Configuring Role Types
- Assigning Role Types for Persons and Organizations to Object Classes
- Defining Default Role Types for an Object Class

Configuring Role Types

A role type is configured in order to allow a functional role to be defined for objects in a specified object class. Role types can be configured to be explicitly available for a user or organization or both.

In the **Reference Data** functionality, you must create all role types that are relevant for your enterprise, regardless of whether they can be assigned to a person or organization.

To create a role type:

1) Go to the **Reference Data** functionality in the **Configuration** module and click **Role Types**.
2) In the toolbar, click **New** > **Create New Role Type**. The **Role Type** editor opens.
3) Enter information into each field, as required.
   - **Name**: Enter a unique name for the role type.
   - **Description**: Enter a meaningful description that will clarify the purpose of the role type. The description will be displayed as a tooltip in the *Responsibilities Page View* when users are defining roles for an object.
   - **Technical Name**: Enter a unique technical name for the role type.
4) Click **OK** to save the role type or **Cancel** if you do not want to save it. To make the role type available for definition for an object, you must assign it to the relevant object class. For more information, see the section Assigning Role Types for Persons and Organizations to Object Classes.

To delete a role type, select the role type in the table and click the **Delete**  button.

If you delete an object in the object class Role Type, it will be irrevocably deleted from the Alfabet database. If any of the following dependent objects or data is defined for the deleted object, these will also be deleted: Role.

Assigning Role Types for Persons and Organizations to Object Classes

You can assign an unlimited number of role types to an object class. Role types can be explicitly specified to be available for the definition of a person's role or an organization's role or both. For example, when a user defines a role for a person in the *Responsibilities Page View* they will see the role types that have been specified for persons in the **Person** drop-down list.

Furthermore, you can specify whether only one person or organization can be assigned a role based on the role type or whether multiple persons or organizations can be assigned the role. Users who attempt to define multiple persons for a role type when only one person is allowed for the role type will be informed via an informational message that only one person may be defined for the selected role type.

If NO role type is configured for an object class, then ALL existing role types will be available in the *Responsibilities Page View* for that object class (unless the *Responsibilities Page View* is not included in the solution configuration by your solution designer). For more information about defining roles, see the *Responsibilities Page View*.

Please note the following regarding the configuration of role types:

- Before you can assign a role type to a selected object class, you must create a role type in the **Role Types** page view in the **Reference Data** functionality.
- For an overview of the object classes for which role types can be configured, see Appendix: Class Configuration for Object Classes.

The process to transition roles can be specified on a class-by-class basis for the role types configured for the class. The user defining the role types can specify whether the change of the person/organization responsible for the role must be governed or not, define the workflow that governs the translation, and control the deletion of roles for the objects of the class.

The view displays all role types that have been defined for a selected class. The following columns are displayed:

- **Name**: Displays the name of the role type.
- **Person Available**: Displays a checkmark if this role type should be available for the definition of a role for a person.
- **Person Plurality**: Displays a checkmark if multiple persons may be defined per role type for a selected object.
- **Organization Available**: Displays a checkmark if this role type should be available for the definition of a role for an organization.
- **Organization Plurality**: Displays a checkmark if multiple organizations may be defined per role type for a selected object.
- **Allow Duplicated Roles**: Displays a checkmark if duplicate roles can be defined per person or organization.
- **Manage Transitions**. Displays whether the change of the person/organization responsible for the role is governed or not. The following values can be selected in the **Manage Transitions** field:
  - **No** The role transition is not governed.
  - **Mandatory**: The user may change the **Proposed Person** field in the **Role** editor in the *Responsibilities Page View*. Changes to the **Person** and **Role Type** fields in the **Role** editor are not allowed.
  - **Optional**: The user may change the **Person** and **Role Type** fields in the **Role** editor or the **Proposed Person** field in the **Role** editor for a person fulfilling the role. If an organization is fulfilling the role, the user may change the **Organization** and **Role Type** fields in the **Role** editor or the **Proposed Organization** field.
- **Role Transitioning Workflow**: Displays the configured workflow template that shall be triggered if a user attempts to transition the associated role.
- **Disable Deletion**: Displays a checkmark if the **Delete** button in the *Responsibilities Page View* will be disabled for a role based on the role type and the role cannot be deleted.

To assign a role type to a selected object class:

1) In the **Class Configuration** explorer tree, select the object class that you want to define a role type for and click **Role Types**.

2) In the **Role Types** page view, click **New** > **Add Role Type**.

3) The selector opens. Enter search criteria, as needed, and click **Search**.

4) Select a role type and click **OK** to add the role type to the selected object class. The role type has been added to the selected class.

5) The default settings are applied. The role type will therefore be available in both the **Persons** and **Organizations** drop-down menu in the *Responsibilities Page View* for objects in the selected object class. Furthermore, multiple roles can be defined for the selected role type and object for multiple persons and organizations. To modify the default settings, select the role type in the table and click the **Edit** button. The **Role Type Configuration** editor opens.

6) Edit the following fields, as needed.

- **Role Type**: Displays the name of the selected role type.
- **Class Configuration**: Displays the name of the object class that the role type is configured for.
- **Manage Transitions**. Specify whether the change of the person/organization responsible for the role is governed or not. The following values can be selected in the **Manage Transitions** field:
  - **No** The role transition is not governed and can be changed as needed in the *Responsibilities Page View*. This value will automatically be set for existing role types to ensure backward compatibility upon migration to Alfabet.
  - **Mandatory**: The user may change the **Proposed Person** field in the **Role** editor in the *Responsibilities Page View*. Changes to the **Person** and **Role Type** fields in the **Role** editor are not allowed.
  - **Optional**: The user may change the **Person** and **Role Type** fields in the **Role** editor or the **Proposed Person** field in the **Role** editor for a person fulfilling the role. If an organization is fulfilling the role, the user may change the **Organization** and **Role Type** fields in the **Role** editor or the **Proposed Organization** field. Please note that the **Proposed Person** (or **Proposed Organization**) field cannot be changed simultaneously when the **Person** / **Role Type** fields (or **Organization** / **Role Type** fields) are changed.
- **Role Transitioning Workflow**: If either **Mandatory** or **Optional** is selected in the **Manage Transitions** field, select a configured workflow template that shall be triggered if a user attempts to transition the associated role.

  Please note that the workflow should be configured to set the **Person** (or **Organization**) field in the **Role** editor to the **Proposed Responsible** field's value and to clear the **Proposed Responsible** field for the role upon successful completion of the workflow. For more information about configuring workflows, see the chapter *Configuring Workflows* in the reference manual *Configuring Alfabet with Alfabet Expand*.

- **Person Available**: Select if this role type should be available for the definition of a role for a person. If a checkmark is set, the role type will be displayed in the **Person** drop-down menu in the **Responsibilities** page view for objects in the selected object class.

  Please note that if the role type is specified as a default role type, the default role type will be applied regardless of the definition of the **Person Available** attribute.

- **Person Plurality**: Select if multiple persons may be defined per role type for a selected object.
- **Organization Available**: Select if this role type should be available for the definition of a role for an organization. If a checkmark is set, the role type will be displayed in the **Organization** drop-down menu in the **Responsibilities** page view for objects in the selected object class.

  Please note that if the role type is specified as a default role type, the default role type will be applied regardless of the definition of the **Organization Available** attribute.

- **Organization Plurality**: Select if multiple organizations may be defined per role type for a selected object.
- **Allow Duplicated Roles**: Select if duplicate roles can be defined per person or organization.
- **Disable Deletion**: If the checkbox is selected, the **Delete** button in the *Responsibilities Page View* will be disabled for a role based on the role type and the role cannot be deleted. The property is set to `False` per default.

7) Click **OK** to save your changes or click **Cancel** to exit without saving.

Defining Default Role Types for an Object Class

If a default role type is defined for an object class, the default role will be automatically assigned when a new object is created. The role type can be changed by the user in the *Responsibilities Page View*, as needed. Only one default role type is allowed per object class.

Please note that if the role type is specified as a default role type, the default role type will be applied regardless of the **Person Available** definition.

Specifying a default role type for an object class will NOT make it available in the list of role types displayed in the *Responsibilities Page View*. In order to allow the default role type to be explicitly defined for a person or organization, you must add it to the object class via the **Role Types** page view in the **Class Configuration** functionality. Nevertheless, if the role type is specified as a default role type, the default role type will be applied to the role definition for an object regardless of the definition of the **Person Available** or **Organization Available** attributes. For more information, see the section Assigning Role Types for Persons and Organizations to Object Classes.

To assign a default role type to a selected object class:

1) In the **Class Configuration** explorer tree, select an object class that you want to define a default role type for and click **Default Role Type**.
2) In the toolbar, click **New** > **Set Default Role Type**.
3) The selector opens. Enter search criteria, as needed, and click **Search**.
4) Select a role type and click **OK** to assign the default role type to the selected object class, or click **Cancel** to exit without saving your changes.

Chapter 9: Configuring Reference and Evaluation Data Required for Project Management

The project management capability in Alfabet allows users to plan and monitor a project and its sub-projects. Project managers can request the skills that are needed for the project as well as set project milestones in order to ensure that the project meets relevant deadlines and will be completed on time. Thus, when projects are planned and sub-projects created in the *Project, Skill Request and Resource Request Time Schedule Page View* or *Project, Skill Request and Resource Request Time Schedule (Gantt) Page View*, the relevant skill requests and milestones can be added to the sub-project.

In the **Configuration** module, you can create project templates that bundle a set of configured skill requests and milestones so that the same set of skills and milestones are passed on to all relevant projects in the project hierarchy. The default values defined for a skill such as the head count needed or the cost of the skill can be changed by the project responsible as needed. Likewise, the default dates for milestones can also be adjusted for the individual project.

A project template is useful if your project planning includes the planning of resources and milestones. For example, if resources (skill requests) will not be defined for projects, you do not necessarily need to define project templates because a milestone template can also be assigned directly to a project in the *Project Milestones Page View*.

To configure the relevant date to plan resources in the context of project management, you must configure the following:

- Ensure that the relevant milestone templates have been configured by your solution designer. A milestone template bundles a set of milestones including the definition the sequence and tolerance period for each milestone. Projected target dates are automatically set for each milestone based on the configuration defined in the milestone template, but these dates may be manually edited by the responsible project manager. Milestone templates must be configured in the configuration tool Alfabet Expand. For more information, see *Configuring Milestone Templates to Track Projects* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- Create a project template and assign the relevant milestone template to it. This is carried out in the **Project Templates** page view in the **Reference Data** functionality.
- Create skills that may be requested as resources needed for a project. This is carried out in the **Skills** page view in the **Reference Data** functionality.
- Create one or more skill request templates for the project template. The skill request template is a template for a skill that is requested by a user for a project. The skill request template specifies such issues as the head count or man days required to deliver the skill, the cost types for which the skill should be budgeted, and organization that will provide the skill. The responsible project manager can later change these values, as needed. The definition of skill requests is carried out in the **Skill Request Templates** page view for a project template in the **Reference Data** functionality.

The following information is available.

- Configuring Project Evaluation Types
- Configuring Cost Management Capabilities for Project Management

Configuring Project Templates to Track Milestones and Skill Requests for Projects

A project template consists of a configured milestone template, which bundles a set of milestones, and a skill request template, which groups a set of skills. A project template can be assigned to a project and its sub-projects in order to ensure that the project hierarchy has a consistent definition of milestones and skill requests.

When you create a project template, you assign a milestone template to it that has been configured by your solution designer in the configuration tool Alfabet Expand. A milestone template bundles a set of milestones including the sequence and tolerance period for each milestone. The milestone template can be assigned to the project via the project template in the *Project, Skill Request and Resource Request Time Schedule Page View* or *Project, Skill Request and Resource Request Time Schedule (Gantt) Page View*.

Milestone templates and their milestones must be configured by your solution designer in the configuration tool Alfabet Expand. For more information, see *Configuring Milestone Templates to Track Projects* in the reference manual *Configuring Alfabet with Alfabet Expand*.

To create a project template:

1) Click the **Reference Data** explorer icon and click **Project Templates**.
2) In the toolbar, click **New** > **Create New Project Template**. The **Project Template** editor opens.
3) Enter information into each field, as required:
   - **Name**: Enter a unique name for the project template.
   - **Select Milestone Template**: Select the milestone template that you want to associate with the project template.
   - **Currency**: Select the currency or currency unit to assign to a new project that is based on the selected project template. The currency/currency unit may be changed in the **Project** editor, as needed. For more information about the configuration of currencies, see the chapter Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.
   - **Project Template Description**: Enter a meaningful description that will clarify the purpose of the project template.
4) Click **OK** to save the project template or **Cancel** if you do not want to save it.

To delete a project template:

If you delete an object in the object class Project Template, it will be irrevocably deleted from the Alfabet database.

Configuring Skills for Skill Requests

A skill is a field of expertise necessary to fulfill a specific aspect of a project. The skill can be assigned to a skill request template that specifies the head count or man days and the expected cost required by the skill in the context of a particular kind of project. Users making a skill request thus have predefined information that helps to estimate the resources and costs involved if the skill is implemented in the project.

Not all skills required to realize a project need to be estimated. Rather, the skills that the enterprise deems scarce should be projected when preparing for the submission of project for prioritization and budgeting.

All skills that you define can later be requested via a skill request in the context of planning resources for projects.

To create a skill:

1) Click the **Reference Data** explorer icon and click **Skills**.
2) In the toolbar, click **New** > **Create New Skill**. The **Skill** editor opens.
3) Enter information into each field, as required.
   - **Name**: Enter a unique name for the object.
   - **Description**: Enter a meaningful description that will clarify the purpose of the object.
   - **Foreground Color** Specify the foreground color of the skill.

     Skill requests will be displayed with the color defined for the skill that they are based on and the color definition of a skill will be consistently applied to all Gantt views displaying skill requests. If no colors have been defined in the **Skill** request editor, the specification of the **Foreground Color** and **Background Color** attributes of the relevant class setting for the class Skill.

   - **Background Color** Specify the background color of the skill.
4) Click **OK** to save the skill or **Cancel** if you do not want to save it.

To delete a skill, select the skill in the table and click the **Delete** button.

If you delete an object in the object class **Skill**, it will be irrevocably deleted from the Alfabet database. If any of the following dependent objects or data is defined for the deleted object, these will also be deleted: **Skill Request**, **Skill Offer**.

Configuring Skill Request Templates

A skill request template allows skills to be specified in order to plan resources for a project. A skill request template specifies the head count or man days required for the skill and the expected cost required by the skill in the context of a particular kind of project. Users making a skill request thus have predefined information that helps to estimate the resources and costs involved if the skill is implemented in the project. Skill request templates can be assigned to a project template in order to streamline the process of project planning.

The costs that are defined for the selected project's skill requests may be aggregated to the project's business case. For more information, see the *Business Case Page View* for the selected project.

Skills must already be created in order to define a skill request template. For more information about creating skills, see the section Configuring Skills for Skill Requests.

You can create multiple skill request templates for a selected project template:

1) Click the **Reference Data** explorer icon and click **Project Templates**.
2) Select the project template that you want to create skill request templates for and click the **Navigate** button.
3) In the object profile of the project template, click **Skill Request Templates**.
4) In the toolbar, click **New** > **Create New Skill Request Template**. The **Skill Request Template** editor opens.
5) Enter information into each field, as required:

- **Skill**: Select the skill that needs to be provided in order to fulfill the skill request.

  Skills must first be configured in the **Reference Data** functionality in the **Configuration** module before you can define skill requests for a project template. For more information, see Configuring Skills for Skill Requests.

- **Task**: Enter a short comment clarifying the task required to fulfill the skill request. It is important to define the task as this is used in other views and reports to identify the work done by the person executing the skill.
- **Count**: Enter a non-negative number representing either the total number of people or the number of man days needed to fulfill the skill request. The value you enter here will depend on your definition of the **Capacity Type** field. Values with decimal point/decimal place are allowed.

  If the **Capacity Type** attribute is set to **Head Count** and a person has been defined in the **Providing Person** field, it is possible to define the **Count** field to a number higher than one. If **Capacity Type** = **Head Count**, then ultimately you should split the skill request in multiple tasks in the **Project Resource Planning** page view and specify each task to be provided by a different person. If **Count** is greater than 1 but the task is assigned to only one person, then the task will be displayed in the **My Tasks and Time Reporting** functionality of the providing person with a count that is greater than 1. This may be problematic when the user attempts to track hours worked on the task. For more information about the functionalities available for users to track their hours worked on tasks, see the chapter *Managing Your Calendar, Tasks, and Timesheets* in the reference manual *Getting Started with Alfabet*.

- **Capacity Type**: Select either **Head Count** to define the total number of people needed to fulfill the skill request or **Man Days** to define the number of man days needed to fulfill the skill request. The value you enter here will determine your definition of the **Count** field.

  Please consider the following regarding the cost calculation of skill requests:

  - For skill requests with number of man days defined (**Man Days** is specified in the skill request's **Capacity Type** field). The cost calculation is based on the **Count** value multiplied by the **Daily Rate** value. (Count *DailyRate).

- For skill requests with number of persons defined (**Head Count** is specified in the skill request's **Capacity Type** field): The cost calculation is based on the **Count** value multiplied by the number of working days (minus Saturdays and Sundays) between the start date and end date of the resource request multiplied by the **Daily Rate** value. (Count *WorkDays*DailyRate).

  Please note that blockout days defined by the providing person are not taken into consideration in the calculation of the cost of a skill request for which the **Capacity Type** is set to **Head Count**. For more information about the functionalities available for users to define personal calendars and track hours worked on tasks, see the chapter *Managing Your Calendar, Tasks, and Timesheets* in the reference manual *Getting Started with Alfabet*.

- Scroll to the right of the dataset to view and enter **Comment** in the comments field.

- **Currency**: Select the default currency or currency unit to capture and analyze skill request template costs. The currency/currency unit may be changed if necessary in the **Skill Request** editor when it is being created for a project. For more information about the configuration of currencies, see the chapter Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.
- **Daily Rate**: Enter the daily cost of the skill for the cost type you select in the **Cost Type** field.
- **Providing Organization**: Select the organization that is expected to provide the skill needed to fulfill the skill request.

  In the selector that opens when you click the **Search** button, you will see a **Providing Organizations** tab. The tab contains all organizations defined to provide skills for the project. The organizations are defined in the *Providing Organizations Page View* of a subordinate project.

- **Cost Type**: Select a preconfigured cost type to capture the costs of the skill request.

  Only cost types that have been specified via the **Show for Skill Request/Resource Requests** in the relevant **Cost Type** editor will be displayed in the drop-down list. For more information about making cost types available for skill requests, see the chapter Configuring Cost Types and Income Types for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

- **Description**: Enter a meaningful description that will clarify the purpose of the skill request.

6) Click **OK** to save the project template or **Cancel** if you do not want to save it.

Configuring Project Evaluation Types

A project evaluation type is an evaluation type that has been configured to allow project monitoring to occur for projects. The projects can be monitored regarding the achievement of target values for the indicator types assigned to the project evaluation type.

You must first create a conventional evaluation type in the **Evaluation Types** page view in the **Evaluations and Portfolios** functionality as explained in the section Configuring Evaluation Types. The evaluation types you create can then be assigned as project evaluation types that you assign to the object class Project via the **Project Evaluation Type** page view in the **Class Configuration** functionality. The project evaluation types will then be available in the *Project Tracking Overview Report Page View* for projects, project groups, and buckets.

Please note that the project evaluation types will not be available in the *Project Tracking Overview Report Page View* if the project evaluation types are assigned to an object class stereotype (Project:Project:Stereotype). The project evaluation types must be assigned to the base class Project.

You can assign an existing evaluation type as a project evaluation type to the object class Project. To assign a project evaluation type:

1) Go to the **Class Configuration** functionality and click the object class Project.
2) Click **Project Evaluation Type**.
3) In the toolbar, click **New** > **Add Evaluation Type**.
4) The selector opens. Enter search criteria, as needed, and click **Search**.
5) Click an evaluation type and click **OK** to add the evaluation type as a project evaluation type to the object class Project or click **Cancel** to exit without saving your changes. The evaluation type is displayed in the table.

Configuring Cost Management Capabilities for Project Management

In order to work with the cost management capabilities in conjunction with project management, you must configure the necessary cost types and income types required by your enterprise to track project costs. Additional non-operational cost types (such as capital expenditures) as well as income types (benefits) can be specified. Non-operational cost types and income types are used to define business cases as well as to capture skill requests (resource costs) for projects. The following page views display operational and non-operational cost types and income types:

Object Class	Page View
Project	*Cost Report Page View*
Project	*Cost Accrual Page View*

Object Class	Page View
Project	*Business Case Page View*
Project	*Business Case Comparison Page View*
Project	*Project, Skill Request and Resource Request Time Schedule Page View*
Project	*Cash Out Planning Page View*
Bucket	*Cost Accrual Page View*
ICT Object	*Lifecycle Costs Page View* and *Lifecycle Costs Chart Page View* (CAPEX costs will be displayed in the views if the ICT object or its associated applications are defined in the **Primary Architecture Element** field in the **Project** editor)

The following must be configured to work with the cost management capabilities in conjunction with the project management capabilities

- All cost types including OPEX and CAPEX cost types as well as income types that are relevant to the definition of business cases for project must be configured in the **Reference Data** functionality. This is explained in the sections Creating Cost Types and Cost Type Hierarchies and Creating Income Types.
- If you plan to track operational costs for projects, you must assign the operational cost types to the class **Project** in the **Maintenance Cost Types** page view in the **Class Configuration** functionality. This is explained in the section Making Maintenance Cost Types Available to Architecture Objects.
- If you plan to specify business cases and skill request costs for projects as well as CAPEX (non.operational) costs for projects:
  - You must assign the project cost types to the class **Project** in the **Cost Types** page view and you must assign income types to the class `Project` in the **Income Types** page view in the **Class Configuration** functionality. This is explained in the section Assigning Cost Types to the Object Class Project.
  - Your solution designer must configure the XML object ***CostManagerDef*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Business Case Definition for Projects* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- To use any cost management capability available in Alfabet:
  - The fiscal year for your enterprise must be configured by your solution designer in the XML object ***CostManagerDef*** in the configuration tool Alfabet Expand. For more information, see the section *Configuring the Fiscal Year for Cost Reporting in Your Enterprise* in the reference manual *Configuring Alfabet with Alfabet Expand*.

- The currency and currency units displayed on all page views and editors in which costs are captured and visualized is configured in the **Reference Data** functionality. For more information about the configuration of currencies, see the chapter Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

Chapter 10: Configuring Data Retention Policies

A data retention policy supports the storage and management of business data. Data retention policies provide a means to document standard information about how business data should be retained and stored including the amount of time that business data should be retained, rules for archiving the business data, permissible means to store and access the business data, and the levels permissible to encrypt the business data.

The configured data retention policies can be assigned to the relevant business data in the **Business Data Usage** editor in the *Business Data Page View*.

For an overview of the methodological approach to working with data retention policies as part of the Information Risk Management capability in Alfabet, see the chapter *Information Risk Management* in the reference manual *IT Governance, Risk and Compliance*.

To create a data retention policy:

1) Go to the **Reference Data** functionality.
2) Click **Data Retention Policies**.
3) In the toolbar, click **New** > **Create Data Retention Policy**. The **Data Retention Policy** editor opens.
4) Enter the information in the fields, as needed.

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each object in the inventory. This number cannot be edited.
   - **Name**: Enter a unique name for the data retention policy.
   - **Short Name**: Enter a short name for the data retention policy.
   - **Description**: Enter a meaningful description that will clarify the purpose of the data retention policy.
   - **Data Retention Time**: Select the retention time. The available values are configured by your enterprise by means of the enumeration `DRP_Time`.
   - **Data Retention Start**: Enter descriptive text explaining when the data retention policy shall begin.
   - **Archival Rule**: Select the rule to archive the data retention policy. The available values are configured by your enterprise by means of the enumeration `DRP_ArchivalRule`.
   - **Permissible Means of Storage**: Select one or more means of storage that is permissible for the data retention policy. The available values are configured by your enterprise by means of the enumeration `DRP_Storage`.
   - **Permissible Means of Access**: Select one or more means of access that are permissible for the data retention policy. The available values are configured by your enterprise by means of the enumeration `DRP_Access`.
   - **Minimum Encryption Level**: Select the minimum level of encryption for the data retention policy. The available values are configured by your enterprise by means of the enumeration `DRP_EncryptionLevel`.

The enumerations required for the data retention policy are configured in the configuration tool Alfabet Expand. For more information, see the section *Defining Protected and Custom Enumerations* in the reference manual *Configuring Alfabet with Alfabet Expand*.

5) Click **OK** to save your changes.

Chapter 11: Configuring Third-Party Data Security Warnings

The **Third-Party Service Data Securing Warnings** page view allows a security warning to be displayed in the AlfaBot warning users that data input may potentially be used by a third-party service. A static message can be configured to be displayed with information about the security issue and a link to additional relevant information. The warning box will be embedded in the AlfaBot below the header and will remain visible throughout the time that the AlfaBot is open.

The security warning consists of a caption, up to 2 texts blocks, and a link URL and link caption. Only one data security message may be configured for the AlfaBot. The data security warning is only displayed in the language that it is captured and will not be translated if the user interface is rendered in another language.

The border, font, link, and text colors to be specified for the security warning message embedded in the AlfaBot can be specified via the attributes available in the **Third-Party Service Data Securing Warning** section of the GUI scheme. For more information, see the section *Configuring GUI Scheme Definitions for the Alfabet Interface* in the reference manual *Configuring Alfabet with Alfabet Expand* and the section *Overview of GUI Scheme Attributes* in the reference manual *Configuring Alfabet with Alfabet Expand - Appendix*.

For more information about the configuration of the AlfaBot, see the section *Configuring the AlfaBot Capability* in the reference manual *User and Solution Administration*.

6) In the toolbar, click **New** > **Create Security Warning**. The **Third-Party Data Security Warning** editor opens.
7) Define the following fields as needed:
   - **Use Case**: Displays `AlfaBot`.
   - **Show Accept Button**: Select the checkbox if the **Accept** button shall be displayed.
   - **Caption**: Enter a caption to be displayed for the third-party security warning.
   - **Text**: Enter text to display in the first block of text for the third-party security warning.
   - **Link Text**: Enter text to display for the URL.
   - **Link URL**: Enter a URL that provides additional information about the third-party security warning.
   - **Text 2**: Enter text to display in the second block of text for the third-party security warning.
8) Click **OK** to save the third-party data securing warning.

Chapter 12: Configuring Color Rules for Map Views and Diagram Views

A color rule is based on one or more queries. Each query finds objects and highlights them with a specified color. Color rules can be assigned to a map view in order to color objects displayed in the map view's *Business Support Map Report* or to a diagram view in order to color objects in standard Alfabet diagrams.

A color rule defined for applications, for example, could be displayed in both the business support map as well as in relevant diagrams. Individual color rules can be grouped in a color rule group, which makes it simpler for users to activate and execute the color rules.

Color rules could be specified to highlight different types of applications displayed in the **Standard Application Diagram**. In this case, for example, color rules could be defined to highlight 1) applications of the type Client Server, 2) applications of the type eBusiness, and 3) applications of the type Mainframe.

The following configuration is required to define and implement color rules:

- Create a color rule in the **Color Rules** view available via the root node of the **Master Plans** explorer or **Diagram Views** explorer or in the **Color Rules Manager** functionality in the **Configuration** module. For more information, see the section Creating a Color Rule.
  - Color rules for map views could target applications, ICT objects, solution building blocks, providing organizations, tactical business supports, strategic business supports, and operational business supports.
  - Color rules for diagram views could target any object class that is relevant for the type of diagram that the diagram view is configured for. This is specified in the **Type** attribute defined for the diagram view in the **Diagram View** editor. For example, if **Application** is selected in the **Type** attribute for the diagram view, a color rule for application diagrams could target applications, solution applications, peripherals, or application groups. (Please note that a color rule is not meaningful for information flows.)
- Activate the color rule. Once a color rule has been specified, the color rule must be either manually activated or activated via a batch process so that the query can be executed. The query should be re-activated periodically in order to update the query results and include changes made to the database.
  - For more information about the manual activation of the color rule, see the section Activating the Color Rule and Executing the Query.
  - For more information about activation via a batch process, see the section *Batch Evaluation of Color Rules with RescanColorRules.exe* in the reference manual *System Administration*.
- To implement the color rule for a map view, the relevant color rules must be assigned to the map view in the **Map View** editor in the *Business Support Map Report*. Additionally, the **Use Color Map** option must be checked in the **Business Support Map Options** editor available in the view. The color rules displayed in the view will be automatically added to the legend. If necessary, individual tactical business supports or strategic business supports can be hidden. For more information about the implementation of

color rules in the relevant pages for map views, see the section *Highlighting Matrix Objects with Color* in the reference manual *IT Planning Basic*.

- To implement the color rule for a diagram view, the relevant color rules must be assigned to each diagram view for which it is relevant. This is done in the **Color Rules** tab in the **Diagram View** editor of the respective diagram view in the **Diagram Views** functionality in the **Configuration** module. For more information about assigning a color rule to a diagram view, see *Defining Color Rules to Display on Diagram View Items*.

The following information is available:

Creating a Color Rule

Color groups and color rules can be created in either the **Color Rules Manager** functionality in the **Configuration** module or the **Color Rules** page view available in either the **Diagram Views** explorer in the **Configuration** module or via the root node of the **Master Plan** explorer. All color rules created for your enterprise are available to any user accessing any of the views in which color rules can be created or assigned to a relevant object.

Please consider the following regarding color rules:

- Color rules for map views should target applications, ICT objects, solution building blocks, providing organizations, tactical business supports, strategic business supports, and operational business supports.
- Color rules for diagram views should target one of the object classes that is relevant for the type of diagram that the diagram view is configured for. This is specified in the **Type** attribute defined for the diagram view in the **Diagram View** editor.

The following columns are displayed:

- **Group**: The name of the color rule group. Structuring color rules in color rule groups makes it easier for users to find and select the correct color rule in enterprise's where many color rules have been defined.
- **Rule**: The name of the color rule.
- **Color**: The color representing the color rule. Matrix objects and diagram objects found by the associated query will be filled with this color.
- **Is Active**: Displays an X if the color rule has been activated and the query has been executed.
- **Activation Date**: Displays the most recent date that the color rule has been activated and the query executed. The query should be executed regularly in order to update the results.
- **Objects Found**: Displays the number of objects found in the Alfabet database by the queries associated with the color rule. The number of objects can be updated by reactivating the color rule.

When you define a color rule, you must define a query to find the relevant set of objects that should be highlighted and the color that should be applied to the objects found via the query. A rule may consist of several queries.

To define a color rule:

1) In the toolbar, click **New** > **Create New Color Rule**. The **Color Rule** editor opens.
2) Enter the information in the fields, as needed.

   Basic Data tab:

- **Color Rule Name**: Enter a name for the color rule.
- **Color**: Select the color that represents the color rule. Objects found by the associated query will be highlighted with this color in map view matrices and standard diagrams if relevant matrix or diagram view settings have been defined.
- **Color Rule Group**: Select a color group in which to structure the color rule. Structuring color rules in color rule groups makes it easier for users to find and select the correct color rule in enterprise's where many color rules have been defined.
- **Color Rule Description**: Provide a meaningful description about the color rule. It is recommended that you provide information about the class that the Alfabet query addresses as well as the purpose of the query.

   Queries tab:

- **Queries**: Enter one or more Alfabet queries or native SQL queries in an XML object for the color rule. The number of objects found by the queries after they have been executed will be displayed in the **Objects Found** column.

  The XML object consists of a root element `Queries` with one or multiple child elements `Query`, each defining a query that finds objects of a defined object class. The element `Query` has two attributes:

  - ***ClassName***: Enter the name of the object class for that objects are found in the query. The class name specification must be identical with the Name attribute of the object class. This name may deviate from the caption of the object class displayed on the Alfabet interface. For example, the name of the class Organisation is `OrgaUnit`. Please note the following:
    - If the color rule is for a map view, one of the following classes should be specified in the `ClassName` attribute: `Application`, `ICTObject`, `SystemBuildingBlock` (for solution building blocks), `OrgaUnit` (for providing organizations), `BusinessSupport` (for operational business supports), `TacticalBusinessSupport`, `StrategicBusinessSupport`.
    - If the color rule is for a diagram view, the `ClassName` attribute would target one of the object classes that is relevant for the type of diagram that the diagram view is configured for. This is specified in the **Type** attribute defined for the diagram view in the **Diagram View** editor. For example, if **Application** is selected in the **Type** attribute for the diagram view, one of the following classes should be specified in the `ClassName` attribute of the XML element `Query`: `Application`, `SolutionApplication`, `Peripheral`, `ApplicationGroup`.(Please note that a color rule is not meaningful for the class `InformationFlow`.)

- ***Query***: Define an Alfabet query or native SQL query that finds objects of the object class defined with the attribute `ClassName`. Please note the following:
  - When the query contains special characters (for example, a greater then (>) or lesser then (<) symbol) you must replace them with the respective HTML code, for example:

    ```
    > for >

    < for <

    &quot; for "
    ```

  - In Alfabet queries, the `FIND` class must be identical to the class specified with the attribute `ClassName`. `SHOW` and `SORT` properties may not be specified.
  - In native SQL queries, the `REFSTR` attribute of the class specified with the attribute `ClassName` must be defined as only column in the `SELECT` clause defining the query result.
  - For detailed information about defining Alfabet queries or the special rules that apply to the definition of native SQL queries in the context of your solution configuration, see the section *Defining Queries* in the reference manual *Configuring Alfabet with Alfabet Expand*.

For example, the Alfabet queries to find applications and solution applications for which the attribute Primary Domain has been set to a defined domain, for example a domain with the name "Account Management" would thus be written as follows:

```
<Queries>
   <Query ClassName="Application" Query="
   ALFABET_QUERY_500 FIND Application WHERE
   Application.Domain = 'Account Management' " />
   <Query ClassName="SolutionApplication" Query="
   ALFABET_QUERY_500 FIND SolutionApplication WHERE
   SolutionApplication.Domain = 'Account Management' " />
</Queries>
```

3) Click **OK** to save your changes.

Creating Color Groups to Structure Color Rules

You can create color groups in which to bundle color rules. Bundling color rules in color rule groups makes it easier for users to find and select the correct color rules in enterprise's where many color rules have been defined.

Color groups are displayed in the **Map View** editor and **Diagram View** editor. All color rules are listed below the color rule group they are assigned to. Users can select the entire color rule group for the relevant map view or diagram view or select individual color rules.

To create a color group:

1) In the toolbar, click **New** > **Create New Color Rule**. The **Color Rule** editor opens.
2) Enter the information in the fields, as needed.

Basic Data tab:

- **Color Rule Group Name**: Enter a name for the color rule group. The color group name should provide information to users to help them understand the grouping of color rules.
- **Description**: Enter a meaningful description that will clarify the purpose of the color rule group.

3) Click **OK** to save your changes.

Activating the Color Rule and Executing the Query

Once a color rule is defined, it must be activated in order for the query to be initially executed.

If changes are made to objects, these changes will not be automatically updated to the query results. The query should be re-activated periodically in order to update the query results and include changes made to the Alfabet database. The color rule may be manually activated or activated via a batch process.

For more information about activation via a batch process, see the section *Batch Evaluation of Color Rules with RescanColorRules.exe* in the reference manual *System Administration*.

To manually activate a color rule, select the color rule in the table and click **New** > **Activate Rule**. Please keep the following in mind:

- If objects have been found by the query, an X will be displayed in the **Is Active** column, the date will be displayed in the **Activation Date** column, and the number of found objects will be displayed in the **Objects Found** column.
- If no objects have been found by the query, the **Is Active** and **Activation Date** columns will remain empty and 0 will be displayed in the **Objects Found** column.

Deactivating the Color Rule

To deactivate the color rule so that it no longer highlights found objects in the map view matrices or Alfabet diagrams, select the color rule in the table and click **New** > **Deactivate Rule**. The number 0 will be displayed in the **Objects Found** column for the color rule. If you reactivate the color rule, the query will be executed, and the number of objects found will be updated.

Chapter 13: Creating Generic Attributes for an Object Class/Object Class Stereotype

A generic attribute allows the ad-hoc capture of content in a semi-structured form for a specific application, component, deployment, device, standard platform, stack, deployment element, standard platform element, stack element, or stack configuration item. A generic attribute can be used if one or more attributes are required for informational purposes only and each attribute is only used for a small subset of objects rather than for all objects in an object class. A generic attribute is an alternative to configuring a custom property for an object class.

The **Generic Attributes** page view available in the **Class Configuration** functionality allows generic attributes to be created for a selected object class/object class stereotype. Generic attributes created for the base class that the object class stereotype is based on may be reused by the object class stereotype.

The generic attributes created for the class/stereotype can be reused when generic attributes are created for an object in the **Generic Attributes***Generic Attributes Page View* (`ObjectGenericAttributes` page view in the standard object profile of the relevant class as well as for the object class stereotype (if configured). The default value specified for a reused generic attribute can be changed, as needed.

The table displays all generic attributes that have been defined for the selected object class/object class stereotype. Each column in the table is defined below:

- **Name**: The name of the generic attribute.
- **Value**: The default value defined for the generic attribute for the selected object class or object class stereotype.
- **Type**: The type of property that the generic attribute is.
- **Generic Attribute Group**: The generic attribute group that the generic attribute belongs to.

For an overview of the configuration required to work with generic attributes, see the section *Implementing the Generic Attribute Concept Instead of Custom Properties* in the reference manual *Configuring Alfabet with Alfabet Expand*.

The following information is available:

- Creating a New Generic Attribute for the Object Class/Stereotype
- Copying an Existing Generic Attribute to the Object Class/Stereotype
- Deleting a Generic Attribute from the Selected Object

Creating a New Generic Attribute for the Object Class/Stereotype

You can create multiple generic attributes for the selected object class or object class stereotype.

1) In the toolbar, click **New** > **Create New Generic Attribute**. The **Generic Attribute** editor opens.
2) Enter information into each field, as required.
   - **Name**: Enter or name for the generic attribute.

- **Type**: Select the type of property that the generic attribute is. The field will display all possible property types as well as permissible enumerations.
- **Generic Attribute Group**: Enter a name for or select an existing name of the generic attribute group that the generic attribute belongs to.
- **Value**: Enter or select a default value that is relevant for the generic attribute for the selected object class or object class stereotype. The values that may be specified will depend on the type of property defined in the **Type** field. The default value specified for a reused generic attribute can be changed, as needed.

  Please note that if you select `Date` in the **Type** field, a placeholder for a non-existing date will be displayed indicating the date format to be used.

3) Click **OK** to save the generic attribute or **Cancel** if you do not want to save it.

Copying an Existing Generic Attribute to the Object Class/Stereotype

You can add an existing generic attribute that has been created for another object class or object class stereotype. All generic attributes created for the object class or object class stereotype will be copied to the selected object class or object class stereotype. You can change the default value of the generic attributes or delete unwanted generic attributes for the selected object class or object class stereotype, as needed.

1) In the toolbar, click **New** > **Copy Existing Generic Attributes Here**. The object selector opens.
2) Select the object class or object class stereotype for which the generic attribute is defined that you are copying to the selected object class or object class stereotype and click **OK**.
3) Confirm the information message by clicking **OK**. All generic attributes defined for the object class or object class stereotype will be copied.
4) To change the default value of a generic attribute for the selected object class or object class stereotype, select the generic attribute and click the **Edit** button.
5) In the **Generic Attribute** editor, edit the **Type** field as needed. The default value will be changed for the selected object class or object class stereotype but not for the original object class or object class stereotype that the generic attribute was copied from.

Deleting a Generic Attribute from the Selected Object

If you delete a generic attribute, the generic attribute will be deleted from the selected object but not from other objects that it is assigned to.

To delete a generic attribute:

1) In the table, click the generic attribute that you want to delete.
2) In the toolbar, click the **Delete** button.
3) Confirm the warning by clicking **Yes,** or click **No** to exit without deleting the selected object(s).

Chapter 14: Configuring Diagram Views for Diagrams

A diagram view is a configuration that is associated with a diagram. It allows users to superimpose qualitative information - such as aggregated indicators or attribute values - associated with these architectural elements. Diagram views can be implemented in diagrams displaying applications, business processes, devices, domains, frameworks, platforms, and solution building blocks. Multiple diagram views may be defined and made available for a diagram. A diagram view may be reused for multiple diagrams.

The **Diagram Views** functionality allows you to create and define diagram views. When you create a diagram view, you specify the type of diagram that is relevant for the diagram view. For example, if you specify **Application** for the **Type** attribute of the diagram view, then the diagram view will be available for users to select in the **Diagram View** field in application-centric diagrams, such as the **Standard Application Diagram** and **Information Flow Diagram**.

For custom diagrams, additional diagram views can be created as part of the configured diagram view report for opening the custom diagram. These diagram views are part of the report configuration and are only available for custom diagrams displayed via the configured diagram report they have been created for. They are selectable for the custom diagram in addition to any diagram views defined via the **Diagram Views** functionality. for more information about the diagram views created as part of a configured diagram view report, see *Creating Custom Diagram Views for a Custom Diagram View Report* in the reference manual *Configuring Alfabet with Alfabet Expand*.

The **Diagram Views** functionality displays an explorer showing existing diagram views. Keep the following in mind when working with the **Diagram Views** explorer:

- Click the diagram views root node  to display the view in which you can define new diagram views.
- Click any diagram view  in the explorer to access the **Items** page view in which you can define diagram view items.

A diagram view item is the configuration that specifies which information should be displayed about the objects visualized for a selected diagram view in a Alfabet diagram. The diagram view item captures the attributes and indicators that should be displayed for the object. The specified attributes and indicators are then displayed on the objects in the diagrams, which are usually visualized as rectangles or lines, if the objects are a connection item such as information flows, roles, or rules.

For example, a diagram view Risk & Standardization has been configured in order to understand the level of standardization and criticality for applications in the enterprise. The diagram view has been configured for diagrams of the type Application and thus is available in the **Diagram View** field in application landscape diagrams such as the **Standard Application Diagram** and **Information Flow Diagram**.

Application landscape diagrams typically display application groups, applications, peripherals, and information flows. Application groups, applications, and peripherals are displayed as diagram items in the diagram and the information flows are displayed as connecting lines between the relevant applications/peripherals. For this diagram view, diagram view items have been configured for applications and information flows, but not for peripherals or application groups as the analysis of these objects is not of interest in this context for the enterprise.

Unless otherwise configured, the diagram view item for applications automatically shows the Application icon and the application's short name (upper left corner) and the version and object state (upper right corner). In the example below, the diagram view item for applications has been configured to also include the attributes Name, SOX Criticality (True or False), Start Date and End Date and the indicators Criticality and Standardization Status. The diagram view item for

information flows has been configured to display the indicators Interface Criticality and Interface Reliability.

Additionally, a color rule has been specified for the diagram view highlighting applications that have the Application Type attribute specified as Mainframe. The meaning of the data displayed on the diagram view item including the attributes, indicator icons, and color rules is explained in detail in the diagram's legend.

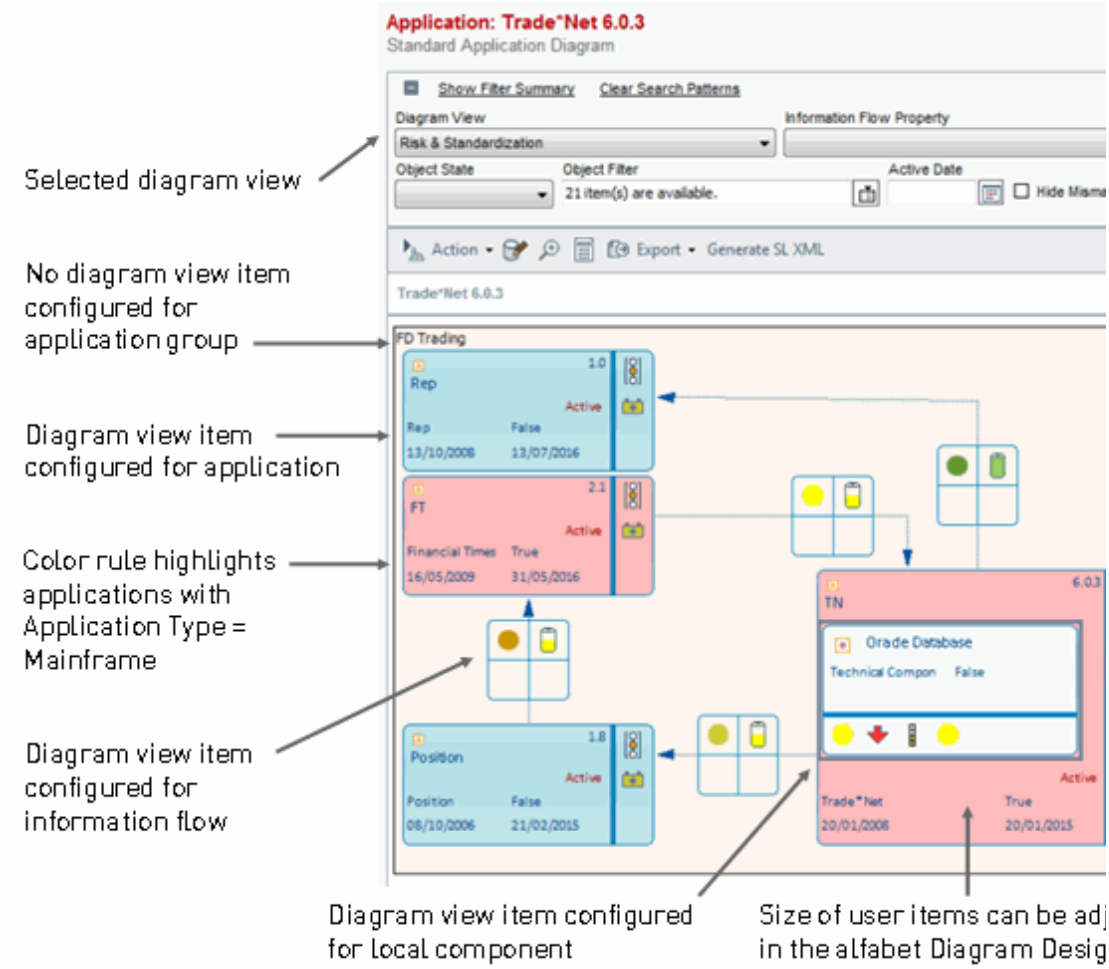


FIGURE: Example of diagram view items configured for the diagram view Risk & Standardization

To create and implement diagram views, follow these steps:

- Configure the diagram view items required for the diagram view. This requires the following:
  - Determine which object classes you want to display specific information about. To understand this, you must decide which Alfabet diagrams you will create the diagram view for and which object classes are potentially displayed in the relevant diagrams. For an overview of all diagrams in which the **Diagram View** field is displayed and the object classes potentially displayed in each diagram, see the section Understanding the Object Classes Relevant for the Diagram View Configuration. For an overview of the object classes for which diagram view items can be configured, see Appendix: Class Configuration for Object Classes. It may be that your enterprise has configured custom diagram item templates that you can

use to base your diagram view items on. For more information about the configuration of diagram item templates, see the section *Configuring Custom Diagram Item Templates* in the reference manual *Configuring Alfabet with Alfabet Expand*.

- Ensure that all indicator types that are relevant for the diagram view items have been configured and assigned to evaluation types. This is done in the **Evaluations and Portfolios** functionality. The evaluation types must be assigned to the relevant object classes in the **Class Configuration** functionality. For more information about configuring indicator types and evaluation types, see the chapter Configuring Evaluations, Prioritization Schemes, and Portfolios.
- Create the diagram view items for the relevant object classes required for the diagram view in the **Class Configuration** functionality. At this point, you must decide which standard or custom diagram item template to use. The diagram item template specifies the graphic information and content to display on the diagram item. For more information about the standard diagram item templates provided by Software AG, see the section Understanding Diagram Item Templates in the Diagram View Item Definition. You can also specify up to four attributes and four indicators to display on the diagram item template.

- Configure the diagram view in the **Diagram Views** functionality.
  - Create the diagram view and, based on the Alfabet diagrams that you are creating the diagram view for, specify the diagram type in the **Type** field. The diagram type associated with each Alfabet diagram is described in the section Understanding the Object Classes Relevant for the Diagram View Configuration. For more information about creating the diagram view, see the section Creating the Diagram View.
  - If color rules are to be implemented in conjunction with the diagram view, ensure that they are relevant for the diagram view items that have been configured. For more information about the configuration and implementation of color rules, see the chapter Configuring Color Rules for Map Views and Diagram Views.
  - Assign the diagram view items to the diagram view. For more information, see the section Assigning Diagram View Items to the Diagram View.

The following information is available:

Creating Diagram View Items

The **Diagram View Items** page view in the **Class Configuration** functionality allows you to create and assign diagram view items to a selected object class. The diagram view items you create here are then assigned to a diagram view, which users can select when working with diagrams.

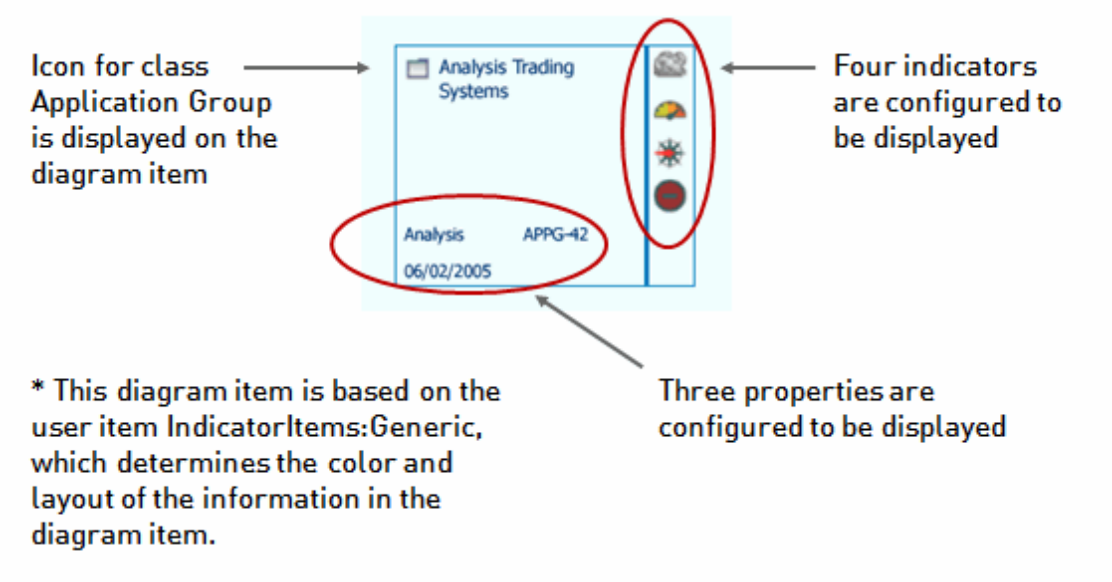


FIGURE: Visualization of a diagram view item configured for the class Application Group

A diagram view may have one diagram view item per object class assigned to it. You do not need to create a diagram view item for every object class displayed in a diagram. A diagram view item defines the attributes and indicators that are to be visualized for objects belonging to an object class. Attributes of the type `Reference` or `ReferenceArray` cannot be configured to be displayed in the diagram view item.

For example, a diagram view that is defined for an application group diagram may have only one diagram view item that is assigned to the class Application, one diagram view item that is assigned to the class Information Flow, and one diagram view item that is assigned to the class Local Component, and no diagram view item configured for the class Peripheral.

Diagram view items defined for an object class stereotype in the **Class Configuration** functionality will be ignored. Diagram view items may only be assigned to the object class that the stereotype is based on. For example, if application stereotypes such as Application:Business Application and Application:Technical Application have been defined, the diagram view items must be assigned to the base class Application. In this case, any diagram view assigned to the object class stereotypes Application:Business Application and Application:Technical Application would be ignored and not available in application diagrams.

To create a diagram view item for a diagram view and assign it to an object class, you must do the following:

- Ensure that all indicator types that are relevant for the diagram view items have been configured and assigned to evaluation types. This is done in the **Evaluations and Portfolios** functionality. The evaluation types must be assigned to the relevant object classes in the **Class Configuration** functionality. For more information about configuring indicator types and evaluation types, see the chapter Configuring Evaluations, Prioritization Schemes, and Portfolios.

- Create the diagram view items for the relevant object classes required for the diagram view in the **Class Configuration** functionality.
- Assign the diagram view items to the existing diagram view. For more information, see the section Assigning Diagram View Items to the Diagram View.

The following information is available:

- Understanding the Object Classes Relevant for the Diagram View Configuration
- Understanding Diagram Item Templates in the Diagram View Item Definition
- Creating a Diagram View Item for a Selected Object Class
- Deleting a Diagram View Item from a Selected Class

Understanding the Object Classes Relevant for the Diagram View Configuration

Before you configure a diagram view, you must first consider the diagrams in which you want the diagram view to be available. Then you need to determine which object classes available in the diagram require diagram view items. The table below lists the names of all diagrams that display the **Diagram View** field. The table allows you to determine which object classes are relevant for a diagram so that you can determine which object classes you want to configure diagram view items for. You can configure a diagram view item for one or more object classes, as needed. In order to make the diagram view bundling the relevant diagram view items available for the desired diagram, you must set the **Type** attribute of the diagram view as stated in the table.

For example, if you want to analyze the cost efficiency of domains in diagrams, you need to understand which diagrams display domains and which of these diagrams are based on the diagram type Domain.

In the table below, four diagrams display domains: *Standard Domain Application Diagram*, *Additional Domain Application Diagrams Page View*, *Standard Domain Function Diagram Page View*, and *Additional Domain Function Diagrams Page View*. According to the table, the diagrams *Standard Domain Application Diagram* and *Additional Domain Application Diagrams Page View* are based on the diagram type Application and thus actually display application landscapes relevant to a selected domain. The diagrams *Standard Domain Function Diagram Page View* and *Additional Domain Function Diagrams Page View* are based on the diagram type Domain and thus display domain landscapes. In this case, you would therefore define a diagram view with the **Type** attribute set to Domain in the **Diagram Views** functionality and you would then define a diagram view item with cost efficiency indicators and assign the diagram view item to the object class Domain in the **Class Configuration** functionality. Because the diagram view is created for the diagram type Domain, the diagram view would only be available in the diagrams *Standard Domain Function Diagram Page View* and *Additional Domain Function Diagrams Page View*.

If you wanted to analyze the cost efficient of domains in the context of the application landscape diagrams, you can do so by configuring a diagram view with the **Type** attribute set to Application and you would then define a diagram view item with cost efficiency indicators and assign the diagram view item to the object class Domain. Because the diagram view is created for the diagram type Application, the diagram view would be available in all application landscape diagrams including the diagrams *Standard Domain Application Diagram* and *Additional Domain Application Diagrams Page View*. However, the diagram view item configured for domains would only be displayed in the diagrams *Standard Domain Application Diagram* and *Additional Domain Application Diagrams Page View* because domains are displayed only in these application landscape diagrams.

Please note that if a diagram view is specified for the diagram type Application, then that diagram view will be available for all diagrams that are based on the diagram type Application. In other words, the diagram view will be available in the **Diagram View** field for the following application-centric diagrams: *Standard Application Diagram Page View*, *Additional Diagrams Page View*, *Information Flow Diagram Page View*, *Interface System Diagram Page View*, *Solution Maps and Application Diagrams Page View*, *As-Is Architecture Diagram Page View*, *Migration Diagram Page View*, and *Additional Domain Application Diagrams Page View*.

The following table provides information about the object classes displayed in diagrams and thus the object classes for which diagram view items may be configured. The table also indicates how the Type attribute must be defined in the diagram view to make the diagram view items available for the specific diagram:

To Configure a Diagram View For:	Configure Diagram View Items For:	Set "Type" Attribute of Diagram View To:
Standard Application Diagram Page View	Application Group Application Peripheral Information Flow Local Component	Application
Additional Diagrams Page View	Application Group Application Peripheral Information Flow Local Component	Application
Information Flow Diagram Page View	Application Information Flow Local Component	Application
Interface System Diagram Page View	Application Component Information Flow	Application
Solution Maps and Application Diagrams Page View	Application Solution Application Peripheral Local Component	Application

To Configure a Diagram View For:	Configure Diagram View Items For:	Set "Type" Attribute of Diagram View To:
	Solution Local Component Information Flow Solution Information Flow	
As-Is Architecture Diagram Page View	Application Peripheral Local Component Information Flow	Application
Migration Diagram Page View	Application ICT Object Solution Building Block Migration Rule	Application
Standard Domain Application Diagram	Domain Application Local Component Peripheral	Application
Additional Domain Application Diagrams Page View	Domain Application Local Component Peripheral	Application
Solution Building Block Diagrams Page View	Solution Building Block Solution Information flow Peripheral	Solution Building Block (SBB)
Business Processes Diagram	Business Process Organization Role	Business Process

To Configure a Diagram View For:	Configure Diagram View Items For:	Set "Type" Attribute of Diagram View To:
	Business Process Information Flow Rule	
Static Device Diagram Page View	Device Device Group Device Information Flow	Device
Location Diagram	Location Device	Device
Standard Domain Function Diagram Page View	Domain Business Function	Domain
Additional Domain Function Diagrams Page View	Domain Business Function	Domain
Platform Diagrams Page View	Standard Platform Platform Element Local Component Platform Information Flow	Platform
Navigation Diagram PageView	Application Business Data Business Function Business Object Business Process Component Device Domain	Framework

To Configure a Diagram View For:	Configure Diagram View Items For:	Set "Type" Attribute of Diagram View To:
	ICT Object Market Product Master Platform Organization Peripheral Standard Platform Vendor Product	
Custom Diagrams		The values that can be selected in the **Type** field that are relevant for custom diagrams will be specified by your solution designer. The protected enumeration `CustomDiagramViewType` is available for the definition of diagram views displayed in the context of custom diagrams. The enumeration items defined for the `CustomDiagramViewType` enumeration can be selected in the **Type** field in the **Diagram View** editor.  The query specified for the `DiagramView` filter field for the the `DiagramViewReport` report must refer to the relevant standard or custom diagram view type. The diagram views based on the standard or custom diagram view type can selected in configured reports based on the `DiagramViewReport` report template.. For more information about configuring protected enumerations, see the section *Overview of Protected Enumerations* in the reference manual *Configuring Alfabet with Alfabet Expand - Appendix*. For more information about configuring custom diagrams, see the section *Configuring Custom Diagrams* in the reference manual *Configuring Alfabet with Alfabet Expand*.

For an overview of all object classes for which diagram view items can be configured, see Appendix: Class Configuration for Object Classes.

Understanding Diagram Item Templates in the Diagram View Item Definition

For each diagram view item you create for a selected object class, you will need to specify a diagram item template. A diagram item template is the graphic information to use to represent the diagram item in the diagram. Diagram items may be either shapes (usually visualized as rectangles), which typically represent applications, components, business processes, organizations, etc., or connection items which typically represent information flows, roles, rules, etc.

Standard diagram item templates are preconfigured by Software AG. However, it may be that your enterprise has configured custom diagram item templates that you can use to base your diagram view items on. The custom diagram item templates may look entirely different than the standard diagram item templates. They may have different coloring, icon, text and attributes displayed. They may even be configured to look like a graphic rather than a rectangle or other shape. For more information about the configuration of custom diagram item templates, see the section *Configuring Custom Diagram Item Templates* in the reference manual *Configuring Alfabet with Alfabet Expand*.

The following figures show the default attributes displayed on a standard diagram item template as well as the placement of the configured attributes and indicators. The following standard diagram item templates can be selected for most object classes.

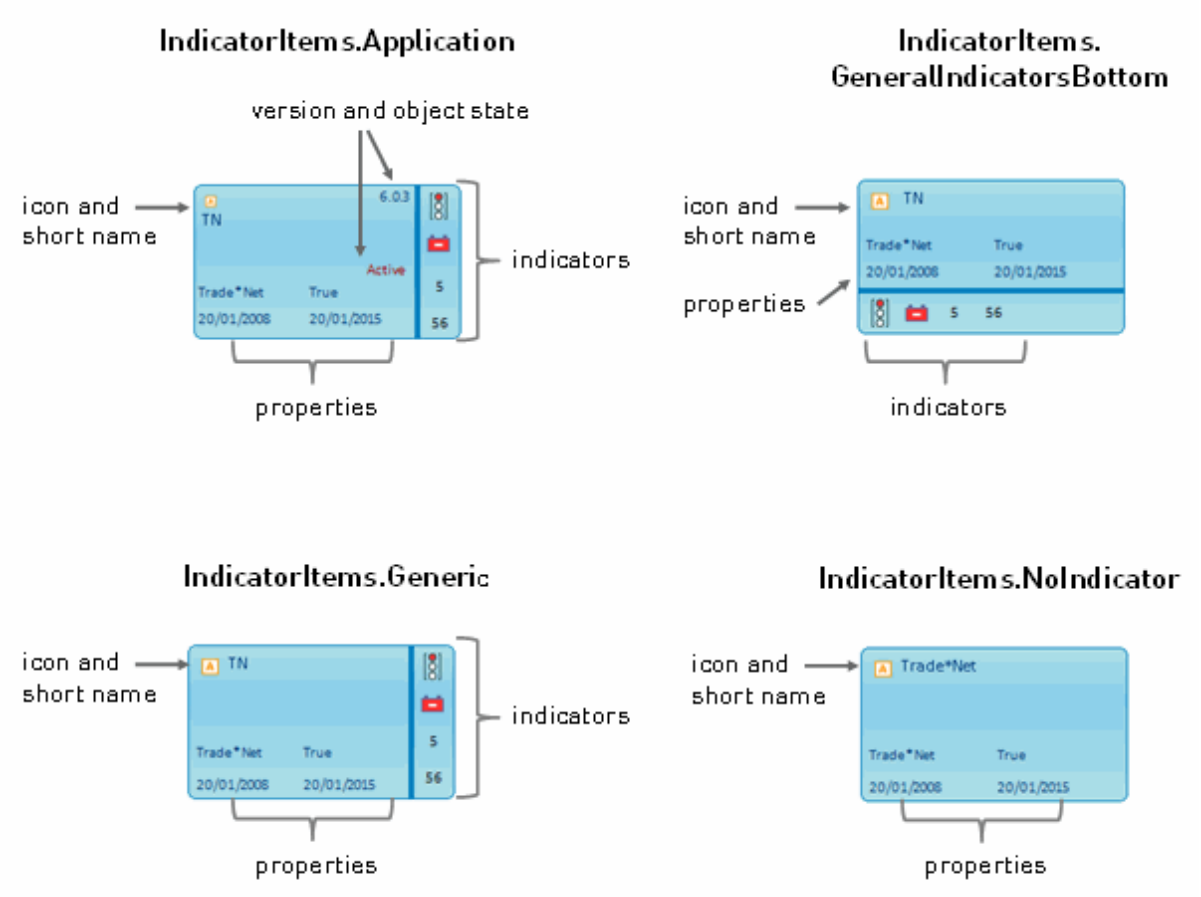


FIGURE: Standard diagram item templates available for architecture elements

The following standard diagram item templates can be configured for connector items such as information flows.

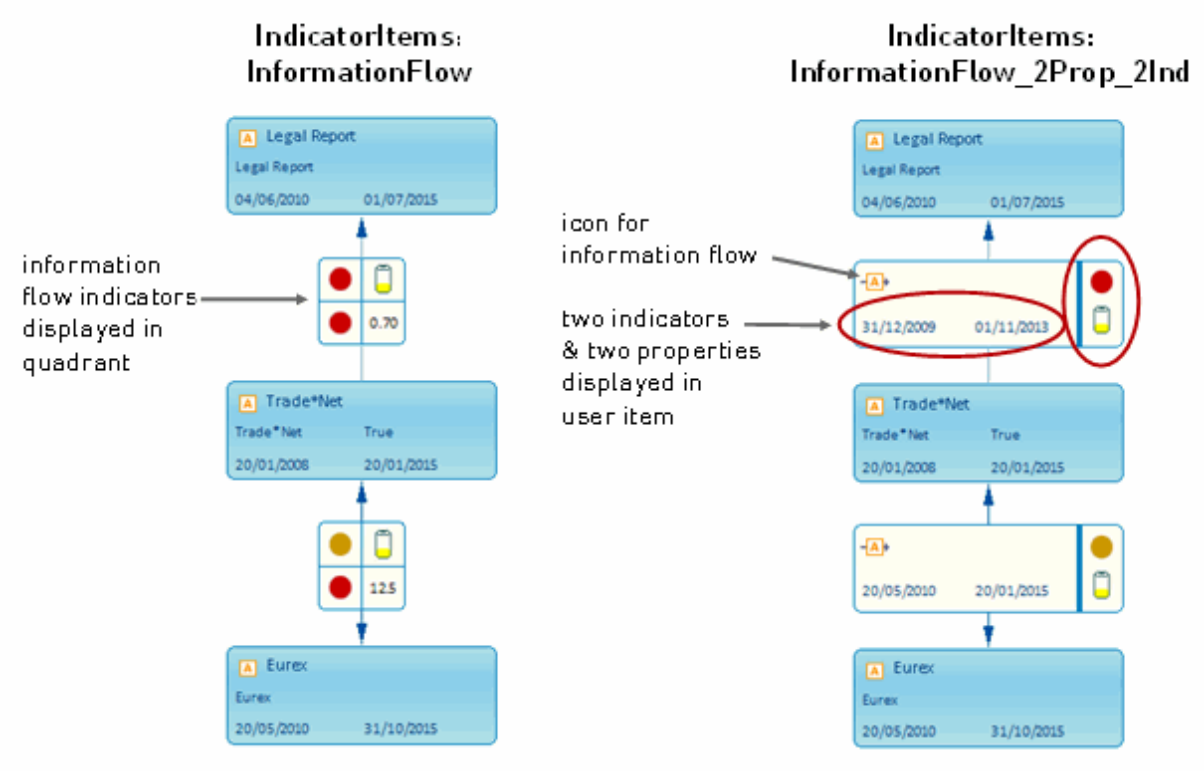


FIGURE: Diagram item templates available for connector items

The **Information Flow Attribute** field allows you to select a standard or custom attribute defined for information flows and display this text on the information flow. Please note that the **Information Flow Attribute** field available in most application-centric diagrams is displayed in the same place that the attribute information for the diagram view item is displayed. If you only want to display one attribute (and no indicators) for an information flow, then the **Information Flow Attribute** field in the page view will suffice and you will not need to define a diagram view item. However, if you also want to include additional attribute data via a diagram view item, then you should ensure that decoration boxes are explicitly designed and positioned off of the information flow. This is done in the context of the Alfabet Diagram Designer. For more information, see the section *Modifying the Decoration Boxes Displayed for Connection Items* in the reference manual *Designing IT Landscape Diagrams in Alfabet*.

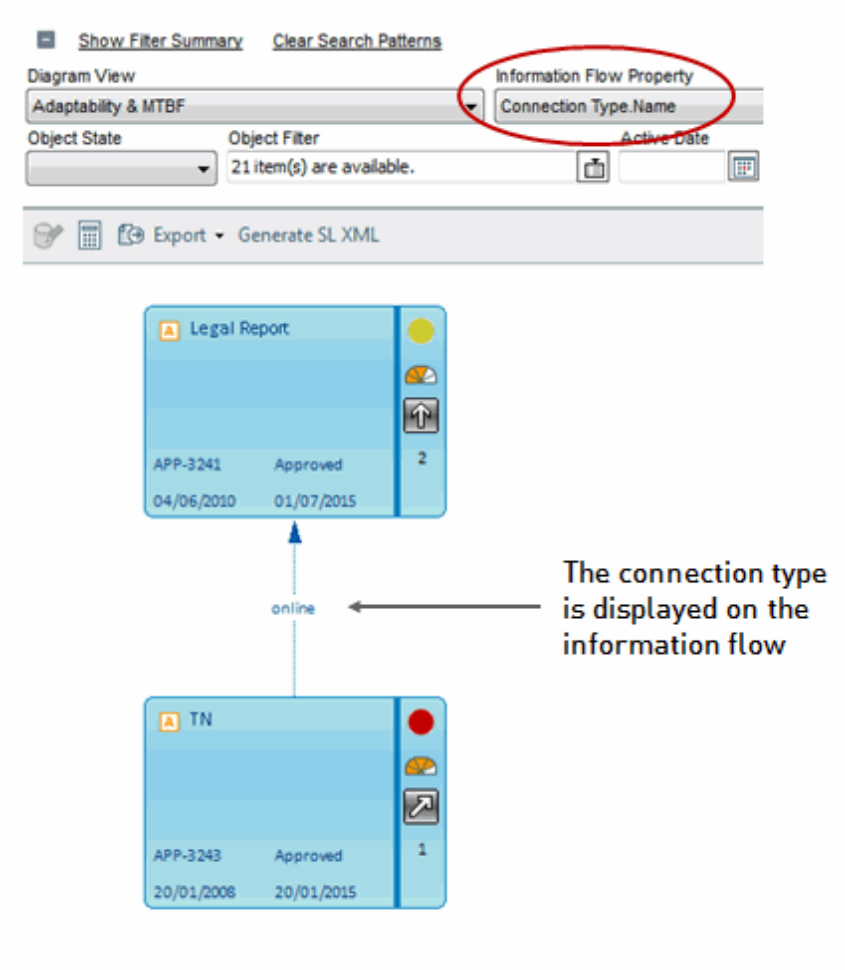


FIGURE: Displaying an information flow attribute via the Information Flow Property filter in the page view

The color of the standard diagram item templates will depend on the object class. Applications are colored azure blue per default, for example, whereas business processes are colored mint green. The color of a diagram item template as well as the size of the diagram item template can be modified in the Alfabet Diagram Designer. In order to ensure that enough space is available for the data to be displayed on the diagram item template, a minimum height of 40 mm should be configured for the diagram item by the diagram designer. The **Name** attribute will be displayed on one or more lines, depending on its length. For more information about the configuration of diagram item templates, see the section *Configuring Custom Diagram Item Templates* in the reference manual *Configuring Alfabet with Alfabet Expand*. For more information about designing diagrams, see the reference manual *Designing IT Landscape Diagrams in Alfabet*.

Creating a Diagram View Item for a Selected Object Class

A diagram view item is the configuration that specifies which information should be displayed about the objects visualized for a selected diagram view in a Alfabet diagram. The diagram view item captures the attributes and indicators that should be displayed for the object. The specified attributes and indicators are then displayed on the objects in the diagrams, which are usually visualized as rectangles or lines, if the objects are a connection item such as information flows, roles, or rules.

Diagram view items defined for an object class stereotype in the **Class Configuration** functionality will be ignored. Diagram view items may only be assigned to the object class that the stereotype is based on. For example, if application stereotypes such as Application:Business Application and Application:Technical Application have been defined, the diagram view items must be assigned to the base class Application. In this case, any diagram view assigned to the object class stereotypes Application:Business Application and Application:Technical Application would be ignored and not available in application diagrams.

To create a diagram view item for an object class:

1) Go to the **Class Configuration** functionality and click the object class that you want to create the diagram view item for. Refer to the table in the section Understanding the Object Classes Relevant for the Diagram View Configuration to understand the object classes that are relevant for the diagram view that you want to configure a diagram view item for.
2) Click **Diagram View Items**.
3) In the toolbar, click **New** > **Create New Item**. The **Diagram View Item** editor opens.
4) Enter the information into each field, as required. Each field is defined below:

   General tab:

   - **Name**: Enter a unique name for the diagram view item.
   - **Diagram Item Template**: Select a diagram item template in the drop-down list. The diagram item template determines the graphic information for the diagram view item. You may see standard diagram item templates predefined by Software AG as well as custom diagram item templates that are configured by your solution designer. For more information about what is displayed for each standard diagram item template, see the section Understanding Diagram Item Templates in the Diagram View Item Definition. For more information about the configuration of custom diagram item templates, see the section *Configuring Custom Diagram Item Templates* in the reference manual *Configuring Alfabet with Alfabet Expand*.
   - **Description**: Enter a meaningful description that will clarify the purpose of the diagram view item.

   Basic Data tab: This tab allows you to select up to four attributes to display on the diagram view item.

   - **Property 1 - 4**: Select a attribute to visualize on the diagram view item. Please note that attributes of the type `Reference` or `ReferenceArray` cannot be displayed in the diagram view item. For an explanation of the attribute types `Reference` or `ReferenceArray`, see the section *Overview of Data Types Available for Custom Properties* in the chapter *Configuring the Class Model* in the reference manual *Configuring Alfabet with Alfabet Expand*.

   Indicator Types tab: This tab allows you to select up to four indicators to display on the diagram view item.

   - **Indicator Type 1 - 4**: Click the **Search** icon to select an indicator type to display on the diagram view item. You will only see the indicator types that have been associated with the object class that the diagram view item is associated with. For more information about assigning evaluation types and their indicator types to an object class, see the section Assigning Evaluations Types to an Object Class.

   Icon Definition tab: This tab allows you to select a custom attribute of the type `Enum` to display instead of the standard object class icon in the upper left corner of the diagram view item. You can

only select a custom attribute of the type `Enum` that has been defined for the object class that the diagram view item is associated with. Once the custom attribute is selected, you can specify an icon gallery to visualize the attribute values. For more information, see the section *Overview of Data Types Available for Custom Properties* in the chapter *Configuring the Class Model* in the reference manual *Configuring Alfabet with Alfabet Expand*.

- **Icon Property**: Select a custom attribute to display in the upper left corner of the diagram view item. The attribute will be displayed instead of the standard object class icon.
- **Icon Gallery**: Select the icon gallery that should visualize the values for the custom attribute selected in the **Icon Property** field.

  Once the icon gallery is selected, the table below the **Icon Gallery** field will display an **Icon** column, which displays the icons in the selected icon gallery, and a **Value** column. Click in the cell in the **Value** column to select the value that the icon should visualize.

  Custom icons are added to an icon gallery in the configuration tool Alfabet Expand. For more information, see the section *Adding and Maintaining Icons for the Alfabet Interface* in the reference manual *Configuring Alfabet with Alfabet Expand*.

5) Click **OK** to save the diagram view item or **Cancel** if you do not want to save it. You will see the diagram view in the table.

Once you have created diagram view items and assigned them to the relevant object classes, you can assign the diagram view items to a diagram view. For more information, see Assigning Diagram View Items to the Diagram View.

Deleting a Diagram View Item from a Selected Class

If you delete an object in the object class Diagram View Item, it will be irrevocably deleted from the Alfabet database.

To delete a diagram view item from a selected class:

1) In the **Diagram View Items** page view, click the diagram view item that you want to delete.
2) In the toolbar, click the **Delete**  button.
3) Confirm the warning by clicking **Yes,** or click **No** to exit without deleting the selected object(s).

Creating the Diagram View

A diagram view is a configuration that is associated with a diagram. It allows users to superimpose qualitative information - such as aggregated indicators or attribute values - associated with these architectural elements. Diagram views can be implemented in diagrams displaying applications, business processes, devices, domains, frameworks, platforms, and solution building blocks. Multiple diagram views may be defined and made available for a diagram. A diagram view may be reused for multiple diagrams.

To create a diagram view:

1) Click the **Diagram Views** explorer icon and click **Diagram Views**.
2) In the toolbar, click **New** > **Create New Diagram View**. The **Diagram View** editor opens.
3) Enter information into each field, as required.

General tab:

- **Name**: Enter a unique name for the diagram view.
- **Type**: Select the diagram type that the diagram view applies to. The diagram types for standard diagrams are predefined by Software AG. Refer to the table in the section Understanding the Object Classes Relevant for the Diagram View Configuration to understand the diagram type you should select to make a diagram view available for a particular view.

  The values that can be selected in the **Type** field that are relevant for custom diagrams will be specified by your solution designer. The protected enumeration `CustomDiagramViewType` is available for the definition of diagram views displayed in the context of custom diagrams. The enumeration items defined for the `CustomDiagramViewType` enumeration can be selected in the **Type** field in the **Diagram View** editor.

  The query specified for the `DiagramView` filter field for the `DiagramViewReport` report must refer to the relevant standard or custom diagram view type. The diagram views based on the standard or custom diagram view type can selected in configured reports based on the `DiagramViewReport` report template. For more information about configuring protected enumerations, see the section *Overview of Protected Enumerations* in the reference manual *Configuring Alfabet with Alfabet Expand - Appendix*. For more information about configuring custom diagrams, see the section *Configuring Custom Diagrams* in the reference manual *Configuring Alfabet with Alfabet Expand*.

- **Show Legend Items**: Select the checkbox to display a legend explaining which attributes and indicators are displayed in the diagram view items bundled in the diagram view.
- **Show Indicator Legend**: Select the checkbox to display a legend explaining the values represented by the indicator icons displayed in the diagram view items.
- **Description**: Enter a meaningful description that will clarify the purpose of the diagram view.

User Profiles tab:

- **User Profiles**: Select none, one, or multiple user profiles that have access permission to this diagram view. If no user profile is selected, all users with access permission to the diagram will be able to select the diagram view.

Color Rules tab:

- **Color Rules to Apply**: Select one or more color groups or their color rules to apply to the diagram view items. All diagram view items associated with objects found by the color rule will be highlighted with the specified color. A legend will be added to the diagram explaining the meaning of the colors.

  Color rules are created in the **Color Rules** view, which you can access when you click the root node of the **Diagram Views** explorer. Once a color rule is defined, it must be activated in order for the Alfabet query to be executed. The query should be re-activated periodically in order to update the query results and include changes made to the database. The color rule may be manually activated or activated via a batch process. For more information about creating a color rule, see the chapter Configuring Color Rules for Map Views and Diagram Views in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

4) Click **OK** to save the diagram view or **Cancel** if you do not want to save it.

Once you have created the diagram view, you can assign existing diagram view items to it. For more information, see Assigning Diagram View Items to the Diagram View.

Deleting a Diagram View

If you delete an object in the object class Diagram View, it will be irrevocably deleted from the Alfabet database. If any of the following dependent objects or data is defined for the deleted object, these will also be deleted: Diagram View Item.

1) In the **Diagram Views** page view, click the diagram view that you want to delete.
2) In the toolbar, click the **Delete**  button.
3) Confirm the warning by clicking **Yes** or click **No** to exit without deleting the selected object(s).

Assigning Color Rules to the Diagram View

You can assign relevant color rules to the diagram view in the **Diagram Views** page view.

A color rule is based on one or more Alfabet queries or native SQL queries that are configured to color a found set of objects in standard Alfabet business support maps and diagrams. If the color rule functionality is activated for a business support map, all activated color rules will be executed, and the matrix cells colored accordingly. In order to visualize color rules in diagrams, color rules must be assigned to a diagram view. When the diagram view is selected in a diagram, all activated color rules will be executed, and the diagram objects will be colored accordingly. The color rules are automatically added to legends where they are applicable.

To do so, select the **Edit** button in the toolbar to open the **Diagram View Editor**. Click the **Color Rules** tab and select one or more color rules to apply to the diagram view items. All diagram view items associated with objects found by the color rule will be highlighted with the specified color. Select **OK** to close the editor and save your changes.

Color rules are created in the **Color Rules** view, which you can access when you click the root node of the **Diagram Views** explorer. Once a color rule is defined, it must be activated in order for the Alfabet query to be executed. The query should be re-activated periodically in order to update the query results and include changes made to the database. The color rule may be manually activated or activated via a batch process. For more information about creating a color rule, see the section Configuring Color Rules for Map Views and Diagram Views in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

Assigning Diagram View Items to the Diagram View

Once the diagram view has been created, you can assign the diagram view items to the diagram view.

Diagram view items are created and assigned to an object class in the **Diagram View Items** page view in the **Class Configuration** functionality.

To add a diagram view item to the selected diagram view:

1) In the **Diagram Views** explorer, click the diagram view that you want to add a diagram view item to and click **Diagram View Items**.
2) In the toolbar, click **New** > **Add Diagram View Item**. The object selector opens and displays all diagram view items that have been configured. The object class that the diagram view item has been defined for is displayed in the **ClassName** column.
3) Select a diagram view item and click **OK**. The diagram view item is assigned to the selected diagram view.

Chapter 15: Configuring Enterprise Calendars for the User Community

The **Enterprise Calendars** functionality allows you to define multiple calendars per year for the enterprise, specifying which days are blocked out as holidays or weekends in each calendar. This allows you to create multiple enterprise calendars for all relevant regions of your enterprise. The enterprise calendars you create here will be available for users as a template when creating their own personal calendars in the **My Calendars** functionality.

Users may only create one personal calendar per year in the **My Calendars** functionality. They can then record the hours worked on tasks that have been assigned to them via skill requests that have been defined for such objects as project, applications, operational business supports, tactical business supports, strategic business supports, components, standard platforms, devices, deployments, and service products.

Regardless of the definition of blockout days, users will be able to record hours that they have worked on blockout days, if necessary.

Enterprise calendars can be created manually, as described below, or imported from a web service. In order to import calendars to Alfabet via the **New** > **Import Calendar** functionality, an ADIF import scheme must first be configured. For more information about the configuration of an ADIF import scheme, see the reference manual *Alfabet Data Integration Framework*.

To display the enterprise calendars created for a specific year, select the year in the **Year** field. To display all enterprise calendars regardless of the year that they were created for, select the empty row in the **Year** field.

To create a new enterprise calendar manually:

1) In the toolbar, click **New** > **Create New Calendar**.

If an existing enterprise calendar already exists, you can use it as a template to create a new enterprise calendar. All existing blocked out days and will be copied to the new enterprise calendar. To do so, click **New** > **Create New Calendar as Copy of Selected**.

2) The **Enterprise Calendar Editor** opens. Define the following fields, as needed:

Basic Data tab:

- **ID**: Displays the unique identification number of the enterprise calendar.
- **Short Name**: Define an abbreviation for the enterprise calendar.
- **Year**: Enter the year that the enterprise calendar is for.
- **Name**: Define the name of the enterprise calendar.
- **Description**: Define a description explaining the purpose of the enterprise calendar.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected object. The authorized user will have Read/Write access permissions to the object and is authorized to maintain the object in Alfabet.
- **Authorized User Groups**: Select the checkbox to assign Read/Write access permissions to all users in the selected user group. For more detailed information about the concept of access

permissions, see the section *Understanding Access Permissions in Alfabet* in the reference manual *Getting Started with Alfabet*.

3) Click **OK** to save your changes. The new enterprise calendar is displayed in the table.

4) Select the enterprise calendar and click **Edit Blockout Days**. The **Calendar Content** view opens displaying the enterprise calendar for the relevant year. The days that are highlighted in the calendar represent days that have been blocked out. Select a day and click the **Edit** button to view more information about the blocked out day.

5) You can define blockout days using any of the methods described below:

- To define a specific day as a blockout day: Select a day in the enterprise calendar and click **New** > **Create New Blockout Day**. In the editor, revise the **Start Date** and **End Date** fields, if needed. Enter a name in the **Name** field and a description in the **Description** field and click **OK**. The blockout day is highlighted in the enterprise calendar.

- To copy all blockout days defined from an existing enterprise calendar to the selected enterprise calendar: Click **New** > **Copy Blockout Days from Another Calendar**. The object selector displays the selected enterprise calendar as well as all defined enterprise calendars for the same year that you are defining. Select the enterprise calendar that you want to copy to the selected calendar and click **OK**. The blockout days are highlighted in the enterprise calendar.

- To blockout specific days of every week in the enterprise calendar: Click **New** > **Block Out Specific Weekdays**. In the editor, click each day that should be highlighted as a blockout day in the calendar. Define the **Start Date** and **End Date** fields and enter a name for the blocked out days in the **Blockout Calendar Item Name** field (for example: Weekend). The name you define will be displayed in a tooltip when the user points to the day in the calendar. Click **OK** to save your changes.

- To edit the blockout day definition: Select the day in the calendar and click the **Edit**  button. You can edit the **Start Date** and **End Date** fields, change the name of the blockout calendar item in the **Name** field, or add comments about the blockout day in the **Description** field. Click **OK** to save your changes.

- To remove the blockout day definition from a day: Select the day in the calendar and click the **Delete**  button. Confirm the message by clicking **Yes**. The blockout day is no longer highlighted.

- To remove all blockout day definitions from the calendar: Click **New** > **Clear All Blockout Days**. Confirm the message by clicking **Yes**. The blockout days are no longer highlighted.

Chapter 16: Managing Business Documents

The **Manage Business Documents** functionality allows the enterprise to capture business documents as enterprise assets. A business document is a non-structured document that is often not directly related to IT. They typically represent reports or operating instructions required for performing the business activities associated with a business function. The business document may be specified as an input needed for a business function or as a deliverable generated as output by a business function. A name and alias can be defined for the business document as well as attachments, dynamic web links, assignments, annotations, evaluations, and responsibilities.

The **Manage Business Documents** functionality allows the business documents to be created and managed. Users can then assign the business documents as inputs needed for a business function in the *Business Function Inputs Page View* or as outputs in the *Business Function Deliverables Page View*. Both views are available in the standard object profile for a business function.

To create a business document, click **New** > **Create New Business Document**. In the editor that opens, define the following fields and click **OK** to save your changes:

Basic Data tab:

- **Name**: Enter a unique name for the business document.
- **Description**: Enter a meaningful description that will clarify the purpose of the business document.
- **Alias**: Enter an alias for the business document.

Authorized Access tab:

- **Authorized User**:

  Click the **Search** icon to assign an authorized user to the selected object. The authorized user will have Read/Write access permissions to the object and is authorized to maintain the object in Alfabet.

- **Authorized User Groups**: Select the checkbox to assign Read/Write access permissions to all users in the selected user group. For more detailed information about the concept of access permissions, see the section *Understanding Access Permissions in Alfabet* in the reference manual *Getting Started with Alfabet*.

Chapter 17: Configuring Business Roles

Business roles allow information to be captured about funtional relationships that are not fulfilled by people or organizations. For example, actors (such as IT Security Manager, Private Network User, Services Owner), personas (such as Stakeholder, Process Owner, Approver), and job descriptions are typical business roles. Business roles are for informational purposes only and do not impact access permissions.

Please note the following:

- Business roles may be based on object class stereotypes that have been configured by your enterprise. For more information about the configuration of object class stereotypes, see the section *Configuring Object Class Stereotypes for Object Classes* in the reference manual Alfabet Expand.
- Business roles may be assigned to applications, brands, custom segments, demands, domains, features, ICT objects, business processes, business process versions, market products, organizations, sales channels and service products in the *Associated Business Roles Page View* available in the object profile of the relevant object.
- Evaluation types may be specified for business roles via the class **Business Role Relationship** in the **Class Configuration** functionality. For more information about configuring evaluation types, see the section Configuring Evaluation Types.

To create a business role:

1) In the toolbar, click **New** > **Create New Business Role**.
2) If object class stereotypes have been configured, the **Stereotype Selector** will open. Select the relevant object class stereotype that the business role shall be based on and click **OK**.
3) The **Business Role** editor opens. Define name and description of the business role.
4) Click **OK** to save the business role or click **Cancel** to close the editor without saving.

Chapter 18: Configuring Business Questions

Business questions allow the focus areas of the business' use cases to be articulated in order to allow the enterprise to capture and evaluate information that supports a particular path of inquiry that is relevant for the business. Business questions are made available for users assigned to a specified user profile. Business questions may be implemented in object cockpits and guide views.

Multiple business questions can be created. All business questions will be available to all user profiles unless a user profile opts out of a business question. In this case, users assigned to the user profile will not have access permissions to the business question.

The following information is available:

Creating Business Questions

You can create multiple business questions in the **Business Questions** page view. A business question is available per default to all user profiles and user groups in the user community.

1) In the toolbar, click **New** > **Create Business Question**. The **Business Question Editor** opens.
2) Define the following fields, as needed:
   - **Name**: Enter a unique name for the business question.
   - **Short Name**: Specify a short name for the business question.
   - **Technical Name**: Specify a technical name of the business question. This is necessary in order to implement the business question in object cockpits and guide views.
   - **Release Status**: Select the current release status for the business question.

     The set of release statuses available for an object class are configured by your solution designer in the configuration tool Alfabet Expand. For more information, see the section *Configuring Release Status Definitions for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*. For general information about release statuses, see the section *Understanding Release Statuses* in the reference manual *Getting Started with Alfabet*.

   - **Description**: Enter a meaningful description that will explain the purpose of the business question.
   - **Keywords**: Enter meaningful keywords for the business description.
3) Click **OK** to save the business question or click **Cancel** to exit without saving.

Specifying the Opt-Out Definition for Business Questions

A business question is available per default to all user profiles and user groups in the user community. In the **Business Questions Opt Outs** page view. You can specify one or more user profiles to opt out of a business question. In this case, the opted-out business question will not be available to the specified user group. The table displays all business questions that have been defined. The following information is available:

- **Business Question ID**: The unique identification number of the business question.
- **Business Question Name**: The unique name of the business question.
- **Business Question Short Name**: The short name of the business question.
- **Opted Out**: Displays a checkmark if one or more user profiles are opted out of the business question.

To specify that one or more user profiles shall opt out of a business question:

1) In the dataset, select the business question that shall not be available to one or more user profiles.
2) In the toolbar, click the **Opt Out** button and select **Opt Out for User Profile**.

   If all user profiles shall be opted out for the selected business question, select the **Opt Out for All User Profiles**. A checkmark will be displayed in the **Opted Out** column.

3) In the selector that opens, select one or more user profiles that shall not have access to the selected business question and click **OK**. If one or more user profiles have been selected for a business question, a checkmark will be displayed in the **Opted Out** column.

To specify that the opt-out definition shall be revoked for a business question, select the business question in the dataset and click the **Revoke Opt-Out for All User Profiles**. All opt-out definitions will be removed from the business question. As a result, the checkmark in the **Opted Out** column will be removed for the business question and the business question will be available to all user profiles per default.

To view all user profiles that may have access to a business question as well as those for which the opt-out definition has been defined, click the business question in the dataset and click the **Navigate** button. Open theBusiness Question Opt Outs for User Profiles Page Viewto further specify the user profiles that do and do not have access to the selected business question.

Reviewing and Modifying the Opt-Out Definitions for a Selected Business Question

The **Business Question Opt Outs for User Profiles** page view displays all user profiles that may have access to a business question as well as those for which the opt-out definition has been defined. A checkmark will be displayed in the **User Profile Opted Out** column for all user profiles that do not have access to the business question. You can remove the opt-out definition for any user profile as well as assign the opt-out definition to a user profile.

To remove the opt-out definition from a user profile, select the user profile in the dataset and click the **Revoke Opt Out for User Profile** button in the toolbar. The checkmark is removed from the **User Profile Opted Out** column.

To assign the op-out definition to a user profile, select the user profile in the dataset and click the **Opt Out for User Profiles** button int he toolbar. A checkmark will be set in the **User Profile Opted Out** column.

Chapter 19: Configuring Semantic Connections for Integration Solutions

The **Integration Solutions Configuration** functionality allows you to create Alfabet database connection objects representing all relevant database connections configured for various integration solutions.

The following information is available:

Specifying Alfabet Database Connections to Import Configuration-Relevant Data from a Master Database

The definition of Alfabet database connections is part of the required configuration for the following functionalities:

- The **Import Data Search** functionality that enables import from configuration relevant data S(such as cost types or role types) from a database in a solution configuration environment to a database in a production environment.

  For a detailed description of the functionality and the required configuration steps to activate it, see the section *Importing Objects of Configuration Relevant Object Classes*

from a Master Database in the reference manual *Configuring Alfabet with Alfabet Expand*.

- Events that trigger automatic start of a workflow or execution of an ADIF scheme via a RESTful service call to the Alfabet Web Application that is automatically send in the background as a consequence of a defined user interaction. for example, if a user leaves a wizard step or enters a workflow step. The sending of the RESTful service call that starts the workflow or executes the ADIF import or export can be defined in two ways:
  - The RESTful service call can target the same Alfabet Web Application that processes the event using a defined user in the Alfabet database for authentication.
  - The RESTful service call can either target the same or another Alfabet Web Application, connecting to a different Alfabet database using a connection definition read from the XML object ***AlfabetIntegrationConfig*** configured in Alfabet Expand.

  If the RESTful service call for an event is configured in the XML object ***AlfabetIntegrationConfig***, an Alfabet database connection pointing to the connection has to be defined and assigned to the event via an event template.

  For a detailed description of functionality and the required configuration steps to activate it, see the chapter *Configuring Events* in the reference manual *Configuring Alfabet with Alfabet Expand*.

Do the following to create an Alfabet database connection:

1) Go to the **Integration Solutions Configuration** functionality and click the **Alfabet Database Connection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create Alfabet Database Connection**.
3) In the **Alfabet Database Connection** editor, define the following fields as needed:

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each ARIS database connection. This number cannot be edited.
   - **Name**: Enter a unique name for the Alfabet database connection. The name is displayed in the **Import Data Search** functionality in the drop-down list for selection of the master database for import of Alfabet configuration data.
   - **Release Status**: Select the Alfabet database connection's current release status.

     The set of release statuses available for an object class are configured by your solution designer in the configuration tool Alfabet Expand. For more information, see the section *Configuring Release Status Definitions for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*. For general information about release statuses, see the section *Understanding Release Statuses* in the reference manual *Getting Started with Alfabet*.

   - **Description**: Enter a meaningful description that will clarify the purpose of the Alfabet database connection.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected ARIS database connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
- **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

Connection tab:

- **Alfabet Connection**: Select the connection to the relevant Alfabet database that is configured in the XML element ***Connection*** in the XML object ***AlfabetIntegrationConfig*** in Alfabet Expand.
- **Allowed Classes**: Select one or multiple object classes for that users authorized to import data via this Alfabet database connection shall be able to import data via the **Import Data Search** functionality. All classes that may be imported according to the definition in the XML element ***Connection*** in the XML object ***AlfabetIntegrationConfig*** in Alfabet Expand are listed in the drop-down list of the field.

  After you have selected object classes for import, you will see your selection in the field **Selected Classes**. The user will see the selected classes in the drop-down list of the object class filter in the **Import Data Search** functionality after having selected the Alfabet database connection.

4) Click **Test Alfabet Database Connection**. If your settings are correct, a message "The connection is valid" is displayed. Otherwise, an error message is displayed.

Defining Semantic Connections to CentraSite

CentraSite connection definitions can only be specified in the **Integration Solutions Configuration** functionality if the prerequisite configuration has been completed. For a detailed description of interoperability with CentraSite® including all required configuration steps in Alfabet, the section *Configuring Interoperability with CentraSite* in the reference manual *API Integration with Third-Party Components*.

The **CentraSite Connections** view allows you to create semantic definitions for all relevant CentraSite connections that have been configured in the XML object ***CentraSiteManager***. The CentraSite connection definitions are necessary so that users can specify which configured connection to use to synchronize Alfabet technical services with CentraSite assets. A CentraSite connection definition should be created for each connection configured in the XML object ***CentraSiteManager***.

Each time the **Technical Services from Repositories** page views (`COMSR_ServicesExt` or `COMSR_Services`) is loaded, each configured connection in the XML object ***CentraSiteManager*** will be established and the technical services based on CentraSite assets will be synchronized with the data in the CentraSite repository. If the user wants to synchronize a selected technical service in the **Technical Services** page view (`COM_TechServices`) with the corresponding CentraSite asset, the relevant connection definition to use for the synchronization must be selected in the **Master Repository Connection** field in the **Operational Repositories** tab of the **Technical Service** editor. If Alfabet should automatically synchronize the technical services based on CentraSite assets when the **Technical Services** page view (`COM_TechServices`) is loaded, you must select the **Automatically Update Services** checkbox in the **CentraSite**

Connection editor. If the checkbox is not selected, the user must explicitly update a selected technical service.

To configure a CentraSite connection to display in the **Connection** field in the **Service Registry** tab of the **Technical Service** editor:

1) Go to the **Integration Solutions Configuration** functionality and click the **CentraSite Connection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create CentraSite Connection**.
3) In the **CentraSite Connection** editor, define the following fields as needed.

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each CentraSite connection. This number cannot be edited.
   - **Name**: Enter a unique name for the CentraSite connection. The name should help the user synchronizing the technical service to identify the CentraSite repository that will be targeted by the connection.
   - **Release Status**: Select the CentraSite connection's current release status.
   - **Description**: Enter a meaningful description that will clarify the purpose of the CentraSite connection.
   - **Background Color**: Specify the background color to be used for technical services that are associated with this CentraSite connection. In the **Technical Services from Repositories** views, the colored rows represent assets which technical services have been created in Alfabet. In the **Technical Services** view, the colored rows represent technical services that exist in Alfabet and have been exported to the operational repository.
   - **Foreground Color**: Specify the foreground color to be used in conjunction with the background color.

   Authorized Access tab:

   - **Authorized User**: Click the **Search** icon to assign an authorized user to the selected CentraSite connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
   - **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

   Connection tab:

   - **CentraSite Connection**: Select the relevant connection configured in the XML element ***Connection*** in the XML object ***CentraSiteManager*** that will be established when the user selects this CentraSite connection in the **Master Repository Connection** field in the **Operational Repositories** tab of the **Technical Service** editor.
   - **Automatically Update Technical Services**: If selected, the technical services that are linked to services in the operational service repository will be automatically updated in the operational repository when the **Technical Services** page view is loaded. If the **Automatically Update Technical Services** checkbox is not selected, the technical services must be explicitly manually updated.

4) Repeat this for all configured ***Connection*** elements in the XML object ***CentraSiteManager*** that users should be able to select in the **Connection** field in the **Operational Repositories** tab of the **Technical Service** editor.

Defining Semantic Connections to webMethods API Portal

API portal connection definitions can only be specified in the **Integration Solutions Configuration** functionality if the prerequisite configuration has been completed. For a detailed description of interoperability with webMethods® API Portal including all required configuration steps in Alfabet, the section *Configuring Interoperability with webMethods API Portal* in the reference manual *API Integration with Third-Party Components*.

The **API Portal Data Connection** view allows you to create semantic definitions for all relevant API portal connections that have been configured in the XML object ***APIPortalConfig***. The API portal data connection definitions are necessary so that users can specify which configured connection to use to synchronize Alfabet technical services with API portal assets. An API portal data connection definition should be created for each connection configured in the XML object ***APIPortalConfig***.

Each time the **Technical Services from Repositories** page views (`COMSR_ServicesExt` or `COMSR_Services`) is loaded, each configured connection in the XML object ***APIPortalConfig*** will be established and the technical services based on API portal assets will be synchronized with the data in the API portal repository. If the user wants to synchronize a selected technical service in the **Technical Services** page view (`COM_TechServices`) with the corresponding API portal asset, the relevant connection definition to use for the synchronization must be selected in the **Master Repository Connection** field in the **Operational Repositories** tab of the **Technical Service** editor. If Alfabet should automatically synchronize the technical services based on APIs from the API portal when the **Technical Services** page view (`COM_TechServices`) is loaded, you must select the **Automatically Update Services** checkbox in the **API Portal Data Connection** editor. If the checkbox is not selected, the user must explicitly update a selected technical service.

Documents from Alfabet may be shared with the API portal specified by the connection if they meet a document type as well as document category specification. Any document shared with API portal must comply to the whitelist and blacklist definition in the XML object ***FileExtensions***.

To configure an API portal data connection to display in the **Connection** field in the **Service Registry** tab of the **Technical Service** editor:

1) Go to the **Integration Solutions Configuration** functionality and click the **API Portal Data Connection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create API Portal Data Connection**.
3) In the **API Portal Data Connection** editor, define the following fields as needed.

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each API portal connection. This number cannot be edited.
   - **Name**: Enter a unique name for the API portal connection. The name should help the user synchronizing the technical service to identify the API portal repository that will be targeted by the connection.
   - **Release Status**: Select the API portal connection's current release status.

- **Description**: Enter a meaningful description that will clarify the purpose of the API portal connection.
- **Background Color**: Specify the background color to be used for technical services that are associated with this API portal connection. In the **Technical Services from Repositories** views, the colored rows represent assets which technical services have been created in Alfabet. In the **Technical Services** view, the colored rows represent technical services that exist in Alfabet and have been exported to the operational repository.
- **Foreground Color**: Specify the foreground color to be used in conjunction with the background color.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected API portal connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
- **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

Connection tab:

- **API Portal Connection**: Select the relevant connection configured in the XML element ***APIPortalConnection*** in the XML object ***APIPortalConfig*** that will be established when the user selects this API portal connection in the **Master Repository Connection** field in the **Operational Repositories** tab of the **Technical Service** editor.
- **Automatically Update Technical Services**: If selected, the technical services that are linked to services in the operational service repository will be automatically updated in the operational repository when the **Technical Services** page view is loaded. If the **Automatically Update Technical Services** checkbox is not selected, the technical services must be explicitly manually updated.
- **Shared Document Types**: Specify the file extensions for the document types that may be allowed as attachments defined for technical services. Documents matching the specification will be included in the export via the API portal connection. The file extensions defined in the **Shared Document Types** field may not be included in the blacklist specified in the XML object ***FileExtensions***. If a whitelist of file extensions is enabled, the file extensions must be included in the whitelist. The list of options is derived from the list of file types supported in webMethods API Portal. For more information about the specification of a whitelist and/or blacklist for file extensions in Alfabet, see the section *Configuring the Permissibility of Files and Web Links in Alfabet* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- **Shared Document Categories**: Specify the document categories that may be allowed as attachments to be included in the export via the API portal connection. The file extensions of the documents in the selected document categories may not be included in the blacklist specified in the XML object ***FileExtensions***. If a whitelist of file extensions is enabled, the file extensions must be included in the whitelist. The list of options is derived from the enumeration items defined for the enumeration ***AlfaDocCategory***.

4) Repeat this for all configured ***Connection*** elements in the XML object ***APIPortalConfig*** that users should be able to select in the **Connection** field in the **Operational Repositories** tab of the **Technical Service** editor.

Defining Semantic Connections to webMethods API Gateway

API gateway connection definitions can only be specified in the **Integration Solutions Configuration** functionality if the prerequisite configuration has been completed. For a detailed description of interoperability with webMethods® API Gateway including all required configuration steps in Alfabet, the section *Configuring Interoperability with webMethods API Gateway* in the reference manual *API Integration with Third-Party Components*.

The **API Gateway Data Connections** view allows you to create semantic definitions for all relevant API gateway connections that have been configured in the XML object ***APIGatewayConfig***. The API gateway data connection definitions are necessary so that users can specify which configured connection to use to synchronize Alfabet technical services with API gateway assets. An API gateway data connection definition should be created for each connection configured in the XML object ***APIGatewayConfig***.

Each time the **Technical Services from Repositories** page views (`COMSR_ServicesExt` or `COMSR_Services`) is loaded, each configured connection in the XML object ***APIGatewayConfig*** will be established and the technical services based on API gateway assets will be synchronized with the data in the API gateway repository. If the user wants to synchronize a selected technical service in the **Technical Services** page view (`COM_TechServices`) with the corresponding API gateway asset, the relevant connection definition to use for the synchronization must be selected in the **Master Repository Connection** field in the **Operational Repositories** tab of the **Technical Service** editor. If Alfabet should automatically synchronize the technical services based on APIs from the API gateway when the **Technical Services** page view (`COM_TechServices`) is loaded, you must select the **Automatically Update Services** checkbox in the **API Gateway Data Connection** editor. If the checkbox is not selected, the user must explicitly update a selected technical service.

To configure an API gateway data connection to display in the **Connection** field in the **Service Registry** tab of the **Technical Service** editor:

1) Go to the **Integration Solutions Configuration** functionality and click the **API Gateway Data Connection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create API Gateway Data Connection**.
3) In the **API Gateway Data Connection** editor, define the following fields as needed.

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each API gateway data connection. This number cannot be edited.
   - **Name**: Enter a unique name for the API gateway data connection. The name should help the user synchronizing the technical service to identify the API gateway repository that will be targeted by the connection.
   - **Release Status**: Select the API gateway data connection's current release status.
   - **Description**: Enter a meaningful description that will clarify the purpose of the API gateway data connection.
   - **Background Color**: Specify the background color to be used for technical services that are associated with this API gateway data connection. In the **Technical Services from Repositories** views, the colored rows represent assets which technical services have been created in Alfabet. In the **Technical Services** view, the colored rows represent technical services that exist in Alfabet and have been exported to the operational repository.

- **Foreground Color**: Specify the foreground color to be used in conjunction with the background color.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected API gateway data connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
- **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

Connection tab:

- **API Gateway Connection**: Select the relevant connection configured in the XML element ***APIGatewayConnection*** in the XML object ***APIGatewayConfig*** that will be established when the user selects this API gateway data connection in the **Master Repository Connection** field in the **Operational Repositories** tab of the **Technical Service** editor.
- **Automatically Update Technical Services**: If selected, the technical services that are linked to services in the operational service repository will be automatically updated in the operational repository when the **Technical Services** page view is loaded. If the **Automatically Update Technical Services** checkbox is not selected, the technical services must be explicitly manually updated.
- **Shared Document Types**: Specify the file extensions for the document types that may be allowed as attachments defined for technical services. Documents matching the specification will be included in the export via the API gateway data connection. The file extensions defined in the **Shared Document Types** field may not be included in the blacklist specified in the XML object ***FileExtensions***. If a whitelist of file extensions is enabled, the file extensions must be included in the whitelist. The list of options is derived from the list of file types supported in webMethods API Gateway. For more information about the specification of a whitelist and/or blacklist for file extensions in Alfabet, see the section *Configuring the Permissibility of Files and Web Links in Alfabet* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- **Shared Document Categories**: Specify the document categories that may be allowed as attachments to be included in the export via the API gateway data connection. The file extensions of the documents in the selected document categories may not be included in the blacklist specified in the XML object ***FileExtensions***. If a whitelist of file extensions is enabled, the file extensions must be included in the whitelist. The list of options is derived from the enumeration items defined for the enumeration ***AlfaDocCategory***.

4) Repeat this for all configured ***APIGatewayConnection*** elements in the XML object ***APIGatewayConfig*** that users should be able to select in the **Connection** field in the **Operational Repositories** tab of the **Technical Service** editor.

Defining Semantic Connections to Apigee

Apigee data connection definitions can only be specified in the **Integration Solutions Configuration** functionality if the prerequisite configuration has been completed. For a detailed description of interoperability with webMethods® API Portal including all required configuration steps in Alfabet, the section *Configuring Interoperability with Google's Apigee API Platform Tools* in the reference manual *API Integration with Third-Party Components*.

The **Apigee Data Connections** view allows you to create semantic definitions for all relevant Apigee connections that have been configured in the XML object ***APIRepositoryConfig***. The Apigee data connection definitions are necessary to define which environments in an organization managed in Apigee the connection will target. Users can specify which configured connection to use to synchronize Alfabet technical services with Apigee assets. At least one Apigee data connection definition should be created for each connection configured in the XML object ***APIRepositoryConfig***.

Each time the **Technical Services from Repositories** page views (`COMSR_ServicesExt` or `COMSR_Services`) is loaded, each configured connection in the XML object ***APIPortalConfig*** will be established and the technical services based on Apigee assets will be synchronized with the data in the Apigee repository. If the user wants to synchronize a selected technical service in the **Technical Services** page view (`COM_TechServices`) with the corresponding Apigee asset, the relevant connection definition to use for the synchronization must be selected in the **Master Repository Connection** field in the **Operational Repositories** tab of the **Technical Service** editor. If Alfabet should automatically synchronize the technical services based on APIs from the Apigee repository when the **Technical Services** page view (`COM_TechServices`) is loaded, you must select the **Automatically Update Technical Services** checkbox in the **API Portal Database Connection** editor. If the checkbox is not selected, the user must explicitly update a selected technical service.

To configure an Apigee data connection to display in the **Connection** field in the **Service Registry** tab of the **Technical Service** editor:

1) Go to the **Integration Solutions Configuration** functionality and click the **Apigee Data Connection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create Apigee Data Connection**.

   If you have already defined a similar connection and want to take over the settings of that connection for your new connection, you can alternatively click **New** > **Create Apigee Data Connection as Copy** and select the existing connection the new connection should base on from the selector that opens. The editor for the new connection will then open with all settings identical to the copied connection and the name set to "Copy of <base connection name>".

3) In the **Apigee Data Connection** editor, define the following fields as needed.

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each Apigee data connection. This number cannot be edited.
   - **Name**: Enter a unique name for the Apigee data connection. The name should help the user synchronizing the technical service to identify the Apigee repository that will be targeted by the connection.
   - **Release Status**: Select the Apigee data connection's current release status.
   - **Description**: Enter a meaningful description that will clarify the purpose of the Apigee data connection.
   - **Background Color**: Specify the background color to be used for technical services that are associated with this Apigee data connection. In the **Technical Services from Repositories** views, the colored rows represent assets for which technical services have been created in Alfabet. In the **Technical Services** view, the colored rows represent technical services that exist in Alfabet and have been exported to the operational repository.

- **Foreground Color**: Specify the foreground color to be used in conjunction with the background color.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected Apigee connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
- **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

Connection tab:

- **Apigee Connection**: Select the relevant connection configured in the XML element ***RepositoryConnection*** in the XML object ***APIRepositoryConfig*** that will be established when the user selects this Apigee data connection in the **Master Repository Connection** field in the **Operational Repositories** tab of the **Technical Service** editor.
- **Organization**: Enter the name of the organization in Apigee to that the connection shall be established.
- **Deployed Environments**: After having selected the organization in the **Organization** field, click the **Get Environments** button to view all implemented Apigee environments in the Apigee organization. Select the relevant environment containing the API Proxies to be integrated with technical services in Alfabet via the Apigee data connection.
- **Consider Revision**: Select the checkbox if you would like to import any other than the latest revision of API Proxies. If the checkbox is selected, a user can choose between all revisions in the selected environments for integration. Export to Apigee will create an API Proxy with revision 1. If the checkbox is not selected, the latest revision of an API Proxy in the selected environments will be used for integration.
- **Proxy Name Template**: Technical services created in Alfabet on basis of API Proxies in Apigee will be named according to the naming convention defined with this field. The definition can contain the following variables:
  - `{Name}` will be substituted with the API Proxy name.
  - `{Version}` will be substituted with the API Proxy version.

  By default, the naming convention without considering revisions is: `{Name}_V{Version}`. If revisions are considered, the specified name will be amended with "`r.RevisionNumber `".

4) Repeat this for all configured ***Connection*** elements in the XML object ***APIPortalConfig*** that users should be able to select in the **Connection** field in the **Operational Repositories** tab of the **Technical Service** editor.

Defining Semantic Connections for ARIS - Alfabet Interoperability Interface

The definition of ARIS database connections is part of the required configuration to use the ARIS - Alfabet Interoperability Interface that enables data integration between ARIS and Alfabet. Precondition for the creation of ARIS database connections is a valid specification of a connection to ARIS Connect in the XML

object ***ArisApiConfig*** in Alfabet Expand. For each database that shall be accessed via the defined ARIS Connect, an ARIS database connection with the access data specific to the database must be created.

For a detailed description of the ARIS - Alfabet Interoperability Interface including all required configuration steps in Alfabet and handling, see the reference manual *ARIS - Alfabet Interoperability*.

To configure an ARIS database connection:

1) Go to the **Integration Solutions Configuration** functionality and click the **ARIS Database Connection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create ARIS Database Connection**.

If you have already defined a similar connection and want to take over the settings of that connection for your new connection, you can alternatively click **New** > **Create ARIS Database Connection as Copy** and select the existing connection the new connection should base on from the selector that opens. The editor for the new connection will then open with all settings identical to the copied connection and the name set to "Copy of <base connection name>".

3) In the **ARIS Database Connection** editor, define the following fields as needed.

Basic Data tab:

- **ID**: Alfabet assigns a unique identification number to each ARIS database connection. This number cannot be edited.
- **Name**: Enter a unique name for the ARIS database connection. The name should help the importing data from ARIS or setting an ARIS diagram link to identify the ARIS database that will be targeted by the connection.
- **Release Status**: Select the ARIS database connection's current release status.

The set of release statuses available for an object class are configured by your solution designer in the configuration tool Alfabet Expand. For more information, see the section *Configuring Release Status Definitions for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*. For general information about release statuses, see the section *Understanding Release Statuses* in the reference manual *Getting Started with Alfabet*.

- **Description**: Enter a meaningful description that will clarify the purpose of the ARIS database connection.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected ARIS database connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
- **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

Connection tab:

- **ARIS Connection**: Select the connection to the relevant ARIS Connect that is configured in the XML element ***ArisApiConnection*** in the XML object ***ArisApiConfig***.

- **ARIS Tenant**: Enter the tenant for authorization with the ARIS API (for example "`default`").
- **ARIS Database**: Enter the name of the ARIS database that includes the data relevant for data integration.
- **ARIS Method Filter**: Enter the name of the filter that shall be applied when accessing the ARIS database (for example "Entire Method").

  Please note the following about the influence of the ARIS method filter on the availability of data for data integration into the Alfabet database:

  - Method filters can hide data and thus the method filter used can change the outcome of the integration significantly. Choose the method filter carefully. If unsure, you can start with method filter "Entire method", although please note that even though it is called "Entire method" it does not retrieve all data, especially custom types might not be included. It might be necessary to create and use an own filter in some cases.
  - ARIS users can get method filters assigned in the context of a database. The same user can have different method filters for different databases!
  - An ARIS database has a defined "Default Filter". If the filter you define with the **ARIS Method Filter** attribute deviates from the default filter, and a user accessing the ARIS database from Alfabet does not have the method filter defined with ARIS Method Filter assigned, the default method filter is applied instead of the method filter defined with ARIS Method Filter. There is no warning that a different filter is used. Only the resulting data might not be as expected.

- **ARIS Versioning State**: Select one of the following:
  - **Workspace** to connect to the workspace of the ARIS database. The workspace is the editable version that contains changes currently under construction.
  - **Current Version** to connect to the current version published for the ARIS database. The current version is the non-editable version of the ARIS database currently published as official process model version.
  - **Change List** to connect to any of the prior versions of the ARIS database stored in the change list for the ARIS database.
- **ARIS Change List Number**: This attribute is only relevant if the **ARIS Versioning State** attribute has been set to **Change List**. Enter the change list number of the ARIS database version in the change list as integer.

4) Click **Test ARIS Database Connection**. If your settings are correct, a message "The database connection is valid" is displayed. Otherwise, an error message is displayed.

Please note that a connection to a non-existent ARIS database version in the change list will only return an error message via the test mechanism. If you do not test the connection and the ARIS database version does not exist, the connection will not return any data without throwing an error message.

Defining Semantic Connections to Jira

Alfabet supports data import from Jira® to Alfabet as well as the export of Alfabet data to Jira. The capability to import Jira data to Alfabet allows the enterprise to import issues, projects, categories, and boards documented in Jira to specified entities in Alfabet such as applications, deployments, standard platforms, projects, etc., thus providing a bridge between agile decisions on the enterprise level with operational decisions taken by SCRUM teams, project teams, etc.

The capability to export Alfabet data to Jira supports the planning and prioritization of operative development work and provides support for bimodal IT and agile projects in the context of portfolio evaluation. The goal of the Jira export is to create new issues in Jira based on issues that have been captured via demands, features, issues, value nodes, etc. in Alfabet.

Jira connection definitions can only be specified in the **Integration Solutions Configuration** functionality if the prerequisite configuration has been completed. This includes, for example, the configuration of the database connection, the ADIF import and export schemes that specify the data to be imported/exported, and the identifiers required to map the Jira project data to Alfabet object data. For a detailed description of interoperability with Jira® including all required configuration steps in Alfabet, the section *Configuring Integration with Jira* in the reference manual *API Integration with Third-Party Components*.

The **Jira Connection** view in the **Integration Solutions Configuration** functionality in the Alfabet user interface allows you to create one or more semantic definitions for all relevant Jira® connections that have been configured in the XML object ***JIRAIntegrationConfig*** in order to export Alfabet data to Jira. A Jira connection definition should be created for each data connection configured in the XML object ***JIRAIntegrationConfig***. You can configure multiple semantic data connections for each data connection configured in the XML element ***DataConnection*** in the XML object ***JIRAIntegrationConfig***.

A semantic Jira connection allows you to map the Alfabet object classes to the Jira project structure and to specify the stereotypes and possible a configured report that shall return the set of Alfabet objects that may be integrated with Jira. For each Jira connection you create, you must specify the integration pattern to map objects in the Alfabet object hierarchy to the Jira project, project version, or project component. You can choose an architecture-based integration pattern or a project-based integration pattern for the mapping.

Whether your enterprise chooses the architecture-based or project-based integration will depend on your methodology of using Jira. If your company associates demands with products in Alfabet, the ICT object integration pattern may be more relevant, but if demands are associated with projects, a project hierarchy integration pattern may be more suitable.

The following are examples of potential use cases for the integration with Jira. The first example is based on a project-based integration pattern and the second is based on an architecture-based integration pattern:

- The IT department of a banking enterprise documents bugs and change requests for their trading software in Jira and captures their operational plans in Alfabet. In this case, for example, the Jira issues could be imported to Alfabet as features that are planned for applications that are owned and fiscally managed via ICT objects. The following mapping schema provides an example of how Jira projects could be mapped to a project structure in Alfabet:

Alfabet Class	Jira Class
ICT Object	`Project`
Application	`ProjectVersion`
Local Component	`ProjectComponent`
Feature	`Issue`

- A banking enterprise captures demands in order to plan the modification of the bank's trading capabilities. The demands are assigned to projects in Alfabet. The Project stereotype "Program" in Alfabet could be mapped to `Projects` in the Jira instance. The demands will be exported as issues to Jira in order to implement the projects and operationalize the needed changes. The subordinate project stereotype Project could be mapped to the Jira `ProjectVersion` and the project stereotype Project Step would be mapped to the Jira `ProjectComponent`. The following mapping schema provides an example of how Jira projects could be mapped to an ICT object hierarchy in Alfabet:

Alfabet Class	Jira Class
Project Stereotype 1	`Project`
Project Stereotype 2	`ProjectVersion`
Project Stereotype 3	`ProjectComponent`
Demand	`Issue`

The following integration patterns may be specified for a project-based approach:

Integration Type	Jira Project	Jira Project Version	Jira Project Component
Jira Projects, Project Versions, and Project Components	**Project** Stereotype 1	**Project** Stereotype 2	**Project** Stereotype 3

Integration Type	Jira Project	Jira Project Version	Jira Project Component
Jira Projects and Project Versions	**Project** Stereotype 1	**Project** Stereotype 2	
Jira Project and Project Components	**Project** Stereotype 1		**Project** Stereotype 3
Jira Projects	**Project** Stereotype 1		
Jira Project Versions		**Project** Stereotype 2	
Jira Project Versions and Project Components		**Project** Stereotype 2	**Project** Stereotype 3
Jira Project Components			**Project** Stereotype 3

The following integration patterns may be specified for an architecture-based approach:

Integration Type	Jira Project	Jira Project Version	Jira Project Component
Jira Projects, Project Versions, and Project Components	ICT Object (or stereotype)	Application (or stereotype)	Local Component (or stereotype)
	ICT Object (or stereotype)	Component (or stereotype)	Local Component (or stereotype)
	ICT Object (or stereotype)	Standard Platform (or stereotype)	Platform Element (or stereotype)
Jira Projects and Project Versions	ICT Object (or stereotype)	Application (or stereotype)	
	ICT Object (or stereotype)	Component (or stereotype)	

Integration Type	Jira Project	Jira Project Version	Jira Project Component
	ICT Object (or stereotype)	Standard Platform (or stereotype)	
Jira Projects	ICT Object (or stereotype)		
Project Versions		Application (or stereotype)	
		Component (or stereotype)	
		Standard Platform (or stereotype)	
Project Versions and Project Components	ICT Object (or stereotype)	Application (or stereotype)	Local Component (or stereotype)
	ICT Object (or stereotype)	Component (or stereotype)	Local Component (or stereotype)
	ICT Object (or stereotype)	Standard Platform (or stereotype)	Platform Element (or stereotype)
Jira Projects, Project Versions, and Project Components		Application (or stereotype)	Local Component (or stereotype)
		Component (or stereotype)	Local Component (or stereotype)
		Standard Platform (or stereotype)	Platform Element (or stereotype)
Project Components		Component (or stereotype)	Local Component (or stereotype)
		Standard Platform (or stereotype)	Platform Element (or stereotype)

You may specify any level in the Jira project structure to begin the integration with. For example, you could choose to map Alfabet applications to Jira project versions and the application's local components to Jira project components. Depending on the integration pattern specified for the relevant data connection, users can navigate to the object profile of the relevant ICT object, project, application, component, standard platform, etc. in the Alfabet user interface and link the object to the relevant Jira project/project version/project component via the **Link to Jira Project** option in the **Action** menu. The **Link to Jira Project** option will only be available for those objects that are permissible for Jira integration via the stereotype definition of configured report.

Once the Alfabet object has been linked with the Jira project in the user interface, users can regularly synchronize the object with the Jira project structure via the **Synchronize with Jira Project Structure** option. The project version and project component will only be synchronized if Alfabet objects exist for the relevant object classes/object class stereotypes. Furthermore, the Alfabet objects will be synchronized with the project's existing project version and component only if the properties specified for the mapping have the same value. For example, if the integration pattern specifies that applications are to be mapped to Jira project versions based on the Alfabet property `Name`, then all applications with the same name that are owned by the selected ICT object will be mapped to the same Jira project version.

The following information is available:

- Creating a Jira Connection for Project-Based Integration
- Creating a Jira Connection for Architecture-Based Integration

Creating a Jira Connection for Project-Based Integration

The **Jira Connection** view in the **Integration Solutions Configuration** functionality allows you to define Jira connections that map project stereotypes in Alfabet that are relevant for Jira integration. You can specify an integration mapping of projects in the project hierarchy configured for your Alfabet solution to the Jira project, project version, and project component. For example, the project stereotype Agile Release Train could be mapped to Jira projects and the project stereotype Program Increment could be mapped to the Jira project version.

You can configure multiple semantic Jira connections for each data connection configured in the XML element ***DataConnection*** in the XML object ***JiraIntegrationConfig***. A semantic Jira connection should be configured for each level in the Jira project structure that the integration can begin with.

The following integration patterns may be specified for a project-based approach:

Integration Type	Jira Project	Jira Project Version	Jira Project Component
Jira Projects, Project Versions, and Project Components	**Project** Stereotype 1	**Project** Stereotype 2	**Project** Stereotype 3
Jira Projects and Project Versions	**Project** Stereotype 1	**Project** Stereotype 2	

Integration Type	Jira Project	Jira Project Version	Jira Project Component
Jira Project and Project Components	**Project** Stereotype 1		**Project** Stereotype 3
Jira Projects	**Project** Stereotype 1		
Jira Project Versions		**Project** Stereotype 2	
Jira Project Versions and Project Components		**Project** Stereotype 2	**Project** Stereotype 3
Jira Project Components			**Project** Stereotype 3

To configure a Jira connection for a project-based integration:

1) Go to the **Integration Solutions Configuration** functionality and click the **Jira Connection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create Jira for Project-Based Integration**.
3) In the **Jira Connection for Project-Based Integrations** editor, define the following fields as needed.

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each Jira data connection. This number cannot be edited.
   - **Name**: Enter a unique name for the Jira connection. The name should help the user synchronizing the Jira instance that will be targeted by the connection.
   - **Description**: Enter a meaningful description that will clarify the purpose of the Jira connection.

   Authorized Access tab:

   - **Authorized User**: Click the **Search** icon to assign an authorized user to the selected Jira connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
   - **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

   Connection tab:

Please note the following about the fields in the **Connection** tab:

- The integration pattern that you select in the **Integration Type** field will determine which fields must be defined in the **Connection** tab. Please note that if you specify the integration pattern that starts with either a project version or project component, the **Property to Save Jira Project ID**, **Property to Save Jira Project Name**, and **Property to Save Jira Project Key** must be specified for the ascendant Jira project. The **Property to Save Jira Project ID**, **Property to Save Jira Project Name**, and **Property to Save Jira Project Key** fields are filled with all custom properties as well as the standard properties of the type `String` specified in the XML element `AlfabetClassMappingStandardPropertySettings` in the XML object ***JiraIntegrationConfig***.
- The **Alfabet Class Aligned with Jira Project** field will always display the object class `Project`. You should define the **Stereotype Filter** field in order to determine which project stereotype may be integrated with Jira. All available project stereotypes availed in the **Stereotype Filter** field will be selected if no project stereotype is selected. Please note that the project stereotypes available in the **Stereotype Filter** fields will be determined by the configuration of project stereotype hierarchies in the XML object ***ProjectManager***. For more information, see the section *Creating Project Stereotypes for the Project Hierarchy* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- The definition of the **Stereotype Filter** field and the **Object Filter Report** field determine the Alfabet project's that are allowed to integrate with Jira projects. The **Link to Jira Project** editor option will be available in the **Action** menu in the object profiles of the projects found via the definition of the **Stereotype Filter** field or the **Object Filter Report** field. For more information about configuring the reports that may be selected in the **Object Filter Report** field, see the section *Configuring Object Filter Reports*.

- **Data Connection**: Select the relevant data connection configured in the XML element ***DataConnection*** in the XML object ***JiraIntegrationConfig***. This data connection will be established when the user selects the integration pattern defined below via the **Action** > **Link to Jira Project** option in the object profile for the relevant project stereotype.
- **Project Filter**: Select one of the Jira project filters defined in the XML object ***JiraIntegrationConfig***. The filters limit the number of Jira projects that users can select for the integration in the editor that opens when the **Link to Jira Project** option is executed for a project in Alfabet. For example, a filter could specify that only Jira projects with a project type equal to "Software" or "Business" are returned.
- **Project Filter Description**: Displays information about the filter selected in the **Select Project Filter** field.
- **Integration Type**: Select the level in the Jira project structure to begin the integration with. The project stereotypes displayed in the **Integration Pattern** field will be determined based on the **Integration Type** field. You can select the specific Alfabet project stereotype to map to the relevant level in the Jira project structure in the respective **Stereotype Filter** field.
- **Integration Pattern**: Select the architecture pattern to map Alfabet projects to the Jira project structure selected in the **Integration Type** field.
- **Alfabet Class Aligned with Jira Project**: Displays the object class `Project`.

- **Stereotype Filter**: Select an Alfabet project stereotype to map to Jira projects. All available project stereotypes will be selected if no project stereotype is selected.
- **Object Filter Report**: Select the Alfabet report that returns the relevant objects that may be integrated with Jira objects. This is an optional report that allows you to limit the projects available to be selected for Jira integration.
- **Property to Save Jira Project ID**: Select the Alfabet class property to be used to capture the `ID` property for Jira projects.
- **Property to Save Jira Project Name**: Select the Alfabet class property to be used to capture the `Name` property for Jira projects.
- **Property to Save Jira Project Key**: Select the Alfabet class property to be used to capture the `Key` property for Jira projects.
- **Alfabet Class Aligned with Jira Project Version**: Displays the object class `Project`.
- **Stereotype Filter**: Select an Alfabet project stereotype to map to Jira project versions. All available project stereotypes will be selected if no project stereotype is selected.
- **Object Filter Report**: Select the Alfabet report that returns the relevant objects that may be integrated with Jira project versions. This is an optional report that allows you to limit the project versions available to be selected for Jira integration.
- **Property to Save Jira Project ID**: Select the Alfabet class property to be used to capture the `ID` property for Jira project versions.
- **Property to Save Jira Project Name**: Select the Alfabet class property to be used to capture the `Name` property for Jira project versions.
- **Property to Create Jira Version on Synchronization**: Select the Alfabet class property to be used to capture the `Name` property when creating new Jira project versions.
- **Alfabet Class Aligned with Jira Project Component**: Displays the object class `Project`.
- **Stereotype Filter**: Select an Alfabet project stereotype to map to Jira project components. All available project stereotypes will be selected if no project stereotype is selected.
- **Object Filter Report**: Select the Alfabet report that returns the relevant objects that may be integrated with Jira project components. This is an optional report that allows you to limit the project components available to be selected for Jira integration.
- **Property to Save Jira Project ID**: Select the Alfabet class property to be used to capture the `ID` property for Jira project components.
- **Property to Save Jira Project Name**: Select the Alfabet class property to be used to capture the `Name` property for Jira project components.
- **Event Templates for the Synchronization**: Select the Alfabet event templates that shall call the relevant ADIF import/export schemes that are to be triggered when synchronizing an Alfabet object with the linked Jira object's data. For details about the configuration of event templates necessary to export Alfabet data to Jira, see the section *Configuring the Event Templates to Trigger the ADIF Export Schemes for Synchronization*.
- **Event Template Sequence**: Reorder the sequence that the selected Alfabet event templates are triggered. To do so, click an event template in the window and click the

4) Repeat this for all necessary configured ***DataConnection*** elements in the XML object ***JiraIntegrationConfig***.

Creating a Jira Connection for Architecture-Based Integration

The **Jira Connection** view in the **Integration Solutions Configuration** functionality allows you to define Jira connections that specify the mapping of the object classes or their object class stereotypes in Alfabet that are relevant for Jira integration. You can specify an integration mapping based on predefined architecture object hierarchies in Alfabet to map to the Jira project, project version, and project component. A semantic definition should be configured for each level in the Jira project structure that the integration can begin with.

The following integration patterns may be specified for an architecture-based approach:

Integration Type	Jira Project	Jira Project Version	Jira Project Component
Jira Projects, Project Versions, and Project Components	ICT Object (or stereotype)	Application (or stereotype)	Local Component (or stereotype)
	ICT Object (or stereotype)	Component (or stereotype)	Local Component (or stereotype)
	ICT Object (or stereotype)	Standard Platform (or stereotype)	Platform Element (or stereotype)
Jira Projects and Project Versions	ICT Object (or stereotype)	Application (or stereotype)	
	ICT Object (or stereotype)	Component (or stereotype)	
	ICT Object (or stereotype)	Standard Platform (or stereotype)	
Jira Projects	ICT Object (or stereotype)		
Project Versions		Application (or stereotype)	
		Component (or stereotype)	

Integration Type	Jira Project	Jira Project Version	Jira Project Component
		Standard Platform (or stereotype)	
Project Versions and Project Components	ICT Object (or stereotype)	Application (or stereotype)	Local Component (or stereotype)
	ICT Object (or stereotype)	Component (or stereotype)	Local Component (or stereotype)
	ICT Object (or stereotype)	Standard Platform (or stereotype)	Platform Element (or stereotype)
Jira Projects, Project Versions, and Project Components		Application (or stereotype)	Local Component (or stereotype)
		Component (or stereotype)	Local Component (or stereotype)
		Standard Platform (or stereotype)	Platform Element (or stereotype)
Project Components		Component (or stereotype)	Local Component (or stereotype)
		Standard Platform (or stereotype)	Platform Element (or stereotype)

You can configure multiple semantic data connections for each data connection configured in the XML element ***DataConnection*** in the XML object ***JiraIntegrationConfig***.

To configure a Jira database connection for an architecture-based integration:

1) Go to the **Integration Solutions Configuration** functionality and click the **Jira Connection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create Jira for Architecture-Based Integration**.
3) In the **Jira Connection for Architecture-Based Integrations** editor, define the following fields as needed.

   Basic Data tab:

- **ID**: Alfabet assigns a unique identification number to each Jira data connection. This number cannot be edited.

- **Name**: Enter a unique name for the Jira connection. The name should help the user synchronizing the Jira instance that will be targeted by the connection.
- **Description**: Enter a meaningful description that will clarify the purpose of the Jira connection.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected Jira connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
- **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

Connection tab:

Please note the following about the fields in the **Connection** tab:

- The integration pattern that you select in the **Integration Type** field will determine which fields must be defined in the **Connection** tab. Please note that if you specify the integration pattern that starts with either a project version or project component, the **Property to Save Jira Project ID**, **Property to Save Jira Project Name**, and **Property to Save Jira Project Key** must be specified for the ascendant Jira project. The **Property to Save Jira Project ID**, **Property to Save Jira Project Name**, and **Property to Save Jira Project Key** fields are filled with all custom properties as well as the standard properties of the type `String` specified in the XML element `AlfabetClassMappingStandardPropertySettings` in the XML object ***JiraIntegrationConfig***.
- You may define the **Stereotype Filter** field in order to determine which object class stereotype may be integrated with Jira. All available object class stereotypes available in the **Stereotype Filter** field will be selected if no object class stereotype is selected.
- The definition of the **Stereotype Filter** field and the **Object Filter Report** field determine the Alfabet object's that are allowed to integrate with Jira projects. The **Link to Jira Project** editor option will be available in the **Action** menu in the object profiles of the objects found via the definition of the **Stereotype Filter** field or the **Object Filter Report** field. For more information about configuring the reports that may be selected in the **Object Filter Report** field, see the section *Configuring Object Filter Reports*.
- **Alfabet Class Aligned with Jira Project**: Specify the Alfabet class in the specified integration pattern that is matched to Jira projects. Whether this field can be defined or not will depend on the selection in the **Integration Type** field.

- **Data Connection**: Select the relevant data connection configured in the XML element ***DataConnection*** in the XML object ***JiraIntegrationConfig***. This data connection will be established when the user selects the integration pattern defined below via the **Action** > **Link to Jira Project** option in the object profile for the relevant object.
- **Project Filter**: Select one of the Jira project filters defined in the XML object ***JiraIntegrationConfig***. The filters limit the number of Jira projects that users can select for the integration in the editor that opens when the **Link to Jira Project** option is executed for

an object. For example, a filter could specify that only Jira projects with a project type equal to "Software" or "Business" are returned.

- **Project Filter Description**: Displays information about the filter selected in the **Select Project Filter** field.
- **Integration Type**: Select the level in the Jira project structure to begin the integration with. The object classes displayed in the **Integration Pattern** field will be determined based on the **Integration Type** field. You can select the specific object stereotype to map to the relevant level in the Jira project structure in the respective **Stereotype Filter** field.
- **Integration Pattern**: Select the architecture pattern to map Alfabet objects to the Jira project structure selected in the **Integration Type** field.
- **Alfabet Class Aligned with Jira Project**: Displays the Alfabet class in the specified integration pattern that is mapped to Jira projects. Whether this field can be defined or not will depend on the selection in the **Integration Type** field.
- **Stereotype Filter**: Select an object stereotype to map to Jira projects. All available object class stereotypes will be selected if no object class stereotype is selected.
- **Object Filter Report**: Select the Alfabet report that returns the relevant objects that may be integrated with Jira projects. This is an optional report that allows you to limit the objects available to be selected for Jira integration.
- **Property to Save Jira Project ID**: Select the Alfabet class property to be used to capture the `ID` property for Jira projects.
- **Property to Save Jira Project Name**: Select the Alfabet class property to be used to capture the `Name` property for Jira projects.
- **Property to Save Jira Project Key**: Select the Alfabet class property to be used to capture the `Key` property for Jira projects.
- **Alfabet Class Aligned with Jira Project Version**: Displays the Alfabet class in the specified integration pattern that is mapped to Jira project versions. Whether this field can be defined or not will depend on the selection in the **Integration Type** field.
- **Stereotype Filter**: Select an object class stereotype to map to Jira project versions. All available object class stereotypes will be selected if no object class stereotype is selected.
- **Object Filter Report**: Select the Alfabet report that returns the relevant objects that may be integrated with Jira project versions. This is an optional report that allows you to limit the objects available to be selected for Jira integration.
- **Property to Save Jira Project ID**: Select the Alfabet class property to be used to capture the `ID` property for Jira project versions.
- **Property to Save Jira Project Name**: Select the Alfabet class property to be used to capture the `Name` property for Jira project versions.
- **Property to Create Jira Version on Synchronization**: Select the Alfabet class property to be used to capture the `Name` property when creating new Jira project versions.
- **Alfabet Class Aligned with Jira Project Component**: Displays the Alfabet class in the specified integration pattern that is mapped to Jira project components. Whether this field can be defined or not will depend on the selection in the **Integration Type** field.

- **Stereotype Filter**: Select an object class stereotype to map to Jira project components. All available object class stereotypes will be selected if no object class stereotype is selected.
- **Object Filter Report**: Select the Alfabet report that returns the relevant objects that may be integrated with Jira project components. This is an optional report that allows you to limit the objects available to be selected for Jira integration.
- **Property to Save Jira Project ID**: Select the Alfabet class property to be used to capture the `ID` property for Jira project components.
- **Property to Save Jira Project Name**: Select the Alfabet class property to be used to capture the `Name` property for Jira project components.
- **Event Templates for the Synchronization**: Select the Alfabet event templates that shall call the relevant ADIF import/export schemes that are to be triggered when synchronizing an Alfabet object with the linked Jira object's data. For details about the configuration of event templates necessary to export Alfabet data to Jira, see the section *Configuring the Event Templates to Trigger the ADIF Export Schemes for Synchronization*.
- **Event Template Sequence**: Reorder the sequence that the selected Alfabet event templates are triggered. To do so, click an event template in the window and click the

4) Repeat this for all necessary configured ***Connection*** elements in the XML object ***JiraIntegrationConfig***.

Specifying MS Project Methodologies for Interoperability with Microsoft Project

Interoperability with Microsoft® Project allows your enterprise to synchronize strategic project portfolio planning with operational project management and evaluate all of the projects in the enterprise in the context of corporate strategy and strategic investment. The capability allows the enterprise to export projects planned in Alfabet to Microsoft Project as well as to import projects managed in Microsoft Project to Alfabet.

The following scenarios are supported:

- Project planning has been conducted in Microsoft Project and the projects are to be imported to the enterprise's project structure in Alfabet to seed the enterprise's strategic project portfolio planning. A methodology to map the Microsoft projects/project tasks to Alfabet projects/skill requests must be specified for the import.
- Operational project planning has been captured and scoped in Alfabet and the projects will be exported to an MPP file. A methodology to map Alfabet projects/skill requests to Microsoft projects/project tasks must be specified for the export.
- Projects exported from Microsoft Project to Alfabet can be synchronized via an MPP file so that data potentially modified in Microsoft Project will be updated to the corresponding project structure in Alfabet.

In order to import and export projects from/to Microsoft Project, you must configure MS project methodologies that describe how to map the projects and project tasks captured in Microsoft Project to the Alfabet project structure. An MS project methodology includes basic information such as the data connection to use, the project stereotypes relevant for the import, the mapping of data in Microsoft Project to the Alfabet

properties for projects and skill requests, as well as the import matching rules to find the relevant skills, organizations, and persons referenced by imported skill requests.

Users triggering an import or export can select the relevant MS project methodology when defining the import/export. The functionalities to import, export, and synchronize projects are available in the **Capture Projects** functionalities, the **Projects** page views for project groups and buckets, and the **Project** object profile or a project stereotype object profile.

Users can import a single project to Alfabet by importing an MPP file, multiple projects to Alfabet by importing multiple MPP files in a ZIP file, or one or more projects via a connection to Microsoft® Project Server® 2013 or 2016. Upon import, the MS project hierarchy will be imported based on the project hierarchy specified in the MS project methodology. All resources specified for the MS projects and project task that are imported to Alfabet will be imported as skill requests that are assigned to the relevant imported project in Alfabet. Any resources assigned to MS projects that are below the projects imported to the leaf-level of the Alfabet project hierarchy will also be imported as skill requests and will be imported to the leaf-level project that was imported.

Users can also export Alfabet projects to an MPP file. Projects that are updated in the exported MPP file can be synchronized so that the projects in the Alfabet database are updated with the changes that have been made to the MS projects.

For an overview of all configuration steps required in order to work with the Microsoft® Project interoperability, see the section *Configuring Interoperability with Microsoft Project* in the reference manual *API Integration with Third-Party Components*. These configuration steps should be completed before defining MS project methodologies in the **Integration Solutions** functionality.

The following information is available:

- Configuring MS Project Methodologies
- Importing One or More Microsoft Projects to Alfabet via an MPP File
- Importing a Microsoft Project via the Microsoft Project Server
- Exporting Alfabet Projects to Microsoft Project
- Synchronizing Exported Alfabet Projects with Their Updated Microsoft Projects

Configuring MS Project Methodologies

MS project methodologies can only be specified in the **Integration Solutions Configuration** functionality if the prerequisite configuration has been completed. For a detailed description of interoperability with Microsoft® Project including all required configuration steps in Alfabet, the section *Configuring Interoperability with Microsoft Project* in the reference manual *API Integration with Third-Party Components*.

A methodology to map the Microsoft projects/project tasks to Alfabet projects/skill requests must be specified in the **MS Project Methodology** node in the **Integration Solutions Configuration** functionality. Please note the following:

- All levels of project stereotypes in the project hierarchy specified in the XML object ***ProjectManager*** may be mapped to the Microsoft project hierarchy. If more than one project stereotype is defined as a child of a project stereotype in the XML object ***ProjectManager***, the

user specifying the methodology must define which of the child stereotypes shall be used to create the sub-project.

- The Microsoft project will be imported to Alfabet as the parent project of the imported project hierarchy. The project stereotype to use to import the Microsoft project must be defined in the first row of the **Project Stereotype Hierarchy and Default Statuses** field in the **MS Project Methodology** editor.
- The import of project tasks will depend on your enterprise's project management methodology. The following is possible:

  1) Project tasks can be imported as sub-projects to Alfabet.

     - For each Alfabet project stereotype defined in the **Project Stereotype Hierarchy and Default Statuses** field, the corresponding project task in the Microsoft project hierarchy will be imported to Alfabet. All but the leaf-level project tasks will be imported to Alfabet as projects.
     - All leaf-level project tasks in each branch of the Microsoft project hierarchy that have resources defined will be imported to Alfabet as skill requests. In other words, any project task with a resource definition that has no subordinate project tasks will be imported to Alfabet as a skill request. Please consider the following import scenarios:

       For example:

       - If 3 levels of project stereotypes are specified for the import and the Microsoft project task hierarchy only has 3 hierarchy levels (Level 1. Project, Level 2. Project Task, Level 3. Project Task), Level 1 and 2 would be imported as projects and Level 3 would be imported as a skill request if resources have been defined for the project task.
       - If 3 levels of project stereotypes are specified for the import and the Microsoft project task hierarchy has 5 hierarchy levels, Level 1, 2, and 3 would be imported as projects and Level 4 and Level 5 would be imported as a skill request if resources have been defined for the project task. These skill requests would be assigned to the projects on Level 3.
       - If one level of project stereotypes is specified for the import and the Microsoft project task hierarchy only has 3 hierarchy levels (Level 1. Project, Level 2. Project Task, Level 3. Project Task), Level 1 would be imported as projects and Level 2 and Level 3 would be imported as a skill request if resources have been defined for the project tasks.

  2) Project tasks with resources defined can be imported as skill requests to Alfabet.

     - If only one Alfabet project stereotype is defined in the **Project Stereotype Hierarchy and Default Statuses** field, then only the Microsoft project will be imported as a project. No subordinate projects will be imported.
     - All project tasks in the Microsoft project hierarchy that have resources will be imported as skill requests to Alfabet. The skill requests will be assigned directly to the imported Alfabet project imported.

       If one level of project stereotypes is specified for the import and the Microsoft project task hierarchy only has 3 hierarchy levels (Level 1. Project, Level 2. Project Task, Level 3. Project Task), Level 1 would be imported as projects and

Level 2 and Level 3 would be imported as a skill request if resources have been defined for the project tasks.

- The creation of new skill requests in Alfabet based on imported project tasks requires the specification of import matching rules. Import matching rules describe how the resource data captured in Microsoft Project shall be used to find the skill, organization, and person referenced by a skill request created in the context of importing a Microsoft project task. Each import matching rule may have one or more properties or property patterns defined. The import matching rules that can be selected in the context of defining an MS project methodology have been configured by your solution designer. For more information about the configuration of import matching rules, see the section *Configuring the Connections for Interoperability with Microsoft Project* in the reference manual *API Integration with Third-Party Components*. Please note the following:
  - An import matching rule must be defined to find the references to the relevant skill in the Alfabet database for the imported skill request. Every skill request that is imported to Alfabet requires a reference to a skill. If no skill is found via an import matching rule, the default skill specified by the user triggering the import will be used.

    For example, if a matching rule `Name` is defined for the class `Skill` and a project task is imported as a skill request to Alfabet and the project task has a resource Java Programming, the system will search in the Alfabet database for a skill with the name JAVA Programming. If an existing skill is found, the new skill request will reference the skill Java Programming. If no skill is found, the default skill defined by the user triggering the import will be used.

  - An import matching rule must be defined to find the references to the relevant organization and the relevant person in the Alfabet database for the imported skill request. However, references to an organization and/or person are optional for a skill request. If no organizations or persons are found, these references will not be set for the imported skill request.
  - Import patterns with criteria such as `Contains`, `Starts With`, etc. can be used to find the existing skills, organizations, and persons in the Alfabet database to map the Microsoft project information to upon importing MPP files to Alfabet.
  - The first relevant skill, organization, or person found to match the criteria in the Alfabet database will be used.
  - The language in Microsoft Project to use for the matching may also be defined. However, all strings imported to Alfabet will be stored in the Alfabet database in the primary culture.

The **MS Project Methodology** view allows you to define multiple MS project methodologies that describe how to import the projects and tasks captured in Microsoft Project to the project structure in Alfabet. The methodology includes basic information such as the data connection to use, the project stereotypes relevant for the import, the mapping of data in Microsoft Project to the Alfabet properties for projects and skill requests, as well as the import matching rules to find the relevant skill, organization, and person referenced by the imported skill request. Users triggering an import or export can select the relevant MS project methodology when defining the import/export.

Please note that if changes are made to the XML object ***MicrosoftProjectPlanConfig*** after the MS project methodologies have been created in the **MS Project Methodology** view in the **Integration Solutions Configuration** functionality, you should review the import and export mapping rules of the MS project methodologies and make adjustments as needed. Changes made to the XML object ***MicrosoftProjectPlanConfig*** will not be automatically updated to existing MS project methodologies. For more information about configuring MS project methodologies, see

the section Specifying MS Project Methodologies for Interoperability with Microsoft Project in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

Changes should not be made to existing MS project methodologies that have already been used for import, export, or synchronization. If changes need to be made to existing MS project methodologies, you should make a copy of an existing MS project methodology and modify it as needed.

To configure a methodology to import Microsoft projects to Alfabet:

1) Go to the **Integration Solutions Configuration** functionality and click the **MS Project Methodology** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create MS Project Methodology**.

To create a new MS project methodology based on an existing MS project methodology, select the MS project methodology in the table and click **New** > **Create MS Project Methodology as Copy**. Proceed as described below.

3) In the **MS Project Methodology** editor, define the following fields as needed.

Basic Data tab:

- **ID**: Alfabet assigns a unique identification number to each MS project methodology. This number cannot be edited.
- **Name**: Enter a unique name for the MS project methodology. The name will be displayed as an option in the **MS Project Methodology to Import MS Project File(s)** field in the **Project** wizard that will be used to trigger the import in the Alfabet user interface.
- **Release Status**: Select the release status for the MS project methodology name. Release statuses must be configured for the class `MicrosoftProjectPlan_Mappings` in the XML object ***ReleaseStatusDefs***. Only methodologies with the release status that is configured as the `ApprovedStatus` can be selected for import or export in the **MS Project Methodology to Import MS Project File(s)** field in the **Project** wizard.
- **Currency**: This field is only relevant for the export capability.
- **Description**: Enter a meaningful description that will clarify the purpose of the MS project methodology for end users. The description will be displayed in the **Description** field of the wizard step in which the import, export, or synchronization is triggered.
- **Project Stereotype Hierarchy and Default Statuses**: These fields allows you to define the project stereotypes to use to import projects to the Alfabet project hierarchy. A row will be displayed for each hierarchy level specified in the XML object ***ProjectManager***. Each row represents a level in the project hierarchy, beginning with the parent project at the top. Specify the project stereotype to map to the corresponding MS project/project task in the **Stereotype Name** column specify the default status for that project stereotype in the **Status** column. Please note the following:
  - The first row in the **Project Stereotype Hierarchy and Default Statuses** field is mandatory. If only the first row is specified, the Microsoft project will be imported but no subordinate projects will be imported. In this case, all project tasks will be imported as skill requests.
  - If project tasks shall be imported as subordinate projects to Alfabet, you may define as many levels in the project hierarchy that shall be imported. For each project stereotype

defined in the methodology, the corresponding project/project task will be imported. The project stereotypes displayed in the **Stereotype Name** field will depend on the definition of the parent project stereotype.

- A default release status must be specified for each project stereotype specified in the **Stereotype Name** column. Each project in Alfabet created via the import will be assigned the default status specified for its project stereotype. The default statuses specified in the XML object ***ProjectManager*** as permissible for the selected project stereotype will be displayed.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected MS project methodology. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the MS project methodology.
- **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

Import/Export Matching tab:

- **Import Matching Rules**: Depending on the methodology you define, a skill request will be created for the projects or project tasks imported from Microsoft project. Specify the import matching rules that describe how the resource data captured in Microsoft Project shall be used to find the skill, organization, and person referenced by a skill request created in the context of importing a Microsoft project task.

  Please note that if the names of matching patterns have more than 64 characters, errors may occur when specifying the MS project methodology.

  - **Organization**: Select the properties or matching patterns to use to find the organizations referenced by the skill requests created in the Alfabet database.
  - **Matching Pattern for Organization**: Specify whether the name of the resource in MS Project must be an exact match, contain, start with, or end with the criteria specified in the **Organization** field.
  - **Skill**: Select the properties or matching patterns to use to find the skills referenced by the skill requests created in the Alfabet database.
  - **Matching Pattern for Skill**: Specify whether the name of the resource in MS Project must be an exact match, contain, start with, or end with the criteria specified in the **Skill** field.
  - **Person**: Select the properties or matching patterns to use to find the persons referenced by the skill requests created in the Alfabet database.
  - **Matching Pattern for Person**: Specify whether the name of the resource in MS Project must be an exact match, contain, start with, or end with the criteria specified in the **Person** field.
- **Export Matching Rules**: Specify the following information to export Alfabet projects to an MPP file.
  - **Organization**: Specify the property or matching pattern to create the resource in the MPP file that is based on the organization associated with the exported skill request.

- **Skill**: Specify the property or matching pattern to create the resource in the MPP file that is based on the skill associated with the exported skill request.
- **Person**: Specify the property or matching pattern to create the resource in the MPP file that is based on the person associated with the exported skill request.
- **ID Mapping Field**: Specify the MS project field where the REFSTR property should be stored for the Alfabet project or skill request that is exported to MS project.
- **Match MS Project Property with Project Alias**: If an Alfabet project is exported to an MPP file, the data may be updated in the MPP file and the changed data can be synchronized to Alfabet, thus updating the data in the Alfabet database. Select the checkbox to ensure that the MPP file that is used for the synchronization is the same MPP file that the MS projects have been exported to.

- **Select MS Template File**: Specify the following information to export Alfabet projects to an MPP file.
  - **Existing MPP Template**: Select the Microsoft project template (MPT) that you want to use when creating MPP files with the exported Alfabet data. The MPT file is used to define various aspects for the exported MPP file such as the calendar, working days, weekly hours, custom columns, filters, groups, and reports. The drop-down field will display all MS templates uploaded via the **Load New Template** field as well as the default MS template provided by Alfabet.

    The MPT file must have the following settings defined in Microsoft Project in order to be accepted by Alfabet as a valid template:

    - The MPT file must be saved in Microsoft Project 2010 or higher.
    - The Resource Sheet setting must not be set in the MPT file.
    - Tasks must not be defined in the MPT file.
    - The Gant Chart setting must be set in the MPT file. (Only Gantt charts are allowed for interoperability with Alfabet.
    - The `24 Hrs Calendar` must be used in the MPT file.
    - The `Work` column must be visible column in the MPT file.
    - All uploaded MPT files are stored in the **Internal Document Selector** can only be used once. If you want to use the same MPT file for a different MS project methodology, you must save a copy of the MPT file name with a different name that doesn't already exist in the `Templates` folder in the `SYSTEM` folder of the **Internal Document Selector**. Please note that if you change the MPT file and want the MPT file deleted, it must be explicitly removed from the `IDOC:\SYSTEM\TEMPLATE` folder in the **Internal Document Selector**.

  - **Load New MPP Template**: Upload MS templates that you want to use to create MPP files with the exported Alfabet data. The uploaded file will be saved to the **Internal Document Selector**.

- **Import Language**: Select the language to import projects and skill requests. If data translation is supported for the relevant properties for projects and skill requests the import language will

be applied to the import matching rules. Please note that this setting applies only to the import of MS projects.

Map MS Project to Alfabet Project tab: Map the Microsoft project properties to corresponding properties of the class `Project`. This is required to import and export projects.

- **Alfabet Property Name**: Lists the technical name of all standard and custom Alfabet properties defined for the class `Project`.
- **Alfabet Property Caption**: Lists the caption of the corresponding standard and custom Alfabet properties defined for the class `Project`.
- **MPP Property Name**: Displays all standard properties for MS projects displayed. Please note that no custom properties will be displayed in the field and therefore cannot be imported to Alfabet. Select the relevant MS project property that shall be matched to the Alfabet project property.

  Please note that software from Aspose Party Ltd. is used as middleware for interoperability with Microsoft Project and does not support reading the `Status` field in Microsoft project. Therefore, it is not possible to map to the MPP property `Status`.

  Please note that when exporting Alfabet projects via an MPP file to Microsoft Project, regardless of the mapping definition of the **End Date** attribute for Alfabet projects, the Finish Date of the summary project will be the same as its Start Date in MS Project. This is an due to mechanisms of Microsoft Project. Therefore, you must explicitly change the Finish Date in the MPP file to align it with the **End Date** attribute of the project in Alfabet.

  For subordinate projects and skill requests that are exported from Alfabet to MS Project, the Start Date and Finish Date of the imported project tasks will be updated based on the Alfabet start and end dates if the dates are within the start and end date of their parent project in Alfabet.

Map MS Project Task to Alfabet Project tab: Map MS project task properties to properties of the class `Project`. Mapping properties in this tab is only relevant if your enterprise wants to map MS project tasks to projects in Alfabet. If your enterprise wants to map all relevant MS project tasks to skill requests, skip this tab and go to the **Map MS Project Task to Alfabet Skill Request** tab

- **Alfabet Property Name**: Lists the technical name of all standard and custom Alfabet properties defined for the class `Project`.
- **Alfabet Property Caption**: Lists the caption of the corresponding standard and custom Alfabet properties defined for the class `Project`.
- **MPP Property Name**: Displays all standard properties for MS project tasks displayed. Please note that no custom properties will be displayed in the field and therefore cannot be mapped to Alfabet properties. Select the relevant MS project property that shall be matched to the Alfabet project property.

Map MS Project Task to Alfabet Skill Request tab: Map Microsoft project task properties to properties for the class `SkillRequest`. Please note that the Alfabet properties `Person` and `Organization` cannot be mapped to an MPP property. You must define matching rules in the **Import Matching Rules** tab to find the skill, organization, and person referenced by a skill request created in the context of importing a Microsoft project task.

- **Alfabet Property Name**: Lists the technical name of all standard and custom Alfabet properties defined for the class `SkillRequest`.
- **Alfabet Property Caption**: Lists the caption of the corresponding standard and custom Alfabet properties defined for the class `SkillRequest`.
- **MPP Property Name**: Displays all standard properties for MS project tasks displayed. Please note that no custom properties will be displayed in the field and therefore cannot be mapped to Alfabet properties. Select the relevant MS project property that shall be matched to the Alfabet project property.

4) Click **OK** to save the MS project methodology.

Importing One or More Microsoft Projects to Alfabet via an MPP File

Alfabet projects can be imported from a Microsoft Project Plan file in the *Capture Projects Functionality*, the *Projects Page View* in the object profile for the class **Bucket**, and the *Projects Page View* in the object profile for the class **Project Group**.

If your enterprise's project planning has been conducted in Microsoft Project, you can import the projects to Alfabet to seed the enterprise's strategic project portfolio planning. When you import a Microsoft project to Alfabet, a new project is created in the Alfabet database. The import functionalities will be available for the first-time import of a project from Microsoft Project to Alfabet.

You can import a single project to Alfabet by importing an MPP file or multiple projects to Alfabet by importing multiple MPP files in a ZIP file. The MS project will be imported, and a new project based on the project stereotype selected in the **Project Type** filter in the **Capture Projects** view will be created. For more information about importing projects via a connection to Microsoft® Project Server® 2013 or 2016, see the section *Importing a Microsoft Project to Alfabet via Microsoft Project Server*.

Please note the following about the import of MS projects and project tasks to Alfabet:

- The Microsoft project will be imported to Alfabet as the parent project of the imported project hierarchy.
- The import of project tasks will depend on your enterprise's project management methodology and the definition of the MS project methodology that you select for the import. The following is possible:

  1) Project tasks will be imported as sub-projects to Alfabet.

     - For each Alfabet project stereotype defined for import in the project hierarchy, the corresponding project task in the Microsoft project hierarchy will be imported to Alfabet. All but the leaf-level project tasks will be imported to Alfabet as projects.
     - All leaf-level project tasks in each branch of the Microsoft project hierarchy that have resources defined will be imported to Alfabet as skill requests. In other words, any project task with a resource definition that has no subordinate project tasks will be imported to Alfabet as a skill request. Please consider the following import scenarios:

       For example:

       - If 3 levels of project stereotypes are specified for the import and the Microsoft project task hierarchy only has 3 hierarchy levels (Level 1. Project, Level 2. Project Task, Level 3. Project Task), Level 1 and 2

would be imported as projects and Level 3 would be imported as a skill request if resources have been defined for the project task.

- If 3 levels of project stereotypes are specified for the import and the Microsoft project task hierarchy has 5 hierarchy levels, Level 1, 2, and 3 would be imported as projects and Level 4 and Level 5 would be imported as a skill request if resources have been defined for the project task. These skill requests would be assigned to the projects on Level 3.
- If one level of project stereotypes is specified for the import and the Microsoft project task hierarchy only has 3 hierarchy levels (Level 1. Project, Level 2. Project Task, Level 3. Project Task), Level 1 would be imported as projects and Level 2 and Level 3 would be imported as a skill request if resources have been defined for the project tasks.

2) Project tasks with resources defined will be imported as skill requests to Alfabet.

- If only one Alfabet project stereotype is defined for import in the project hierarchy, then only the Microsoft project will be imported as a project. No subordinate projects will be imported.
- All project tasks in the Microsoft project hierarchy that have resources will be imported as skill requests to Alfabet. The skill requests will be assigned directly to the imported Alfabet project imported.

If one level of project stereotypes is specified for the import and the Microsoft project task hierarchy only has 3 hierarchy levels (Level 1. Project, Level 2. Project Task, Level 3. Project Task), Level 1 would be imported as projects and Level 2 and Level 3 would be imported as a skill request if resources have been defined for the project tasks.

- Once you have completed the import, you can navigate to the *Project, Skill Request and Resource Request Time Schedule Page View* and review the projects, sub-projects, and skill requests that have been imported. Please note that the naming convention for the imported skill requests will be as follows: <Resource> - <Project Task>. Depending on the mapping defined in the MS project methodology, additional data such as the skill request's start and end dates, man days or head count, and providing organizations or persons may be included in the import and displayed in the *Project, Skill Request and Resource Request Time Schedule Page View*.

To import an MS project and create a new project in Alfabet:

1) In the toolbar, click either:
   - **New** > **Import <Project> from MS Project** to create a single project, or
   - **New** > **Import Multiple <Projects> from MS Project File as Zip Archive** to create multiple projects based on multiple MPP files in a ZIP file.
2) The **<Project>** wizard opens. Enter information in the fields, as needed.
   - **MS Project File to Import:** Select the MPP or ZIP file that you want to import.
   - **MS Project Methodology to Import MS Project File(s)**: Select the MS project methodology to use for the import. MS project methodologies must be configured by your enterprise in the **MS Project Methodology** view in the **Integration Solutions** functionality. For more information, see the section Specifying MS Project Methodologies for Interoperability with Microsoft Project

in the reference manual *Configuring Evaluation and Reference Data in Alfabet*. Confirm the message informing you that projects and skill requests will be created when the **Next** button is clicked.

- **Description**: Displays the description of the MS project methodology selected in the **Methodology to Import MS Project File(s)** field.
- **Project Stereotype**: Displays the project stereotype that the MS project will be imported to. A new project will be created in the Alfabet database based on this project stereotype.
- **Default Skill**: Select a skill that the new skill requests shall reference if no skill is found based on the configured import matching rules specified in the MS project methodology.

3) Click **Next** to trigger the import and create the new project in Alfabet based on the imported MS project.
4) In the next wizard step, edit the information in the fields, as needed.

   Basic Data tab:

- **ID**: Alfabet assigns a unique identification number to each object in the inventory. This number cannot be edited.
- **Name**: Enter a unique name for the project.
- **Status**: Select a level indicating the release status of the project in the planning process. To change the status of the selected project, close this dialog box and click the **Change Status** button in the toolbar.

  The set of release statuses available for an object class are configured by your solution designer in the configuration tool Alfabet Expand. For more information, see the section *Configuring Release Status Definitions for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*. For general information about release statuses, see the section *Understanding Release Statuses* in the reference manual *Getting Started with Alfabet*.

  For more information, see *Changing the Status of the Selected Project*

- **Planned Start Date**: Enter the planned start date of the project in the format appropriate to your culture settings or select the planned start date in the calendar.
- **Planned End Date**: Enter the planned end date of the project in the appropriate format or select the planned end date in the calendar.

  The start date and end date of all skill requests and resource requests must fall within the range of the start and end dates of the project they are assigned to. In other words, the start date of a skill request and resource request may not be before the start date of the parent project and the end date may not be after the end date of the parent project. If a date is outside of this range, the erroneous date will be highlighted red. You can edit the dates by selecting **New** > **Shift Start/End Dates**.

- **Primary Architecture Element**: The primary architecture element that is targeted by the selected project. You can select any application or ICT object that has been added to the *Affected Architecture Page View* of the selected project.

CAPEX costs or non-operational costs defined for the project will be included in the *Lifecycle Costs Page View* and *Lifecycle Costs Chart Page View* for an ICT object selected as primary architecture element or associated with the application selected as the primary architecture element.

- **Description**: Enter a meaningful description that will clarify the purpose of the project.
- **Project Number**: Enter the project number that the selected project is associated with. This project number is usually the ID number used in your company's external multi-project management solution.
- **Currency**: Select the default currency or currency unit to capture and analyze project costs. The currency/currency unit may be changed in the relevant views and reports, as needed. For more information about the configuration of currencies, see the chapter Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.
- **Buckets Equal Allocation**: Select the checkbox if the distribution of costs across buckets assigned to the project is equal. If you select the checkbox, the **Edit Allocation** functionality will be disabled in the *Project Buckets Page View*. Clear the checkbox if the cost allocation should be manually defined for the project. In this case, the **Edit Allocation** functionality will be available in the *Project Buckets Page View*. Please note that if cost allocations have already been defined for the selected project, the cost allocation will be recalculated and updated in the *Project Buckets Page View* if you change the setting of the **Buckets Equal Allocation** checkbox.
- **Cash Out Budget Complete**: Select the checkbox if the budget for the cash out plan has been completed.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected object. The authorized user will have Read/Write access permissions to the object and is authorized to maintain the object in Alfabet.
- **Authorized User Groups**: Select the checkbox to assign Read/Write access permissions to all users in the selected user group.

5) Click **Next** to save your changes.
6) Continue to define the following wizard steps as needed.

Importing a Microsoft Project via the Microsoft Project Server

Alfabet projects can be imported from a Microsoft Project Plan file in the *Capture Projects Functionality*, the *Projects Page View* in the object profile for the class **Bucket**, and the *Projects Page View* in the object profile for the class **Project Group**.

You can import a Microsoft® project to Alfabet via a connection to a Microsoft® Project Server® instance. The connection must be configured by your solution designer. For more information about configuring the connection to an MS Project Server instance, see the section *Configuring Interoperability with Microsoft Project* in the reference manual *API Integration with Third-Party Components*. The MS project will be imported, and a new project based on the project stereotype selected in the **Project Type** filter in the **Capture Projects** view will be created.

For general information about the import of MS projects and project tasks to Alfabet, see the section *Importing One or More Microsoft Projects to Alfabet via MPP Files*.

To import an MS project via a connection to a Microsoft Project Server instance:

1) In the toolbar, click **New** > **Import <Project> from MS Project Server** to create a single project via a connection to the Microsoft Project Server.
2) The **<Project>** wizard opens. Enter information in the fields, as needed.
   - **MS Project Server Connection**: Specify the connection to the Microsoft Project Server instance from which you want to import the project.
   - **Search String**: Enter search criteria to limit the display of projects in the selected MS Project Server instance. Click the **Search** button to display the projects in the MS Project Server instance.
   - **MS Project Methodology to Import MS Project File(s)**: Select the MS project methodology to use for the import. MS project methodologies must be configured by your enterprise in the **MS Project Methodology** view in the **Integration Solutions** functionality. For more information, see the section Specifying MS Project Methodologies for Interoperability with Microsoft Project in the *Configuring Evaluation and Reference Data in Alfabet*. Confirm the message informing you that projects and skill requests will be created when the **Next** button is clicked.
   - **Description**: Displays the description of the MS project methodology selected in the **Methodology to Import MS Project File(s)** field.
   - **Project Stereotype**: Displays the project stereotype that the MS project will be imported to. A new project will be created in the Alfabet database based on this project stereotype.
   - **Default Skill**: Select a skill that the new skill requests shall reference if no skill is found based on the configured import matching rules specified in the MS project methodology.
3) Click **Next** to trigger the import and create the new project in Alfabet based on the imported MS project.
4) In the next wizard step, edit the information in the fields, as needed.

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each object in the inventory. This number cannot be edited.
   - **Name**: Enter a unique name for the project.
   - **Status**: Select a level indicating the release status of the project in the planning process. To change the status of the selected project, close this dialog box and click the **Change Status** button in the toolbar.

     The set of release statuses available for an object class are configured by your solution designer in the configuration tool Alfabet Expand. For more information, see the section *Configuring Release Status Definitions for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*. For general information about release statuses, see the section *Understanding Release Statuses* in the reference manual *Getting Started with Alfabet*.

For more information, see *Changing the Status of the Selected Project*

- **Planned Start Date**: Enter the planned start date of the project in the format appropriate to your culture settings or select the planned start date in the calendar.
- **Planned End Date**: Enter the planned end date of the project in the appropriate format or select the planned end date in the calendar.

  The start date and end date of all skill requests and resource requests must fall within the range of the start and end dates of the project they are assigned to. In other words, the start date of a skill request and resource request may not be before the start date of the parent project and the end date may not be after the end date of the parent project. If a date is outside of this range, the erroneous date will be highlighted red. You can edit the dates by selecting **New** > **Shift Start/End Dates**.

- **Primary Architecture Element**: The primary architecture element that is targeted by the selected project. You can select any application or ICT object that has been added to the *Affected Architecture Page View* of the selected project.

  CAPEX costs or non-operational costs defined for the project will be included in the *Lifecycle Costs Page View* and *Lifecycle Costs Chart Page View* for an ICT object selected as primary architecture element or associated with the application selected as the primary architecture element.

- **Description**: Enter a meaningful description that will clarify the purpose of the project.
- **Project Number**: Enter the project number that the selected project is associated with. This project number is usually the ID number used in your company's external multi-project management solution.
- **Currency**: Select the default currency or currency unit to capture and analyze project costs. The currency/currency unit may be changed in the relevant views and reports, as needed. For more information about the configuration of currencies, see the chapter Configuring Currencies and Currency Exchange Rates for Cost Management Capabilities in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.
- **Buckets Equal Allocation**: Select the checkbox if the distribution of costs across buckets assigned to the project is equal. If you select the checkbox, the **Edit Allocation** functionality will be disabled in the *Project Buckets Page View*. Clear the checkbox if the cost allocation should be manually defined for the project. In this case, the **Edit Allocation** functionality will be available in the *Project Buckets Page View*. Please note that if cost allocations have already been defined for the selected project, the cost allocation will be recalculated and updated in the *Project Buckets Page View* if you change the setting of the **Buckets Equal Allocation** checkbox.
- **Cash Out Budget Complete**: Select the checkbox if the budget for the cash out plan has been completed.

Authorized Access tab:

- **Authorized User**: Click the **Search** icon to assign an authorized user to the selected object. The authorized user will have Read/Write access permissions to the object and is authorized to maintain the object in Alfabet.

- **Authorized User Groups**: Select the checkbox to assign Read/Write access permissions to all users in the selected user group.

5) Click **Next** to save your changes.

6) Continue to define the following wizard steps as needed.

Exporting Alfabet Projects to Microsoft Project

Alfabet projects can be exported to a Microsoft Project Plan file in the *Capture Projects Functionality*, the *Project Object Profile*, the *Projects Page View* in the object profile for the class **Bucket**, and the *Projects Page View* in the object profile for the class **Project Group**.

If operational project planning is captured and scoped in Alfabet, you can export the projects to an MPP file. Export from Alfabet to Microsoft Project Server is not supported. The projects and skill requests will be mapped to MS projects, project tasks, and skill requests as specified in the MS project methodology used for the export.

Projects exported from Alfabet to a Microsoft Project Plan (MPP) file can be updated in the MPP file and the modified data can be synchronized to the corresponding project structure in Alfabet. Alfabet projects can be synchronized with the exported project in the MPP file in the *Project Object Profile*.

Please note that when exporting Alfabet projects via an MPP file to Microsoft Project, regardless of the mapping definition of the **End Date** attribute for Alfabet projects, the Finish Date of the summary project will be the same as its Start Date in MS Project. This is an due to mechanisms of Microsoft Project. Therefore, you must explicitly change the Finish Date in the MPP file to align it with the **End Date** attribute of the project in Alfabet.

For subordinate projects and skill requests that are exported from Alfabet to MS Project, the Start Date and Finish Date of the imported project tasks will be updated based on the Alfabet start and end dates if the dates are within the start and end date of their parent project in Alfabet.

If the name of the Alfabet project has special characters, the special characters will be replaced by a blank space < >. Please note however that the browser may escape the blank space and replace it with an underscore <_>.

To export a project to an MPP file:

1) In the view, select the project that you want to export to an MPP file.

2) In the toolbar, click **New** > **Export Selected <Project> as MS Project File** to create a single project via a connection to the Microsoft Project Server.

3) The **<Project>** wizard opens. Enter information in the fields, as needed.

- **MS Project Methodology to Export MS Project File**: Select the MS project methodology to use for the export. MS project methodologies must be configured by your enterprise in the **MS Project Methodology** view in the **Integration Solutions** functionality. For more information, see the section Specifying MS Project Methodologies for Interoperability with Microsoft Project in the *Configuring Evaluation and Reference Data in Alfabet*.

- **Description**: Displays the description of the MS project methodology selected in the **Methodology to Import MS Project File(s)** field.

4) Click **Export** to trigger the export.

5) In the **Download File** editor, click the **Download** button to save the file.

Synchronizing Exported Alfabet Projects with Their Updated Microsoft Projects

Alfabet projects can be synchronized with the exported project in a Microsoft Project Plan file in the *Project Object Profile*.

Projects exported from Microsoft Project to Alfabet can be synchronized via an MPP file so that data potentially modified in Microsoft Project will be updated to the corresponding project structure in Alfabet.

Please note the following:

- You can only synchronize a project with an MS project if the project has been exported to an MPP file.
- If new project tasks have been added to the Microsoft project structure, either new projects or new skill requests will be created in Alfabet. If a matching person is found for a resource for MS project tasks that are imported via an MPP file, the person will be associated with the relevant skill request and will also be specified for the capacity assignment definition for the relevant skill request. The creation of projects and skill requests will be determined by the MS project methodology.
- If project tasks have been deleted from the Microsoft project structure, the corresponding projects or skill requests will be deleted in Alfabet upon synchronization with the respective MPP file.
- Changes in project structure in Microsoft project can only be synchronized if the watermark information in the Microsoft project tasks is not changed. The watermark information is captured in the **ID Mapping Field** attribute in the MS project methodology definition.

To export a project to an MPP file:

1) In the toolbar, click **Action** > **Synchronize with MS Project File**. This functionality is only available if the project that you are working with has been exported to an MPP file.

2) In the **Synchronize Alfabet Project from MPP File** editor, define the following fields as needed:

   - **MS Project File to Synchronize:** Select the MPP file that has the MS project that you want to synchronize the selected Alfabet project with.
   - **MS Project Methodology to Synchronize with MS Project File**: Displays the MS project methodology that was used for the export of the project to the MPP file. Please note that the mapping information for projects, project tasks, and skill requests specified in the MS project methodology used for the export must be the same as the mapping information specified in the MS project methodology selected for the synchronization. For more information about the configuration of MS project methodologies, see the section Specifying MS Project Methodologies for Interoperability with Microsoft Project in the *Configuring Evaluation and Reference Data in Alfabet*.
   - **Description**: Displays the description of the MS project methodology selected in the **Methodology to Import MS Project File(s)** field.
   - **Default Skill**: Select a skill that the new skill requests shall reference if no skill is found based on the configured import matching rules specified in the MS project methodology.

3) Click **Synchronize** to trigger the synchronization. The project will be updated based on the changes in the MPP file.

Specifying REST API Connections for Events Triggering RESTful Service Calls

Via events, RESTful service calls to external RESTful services can be triggered automatically in the background when a user performs an action defined to execute the event, like for example entering, cancelling or exiting a wizard or workflow step.

The connection data for the call to the external RESTful service has to be specified in the ***GenericRestConfig*** XML object. The connection data has then to be referenced in a REST API connection object.

This configuration is part of a number of steps that are required to configure the event. The complete configuration is described in the chapter *Configuring Events* in the reference manual *Configuring Alfabet with Alfabet Expand*.

Do the following to create a REST API connection:

1) Go to the **Integration Solutions Configuration** functionality and click the **REST APIConnection** node in the **Integration Solutions Configuration** explorer.
2) In the view, click **New** > **Create REST API Connection**.

   If you have already defined a similar connection and want to take over the settings of that connection for your new connection, you can alternatively click **New** > **Create REST Connection as Copy** and select the existing connection the new connection should base on from the selector that opens. The editor for the new connection will then open with all settings identical to the copied connection and the name set to "Copy of <base connection name>".

3) In the **REST API Connection** editor, define the following fields as needed:

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each REST API connection. This number cannot be edited.
   - **Name**: Enter a unique name for the REST API connection.
   - **Release Status**: Select the REST API connection's current release status.

     The set of release statuses available for an object class are configured by your solution designer in the configuration tool Alfabet Expand. For more information, see the section *Configuring Release Status Definitions for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*. For general information about release statuses, see the section *Understanding Release Statuses* in the reference manual *Getting Started with Alfabet*.

   - **Description**: Enter a meaningful description that will clarify the purpose of the REST API connection.

   Authorized Access tab:

   - **Authorized User**: Click the **Search** icon to assign an authorized user to the selected Alfabet database connection. The authorized user will have Read/Write access permissions for the object and is responsible for the maintenance of the object.
   - **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

Connection tab:

- **REST Connection**: Select the connection to the relevant Alfabet database that is configured in the XML element ***Connection*** in the XML object ***GenericRestConfig*** in Alfabet Expand.

4) Click **Test REST API Connection**. If your settings are correct, a message "The connection is valid" is displayed. Otherwise an error message is displayed.

Specifying Analytics Dashboard Data Providers

Alfabet provides analytics dashboards that are designed based on the embedded third-party tool DevExpress® Dashboard. Analytics dashboards support end users to provide ad-hoc information-rich data visualizations that they create. Users can use the full range of visualization possibilities available in the DevExpress Dashboard Designer. One or more dashboard items such as charts, scatter charts, grids, cards, gauges, pivots, range maps, treemaps, etc. can be added to the analytics dashboard and filtering options such as combo-boxes, list boxes, and tree views can be leveraged. Furthermore, the user creating analytic dashboards can share them with other users and specify whether the dashboards are ReadOnly or editable by other users.

In order for end users to create analytics dashboards, you must specify analytics dashboard data providers that determine which configured reports or database views shall be used to fetch the data that will be displayed in the analytics dashboard. Each analytics dashboard data provider may have one or more configured reports assigned to it or one or more database views assigned to it. Users may then select one analytics dashboard data provider to define and design an analytics dashboard.

Before analytics dashboard data providers can be specified, the following must be configured:

- The configured reports or database views must first be configured by a solution designer in Alfabet Expand. Any tabular configured report or database view based on AQL- or native SQL-based queries can be used in this context. For example, if analytics dashboards are to report on technology use in the enterprise, the queries associated with an analytics dashboard data provider might be configured to return applications and local components and their indicators. The queries should be configured so that numerical data is returned rather than strings. For more information, see the sections *Configuring Reports* and *Creating Database Views To Enhance Performance And Support Search Functionalities* in the reference manual *Configuring Alfabet with Alfabet Expand*.
- The set of release statuses must be configured for the object classes **Analytics Dashboard Data Provider** (`AlfabetLocalDatasetsConnection`) and **Analytics Dashboard** (`DevExpressDashboard`) available for an object class are configured by your solution designer in the configuration tool Alfabet Expand. For more information, see the section *Configuring Release Status Definitions for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*.

The creation and design of analytics dashboards is carried out in the **Analytics Dashboards** functionality. For more information, see the chapter *Designing, Sharing, and Viewing Analytics Dashboards* in the reference manual *Getting Started with Alfabet*.

To create a analytics dashboard data provider.

1) Go to the **Integration Solutions Configuration** functionality and click the **Analytics Dashboard Data Provider** node in the **Integration Solutions Configuration** explorer.

2) In the view, click **New** > **Create Analytics Dashboard Data Provider**.

3) In the **Stereotype Selector**, select **Reports Data Provide** to create an analytics dashboard data provider object based on a configured report or select **Database Views Data Provide** to create an analytics dashboard data provide object based on a database view and click **OK**. The configured reports or database views that will be assigned to the analytics dashboard data provider must first be configured by your solution designer. For more information, see the sections *Configuring Reports* and *Creating Database Views To Enhance Performance And Support Search Functionalities* in the reference manual *Configuring Alfabet with Alfabet Expand*.

4) In the **Analytics Dashboard Data Provider** editor, define the following fields as needed:

   Basic Data tab:

   - **ID**: Alfabet assigns a unique identification number to each analytics dashboard data provider. This number cannot be edited.
   - **Name**: Enter a unique name for the analytics dashboard data provider. End users will select the analytics dashboard data provider when creating their analytics dashboards. Therefore, the name should in some way indicate
   - **Release Status**: Select the analytics dashboard data provider's current release status.

     The set of release statuses available for an object class are configured by your solution designer in the configuration tool Alfabet Expand. For more information, see the section *Configuring Release Status Definitions for Object Classes* in the reference manual *Configuring Alfabet with Alfabet Expand*. For general information about release statuses, see the section *Understanding Release Statuses* in the reference manual *Getting Started with Alfabet*.

   - **Description**: Enter a meaningful description that will clarify the purpose of the analytics dashboard data provider.

   Authorized Access tab:

   - **Authorized User**: Click the **Search** icon to assign an authorized user to the selected analytics dashboard data provider. The authorized user will have Read/Write access permissions for the analytics dashboard data provider and is responsible for the maintenance of the analytics dashboard data provider.
   - **Authorized User Groups**: Select one or more checkboxes to assign Read/Write access permissions to all users in the selected user group(s).

   Analytics Dashboard Data Provider tab:

   - **Permissible Configured Reports** / **Permissible Database Views**: Select one or more configured reports/database views in the drop-down list that shall be used to find the data to display in database views for the analytics dashboard data provider.
   - **Selected Configured Reports** / **Selected Database View**: Displays the configured reports/database views assigned to the analytics dashboard data provider.

5) Click **OK** to save your changes or click **Cancel** to exit without changing.

Appendix 1: Class Configuration for Object Classes

This table provides an overview of which object classes can be configured for various aspects in the **Class Configuration** functionality in the **Configuration** module. Each column in the table is explained:

Please note the following:

- **Role Types**: An X indicates that role types can be configured for the object class. For more information about the configuration of role types, see the section Configuring Role Types to Define Roles in the Responsibilities Page View in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.
- **Evaluations Types**: An X indicates that evaluation types (including indicator types and computation rules) can be configured for the object class. For more information about the configuration of evaluation types, see the section Configuring Evaluations, Prioritization Schemes, and Portfolios in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.
- **Portfolios**: An X indicates that portfolios can be configured for the object class. The information in the column indicates which standard portfolio views are associated with the object class. For more information about the configuration of portfolios, see the section Configuring Evaluations, Prioritization Schemes, and Portfolios in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.
- **Prioritization Scheme** s: An X indicates that prioritization schemes can be configured for the object class. The information in the column indicates which standard prioritization schemes are associated with the object class. For more information about the configuration of prioritization schemes, see the section Configuring Evaluations, Prioritization Schemes, and Portfolios in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.
- **Diagram View Items**: An X indicates that diagram view items can be configured for the object class. The information in the column indicates which standard diagrams are associated with the object class. For more information about the configuration of diagram view items, see the section Configuring Diagram Views for Diagrams in the reference manual *Configuring Evaluation and Reference Data in Alfabet*.

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
ALFA_QUESTIONARY	Survey			X		
Application	Application					
ApplicationGroup	Application Group	X	X		Aspect Portfolio view (APPG_ApplicationsPortfolio)	Information Flow Diagram view (APP_DiagramReport) Standard Application Diagram view (APP_StaticDiagram)
BFActivity	Activity		X			
BFCategory	Business Function Category	X				
BFDimension	Business Dimemsion Connection		X			
BFServiceActivity	Service Activity		X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
Brand	Brand	X	X			
Bucket	Bucket	X	X		Project Portfolio view (BKT_SubBucketsPortfolio)	
BusinessAppraisal	Business Appraisal		X			
BusinessData	Business Data	X	X			Information Flow Diagram view (APP_DiagramReport) Standard Application Diagram view (APP_StaticDiagram)
BusinessDataUsage	Business Data Usage	X	X			
BusinessFunction	Business Function	X	X		Application Portfolio view (BFN_ApplicationAssesmentPortfolio)	

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
BusinessObject	Business Object	X	X			Information Flow Diagram view (APP_DiagramReport) Standard Application Diagram view (APP_StaticDiagram)
BusinessObjectCategory	Business Object Category	X				
BusinessProcess	Business Process	X	X		Business Process Portfolio view (PROC_ProcessPortfolio) Application Portfolio view (BSP_ApplicationsPortfolio)	Business Processes Diagram view (PROC_BusinessProcessDiagram) Information Flow Diagram view (APP_DiagramReport) Standard Application Diagram view (APP_StaticDiagram)
BusinessProcessGroup	Business Process Group	X	X		Application Portfolio view (PROCG_BusinessProcessPortfolio)	Information Flow Diagram view (APP_DiagramReport)

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
BusinessProcess-Model	Business Process Model	X	X			Business Processes Diagram view (PROC_BusinessProcessDiagram)
BusinessProcessVariant	Business Process Variant	X	X			Business Processes Diagram view (PROC_BusinessProcessDiagram)
BusinessProcess-Version	Business Process Version	X	X			
BusinessRoleArch	Business Role Relationship		X			
BusinessService	Business Service	X	X			Information Flow Diagram view (APP_DiagramReport)
BusinessSupport	Operational Business Support	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
ComplianceControl	Compliance Control	X	X			
Component	Component	X	X			Information Flow Diagram view (APP_Connections_Diagram) Interface System Diagram view (COM_InterfaceSystemDiagram)
ComponentCatalog	Component Catalog	X	X			
ComponentCatalogElement	Component Catalog Element	X	X			
ComponentCategory	Component Category	X			Components Portfolio view (COMC_ComponentsPortfolio)	
ComponentGroup	Component Group	X	X		Aspect Portfolio view (COMG_ComponentsPortfolio)	

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
ComponentTest	Component Test	X	X			
Contract	Contract	X	X			
ContractDeliverable	Contract Deliverable	X	X			
ContractDependency	Contract Dependency					
ContractGroup	Contract Group	X	X		Contract Portfolio view (CNTRG_ContractsPortfolio)	
ContractItem	Contract Item	X	X			
ContractPayment	Contract Payment	X				
CostCentre	Cost Center	X				

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
CustomerSegment	Customer Segment	X	X			
Demand	Demand	X	X		Application Portfolio view (DEM_ApplicationsPortfolio) Similar Demands Portfolio view (DEM_SimilarDemandsPortfolio)	Information Flow Diagram view (APP_DiagramReport)
DemandGroup	Demand Group	X	X		Business Capability Portfolio view (DEMG_BusinessCapabilityPortfolio)	
Deployment	Deployment	X	X			
DeploymentElement	Deployment Element	X	X			
Device	Device					
DeviceCategory	Device Category					
DeviceGroup	Device Group	X	X			Information Flow Diagram view (DVC_DiagramReport)

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
						Static Device Diagram view (DVCG_Diagram)
Domain	Domain	X	X		Subordinate Domains and Business Functions Portfolio view (DOM_DomainPortfolio) Application Portfolio view (DOM_ApplicationsPortfolio) Components Portfolio view (DOM_ComponentsPortfolio)	Standard Domain Function Diagram view (DOM_FunctionStaticDiagram) Standard Domain Application Diagram view (DOM_APP_StaticDiagram) Information Flow Diagram view (DOM_InformationFlowDiagram)
DomainGroup	Domain Group	X	X		Application Portfolio view (DOMG_DomainPortfolio)	Information Flow Diagram view (APP_DiagramReport)
EnterpriseRelease	Enterprise Release	X	X			Standard Application Diagram view (APP_StaticDiagram) Information Flow Diagram view (APP_DiagramReport)
Feature	Feature	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
FrameworkGroup	Framework Group	X	X			Navigation Diagram view (FWG_StaticDiagram)
FunctionalModule	Functional Module	X	X			
GenericAffectedArch	Generic Architecture Connection		X			
ICTObject	ICT Object	X	X		Solution Building Block Portfolio view (ICTO_SBBPortfolio)	Standard Application Diagram view (APP_StaticDiagram)
ICTObjectCategory	ICT Object Category	X	X			
ICTObjectGroup	ICT Object Group	X	X			Standard Application Diagram view (APP_StaticDiagram)
InformationFlow	Information Flow	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
InterfaceSystem	Interface System	X	X			
Issue	Issue		X			
ITCapability	IT Capability	X	X			
ITMapView	Map View	X				
ITMasterPlan	Master Plan	X	X			
ITMasterPlanMap	Master Plan Map	X	X		Business Process Portfolio view (ITMPM_BusinessProcessPortfolio)	
ITPolicy	Policy	X	X			
ITPolicyGroup	Policy Group	X	X			
ITResource	Resource Request					

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
ITSolutionMap	Solution Map	X	X		Business Process Portfolio view (ITMPM_BusinessProcessPortfolio)	
ITStrategy	IT Strategy	X	X			
ITStrategyMap	IT Strategy Map	X	X		Business Process Portfolio view (ITMPM_BusinessProcessPortfolio)	
LegalOwnership	Legal Ownership	X	X			
LocalComponent	Local Component	X	X			
Location	Location	X	X			Location Diagram view (LOC_Diagram)
Market	Market	X	X			
MarketProduct	Market Product	X	X		Market Products Portfolio view (PROD_ProductsPortfolio)	Information Flow Diagram view (APP_DiagramReport)

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
					Application Portfolio view (BSP_ApplicationsPortfolio)	Standard Application Diagram view (APP_StaticDiagram)
MarketProduct-Group	Market Product Group	X	X		Market Products Portfolio view (PRODG_ProductsPortfolio)	Information Flow Diagram view (APP_DiagramReport) Standard Application Diagram view (APP_StaticDiagram)
MasterPlatform	Master Platform	X	X			
MasterPlatform-Category	Master Platform Category	X				
Migration	Migration	X	X			Migration Diagram view (MGR_Diagram)
MigrationGroup	Migration Group	X	X			
MigrationLink	Migration Rule	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
Milestone	Milestone		X			
Network	Network	X	X			
NetworkRoute	Network Route	X	X			
Operation	Business Function Operation	X	X			
OrganizationGroup	Organization Group	X	X			
OrgaUnit	Organization	X	X		ICT Object Portfolio view (ORG_ICTObjectsPortfolio) Application Portfolio view (BSP_ApplicationsPortfolio) Service Product Portfolio view (ORG_ServiceProductPortfolio)	Information Flow Diagram view (APP_DiagramReport) Standard Application Diagram view (APP_StaticDiagram)
Peripheral	Peripheral	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
PeripheralGroup	Peripheral Group	X	X			
Platform	Platform					
PlatformElement	Platform Element		X	X		
PlatformInformationFlow	Platform Information Flow		X			
PlatformLayer	Platform Layer	X				
PlatformTemplate	Platform Template	X				
PlatformTier	Platform Tier	X				
Project	Project	X	X		Application Portfolio view (PRJ_ApplicationsPortfolio) Solutions Portfolio view (PRJ_SolutionsPortfolio)	As-Is Architecture Diagram view (PRJ_AsIsArchitectureDiagram)

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
					Similar Projects Portfolio view (PRJ_SimilarProjectsPortfolio)	Information Flow Diagram view (APP_DiagramReport)
ProjectDependency	Project Dependency	X	X			
ProjectGroup	Project Group	X	X		Project Portfolio view (PRJG_SubGroupPortfolio)	
ReleaseItem	Enterprise Release Item	X	X			
RequestICTO	ICT Object Request	X	X			
Risk	Risk	X	X	X		
RiskGroup	Risk Management Group	X	X		Risk Objects Portfolio view (RISKGR_ObjectsPortfolio)	
RiskMitigation	Risk Mitigation	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
RiskMitigationCategory	Risk Mitigation Template Category					
RiskMitigationTemplate	Risk Mitigation Template					
RiskObject	Risk Object	X	X		Risks Portfolio view (RISKOBJ_RisksPortfolio)	
SalesChannel	Sales Channel	X	X			
Service	Technical Service					
ServiceContract	Service Contract					
ServiceOperation	Technical Service Operation	X	X			
ServiceProduct	Service Product	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
ServiceProduct-Group	Service Product Group	X	X		Service Product Portfolio view (SRVPRDG_ServiceProductPortfolio)	
Skill	Skill					
SLA	Service Level Agreement					
SolutionApplication	Solution Application	X	X			
SolutionBusinessData	Solution Business Data	X	X			
SolutionBusinessProcess	Solution Business Process	X	X			
SolutionBusinessProcessModel	Solution Business Process Model	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
SolutionBusinessService	Solution Business Service	X	X			
SolutionBusinessSupport	Solution Business Support	X	X			
SolutionComponent	Solution Component	X	X			
SolutionDevice	Solution Device	X	X			
SolutionDomain	Solution Domain	X	X			
SolutionDomainProject	Solution Domain Project	X	X			
SolutionFunctionalModule	Solution Functional Module	X	X			
SolutionInformationFlow	Solution Information Flow	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
SolutionLocalComponent	Solution Local Component	X	X			
SolutionPeripheral	Solution Peripheral	X	X			
SolutionPlatformElement	Solution Platform Element		X			
SolutionPlatformInformationFlow	Solution Platform Information Flow					
SolutionService	Solution Technical Service	X	X			
SolutionStandardPlatform	Solution Standard Platform	X	X			
Stack	Stack	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
StandardPlatform	Standard Platform	X	X			
StandardPlatformCategory	Standard Platform Category	X				
StrategicBusinessSupport	Strategic Business Support	X	X			
SystemBuildingBlock	Solution Building Block	X	X			
TacticalBusinessSupport	Tactical Business Support	X	X			
Technology	Technology	X	X			
TechnologyGroup	Technology Group	X	X			
Threat	Threat	X	X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
ThreatGroup	Threat Group	X	X			
ValueNode	Value Node	X	X		Business Capabilities Portfolio view (VMND_BusinessCapabilityPortfolioSvg)	
ValueNodeArch	Value Node Architecture Connection					
ValueStream	Value Stream	X	X			Value Stream Diagram view (VST_Diagram)
ValueStreamCondition	Value Stream Condition		X			
ValueStreamGroup	Value Stream Group	X	X			
ValueStreamStep	Value Stream Step		X			

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
ValueStreamValue	Value Stream Delivered Value		X			
Vendor	Vendor	X	X			
VendorGroup	Vendor Group	X	X		Vendors Portfolio view (VDRG_VendorsPortfolio)	
VendorProduct	Vendor Product	X	X			
VendorProductCategory	Vendor Product Category	X				
VendorProductGroup	Vendor Product Group	X				
VMMeasureCategory	Measure Type Category	X				
VMMeasureType	Measure Type	X				

Object Class Name	Object Class Caption	Role Types	Evaluation Types	Prioritization Schemes	Portfolios	Diagram Views
VMMeasure-TypeArch	Measure Type Architecture Connection		X			

Index

