# Release Notes 11.9.1 Platform Changes

The Alfabet platform has been updated to include many new capabilities and enhancements to the functional scope of Alfabet.

## Improvements

The following enhancements to the Alfabet platform are available with this release.

### Flexible configuration of text for cards

For indicator cards and grouped indicator cards, the caption and description of the evaluation type or indicator type can be overwritten with text specified in the card definition.

Card templates must be updated to apply this change. Customers using preconfigured Alfabet solutions will automatically have the new cards included in the meta-model configuration. Other customers must implement the new cards via the AMM file to upgrade cards delivered with the 11.9.1 CD.

End-user documentation is available about the *configuration of text for cards*. Documentation is also available for customers *adapting card templates in Alfabet Expand*.

### Easier navigation via breadcrumbs

Breadcrumbs have been changed to show when a view has been opened from an explorer or workflow. For example, if the workflow activity **Assess Demand** is selected in the **Workflows** explorer, the breadcrumb will display "Workflows/Assess Demand".

### Alignment of user and user profile types

The spelling of user profile types in the **User Profile** editor and user types in the **User** editor has been changed to **Read Only** and **Read/Write** to align with the spelling used in other places in the Alfabet user interface.

### Enhanced support of URL attributes

Links to external URLs has been enhanced:

- The object class `ALFA_URI` may be specified as a start class for workflows. This is a resolution for ticket ASD-7540.
- Administrative users can create extended attributes of the type `Url`.

## Search for Technopedia ID in Technopedia-based selectors

Customers interfacing with Technopedia® can search for assets via the **Technopedia ID** search field in the relevant Alfabet selectors. This is a resolution for ticket ASD-7466.

## Advanced selector specification

Selector definition has been considerably enhanced for object class properties of the type `Reference` or `ReferenceArray`:

- Selectors can be defined for object class properties both on the level of the object class model and class settings. Specification in the class setting overwrites specifications in the class model. If no selector is defined for the object class property, the selector defined in the class settings of the target object class is used.
- A new **Hide Selector** attribute is introduced both directly in the attributes of the object class property and in the property attributes in the class settings. The attribute is set to `False` per default. If the **Hide Selector** attribute is set to `True` , the **Advanced Search** link is hidden in edit fields where the reference could be set.
- A new **Selector Configuration** section is available in the attributes window to group the selector-relevant attributes.

The settings are described in the documentation on *the modification of object class properties*.

## Show usage extended to generic operations

The **Show Usage** functionality of wizards, workflow templates, and ADIF schemes in Alfabet Expand has been enhanced to show their usage in generic operations.

## Accessible information about shadow references

The new `IS_SHADOW` and `BACK_REFERENCE` attributes have been added to the `ALFA_MM_PROP_INFO` table. The attributes store the values of the **Shadow Reference** and **Back Reference** attributes of the object class properties in the current meta-model configuration. The information can be used to create reports that provide information about how object class properties shall be imported via ADIF or for any integration with third-party components, for example. If an object class property is a shadow reference, it must not be imported via the Alfabet RESTful services or via ADIF to this object class property but to the object class property of the reference target object class defined in the **Back Reference** attribute of the shadow reference property.

## Usability enhancements for Alfabet Expand

- The SQL editor that opens to define the **Calculation SQL Query** attribute of calculated properties has been changed. If the validity check upon closing the editor fails, the editor will not close and changes to the query will not be lost.
- The type `ReferenceArray` has been removed from the **Property Type** attribute of calculated properties because reference arrays cannot be calculated.
- The **Caption** attribute has been removed from the attributes of content items of the types `Indicator` and `Evaluation`. Captions are exclusively defined in the indicator card or evaluation card definition on the user interface.
- The **Content Item Preview** section has been removed from the attributes of content items of the type `Indicator`. An indicator card functions as a preview of indicators.

## Fixes

- The sequence of status values for **Status** or **Release Status** fields in data workbenches and content areas has been corrected to use the order specified in the `ReleaseStatusDef` XML object.
- The **Structure** button in graphic visualizations of data workbenches has been fixed to allow columns to be changed.
- Rendering issues in the **Change Object State** editor for applications have been fixed. These fixes are a resolution for tickets ASD-7567 and ASD-7904.
- The multi-edit editor of the **Information Flows** view ( `APP_InformationFlows`) correctly displays all object states. This fix is a resolution for ticket ASD-7608.
- Selectors for generic operations have been aligned and allow multi-selection for properties of the type `ReferenceArray`.
- The **Advanced Search** to select a responsible organization in a property group in the content area of an asset has been fixed.
- Business supports can be created for applications or ICT objects even if the applications/ICT objects do not have a start or end date defined.
- Handling of the active period has been corrected in the object lifecycle editor. After changing the start and/or end date, users can use the **Align Active Period** option to see the change directly in the active period information in the editor. The data is stored in the database when saving the changes in the editor.
- If a named user without a user profile assigned logs in via SAML, an error message will be displayed and the system will no longer be in an endless authentication loop. This fix is a resolution for ticket ASD-7622.
- Custom generic operations for detaching objects now work in the context of object class stereotypes.
- Changes have been made to the underlying functionality of the **Configured Reports** (SRCH_Reports_Only) view to avoid exceedingly long loading times. This fix is a resolution for ticket ASD-7090.
- Visibility conditions for editors and wizards have been fixed to work with all kind of editor fields. This fix is a resolution for ticket ASD-7914.
- Missing error messages and location titles have been re-established for the **Meta-Model** > **Check All Queries** functionality in Alfabet Expand. This fix is a resolution for ticket ASD-6806.

- Block processing of ADIF jobs has been fixed to work with all kind of ADIF jobs. To enhance performance, block processing is now set to enabled per default in all new ADIF jobs. This fix is a resolution for ticket ASD-7564.
- Files were not stored on ADIF export of data to file via a button in a configured report. This issue has been resolved. This fix is a resolution for ticket ASD-7313.
- The log file of the ADIF_Console.exe command line tool has been enhanced to show errors in ADIF execution during synchronous execution via event bus. This fix is a resolution for ticket ASD-7026.
- The `updateMM` endpoint of the Alfabet RESTful services has been fixed to accept the AMM file format. This fix is a resolution for ticket ASD-7487.

## Known Limitations

- If multiple rows are selected in a data workbench and the user scrolls while holding the SHIFT key to select the final row of the selection, the rows that are no longer visible on the screen while scrolling will not be selected.
- Reference array fields currently do not support adding or removing related objects for more than 300 items in inline editing drop-down menus. In addition, editing relations using the associated advanced search is limited by the maximum number of results that the selectors can return. To manage properties that require more than 300 related objects, it is recommended to add a data workbench as a content item to the relevant classes of a content area. The data workbench should be configured with the parent class as the base class allowing users to maintain relationships using the **Add** or **Remove** buttons in the data workbench.
- Stored filter values if a data workbench-based content item will be removed when the Alfabet user interface is rendered in a language other than the language the filter has been set in. This also changes the way a content item content is displayed if the data workbench-based content item is created with a filter setting.
- When the user interface is rendered in dark mode, icons in legends are not correctly colored.
- It is not possible to install multiple instances of the same version of Alfabet on the same host using the installer.

## Forthcoming Changes

- The logging mechanisms for Alfabet components configured in the server alias will be substituted with NLog logging with Alfabet 11.10. This includes logging to a file, an external log server, and to the Microsoft® event log.

# Release Notes 11.9.0 Platform Changes

The Alfabet platform has been updated to include many new capabilities and enhancements to the functional scope of Alfabet.

## New

The following is introduced to the Alfabet platform this release.

### User interface rendering in dark mode

Dark themes cause less strain to the eyes and are favorable in low light environments. With dark themes, highlighting can be enhanced and light colors are more prominent on a dark background. The user interface can be rendered in dark mode showing light text on a dark background. A default dark scheme is available for users after migration to Alfabet 11.9. In addition, administrative users and solution designers can decide to base newly created color themes on dark mode.

Dark themes are only recommended for customers using a preconfigured Alfabet 11.9 solution. Only colors and icons defined in the standard Alfabet configuration will automatically be adapted to the dark theme. Customers who have customized the solution with configured reports will need to adapt all color definitions in the report configurations to work with both dark and light themes. Depending on the number of configured reports, this may require extensive configuration.

*A description of the configuration requirements for dark themes is available in the documentation.*

### Navigation via keyboard

The Alfabet user interface has been enhanced to support keyboard navigation. This is an important step towards 508 compliance for the Alfabet 11.x user interface. Users need know only a small number of key combinations to perform the most common actions in the user interface:

- ALT + N: Move focus to left navigation pane
- ALT + M: Move focus to the masthead
- ALT + C: Move focus to the content area.
- TAB: Move focus forward to next item
- SHIFT + TAB: Move focus back to last item.
- Arrow keys. Move focus to another item.
- ENTER: Execute action on the selected item.

*Users can look up keyboard combinations in the documentation.*

## Automatically calculated properties

Calculated properties have been introduced as a new type of object class property to automatically calculate values for objects based on data already available in the Alfabet database. The calculated value is visualized as a read-only value and can be displayed in graphics, data workbenches, and content areas and provide the reliable display of up-to-date values without requiring users to click a button to trigger the calculation.

Calculated properties cover a higher range of use cases than calculated indicators. They can be used to calculate values that are not related to evaluations and can return multiple data types like dates and references. Customers can create a calculated property based on a native SQL query that defines a database view to store the calculated values. The database view calculates the values on demand at runtime. To ensure good performance for runtime calculation, calculated properties shall be based on simple queries only. The availability of calculated properties is adaptable per class setting.

*A description of how to define calculated properties* and *how to modify them in class settings* is available in the documentation for Alfabet Expand.

## Enhanced user support for multi-select fields

Tooltips are displayed for values in drop-down fields in data workbenches, content areas, and editors. Users can hover over the ⓘ symbol next to a value in the drop-down list and display a tooltip that provides information about the option for selection. The tooltips show existing hints that have been specified for protected and public enumeration items as well as descriptions specified in the **Description** field for objects like applications, ICT objects, etc.

## Role and indicator editable in property groups

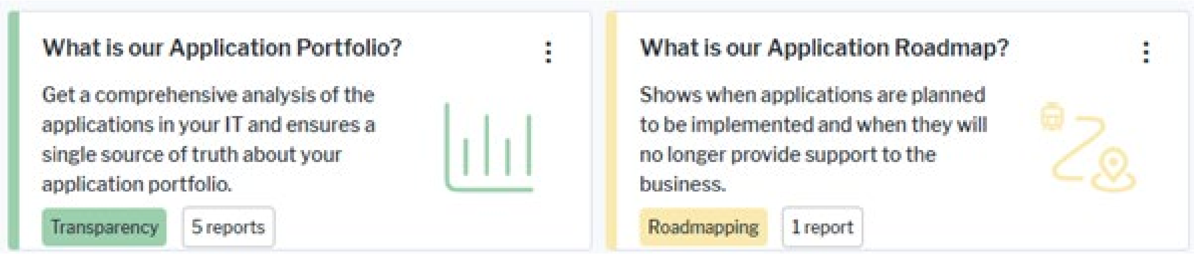
Roles and indicators can be set in property groups in class-based content areas. The indicator type or role type can be assigned to the property group in the new **Property Group** field in the standard editors for definition of indicator types and for assignment of a role type to an object class.

Some property group captions have been changed in the preconfigured Alfabet solution to enable unambiguous selection of property groups.

The description of the field has been added to the documentation of *role type configuration* and *indicator type configuration* and the positioning of the fields in the property group is described in the *Alfabet Expand documentation about property group assignment of object class properties*.

## Advanced placeholder definition for content items in content areas

Categorized visual items are introduced to define content items with placeholder text for the actual content. Categorized visual items hide the actual content behind a placeholder which can include a description, an image, information about the category the categorized visual item belongs to, and a tag with highlighted information. The category provides a mean to structure the content items according to use case. Users can see immediately which category the content item belongs to. Coloring can be used as an additional means to distinguish between categories.



What is our Application Portfolio?

Get a comprehensive analysis of the applications in your IT and ensures a single source of truth about your application portfolio.

Transparency 5 reports

What is our Application Roadmap?

Shows when applications are planned to be implemented and when they will no longer provide support to the business.

Roadmapping 1 report

Categorized visual items are defined independent of a specific content area and can be added to multiple content areas as content item.

Categorized visual items added to the content repository are by default added with the **Categorized Visual Items** item type. You can optionally define additional subtypes for this item type, which are then also selectable by the user to see only categorized visual items you assigned to the subtype.

*The definition of categorized visual items* has been added to the documentation of Alfabet Expand.

## Display integers as text via enumeration

Enumerations can be assigned to object class properties of the type `Integer` to display the integer values as text in the Alfabet user interface. For example, the user type for users is stored as an integer. While it was displayed as text in standard graphic views, it was displayed as integer in data workbenches generated for the object class `PERSON`.

A new enumeration type integer enumeration has been introduced. While enumeration items for string enumerations consist of a string value only, enumeration items for integer enumerations consist of both the integer the string represents and the string that shall be displayed for the integer. Users will see the string value on the user interface while the integer will be stored when the user selects the string value.

In data capture templates, either the integer value or the string value of the enumeration items can be returned.

*The definition of integer enumerations* has been added to the documentation of Alfabet Expand.

## Data workbenches for user administration

The following enhancements allow user administration in data workbenches:

- The button interactions to manage user authentication have been added to the meta-model as generic operation of the object class `PERSON` and will be available in any data workbench or content area of the object class. The buttons can be excluded from display in the class settings.
- Whether the user is a named user or an anonymous user will be displayed as string in any data workbench or content area generated for the object class `PERSON`.

The user type is stored as an integer and was displayed as integer in data workbenches for the object class `PERSON`. A new, private `PersonType` enumeration of the *new integer enumeration type* has been added to the Alfabet meta-model. The enumeration has been assigned to the `TYPE` object class property of the `PERSON` object class.

- Whether a user profile is read only or offers editing capabilities will be displayed as string in any data workbench or content area generated for the object class `ALFA_USERPROFILE`.

The user profile type is stored as an integer and was displayed as integer in data workbenches for the object class `ALFA_USERPROFILE`. A new, private `UserProfileType` enumeration of the *new integer enumeration type* has been added to the Alfabet meta-model. The enumeration has been assigned to the `TYPE` object class property of the `ALFA_USERPROFILE` object class.

## New standard generic operations

The standard generic operations have been added to the standard Alfabet meta-model:

- A generic operation to create interface systems has been added to the object class `InformationFlow`. The selected information flow will be automatically assigned to the interface system. Predecessor and successor are prefilled in the editor with the components the information flow is targeting.
- Generic operations to define the relation of applications to business data and business processes to business objects via business data usage have been added to the object class `BusinessDataUsage`.

- New generic operations have been added to information flows to provide definition of information flows for components and peripherals. This is a resolution for ticket ASD-6167.
- The button interactions to manage user authentication have been added to the meta-model as generic operation of the object class `PERSON` and will be available in any data workbench or content area of the object class. The buttons can be excluded from display in the class settings.
- The basic functionality for editing, deleting, and creating objects has been moved from object profile buttons to generic operations from the following object classes: `BusinessData`, `BusinessDataAttribute`, `BusinessDataUsage`, `BusinessObject`, `BusinessObjectCategory`, `BusinessSupport`, `Contract`, `ContractDeliverable`, `ContractDeliverableUsage`, `ContractDependency`, `ContractGroup`, `ContractItem`, `ContractPayment`, `DataRetentionPolicy`, `ProjectDependency`, `ProjectGroup`, `ValueNode`, `ValueArc`, `ValueNodeArch`, `VMMeasure`, `VMMeasureType`.

## Custom buttons for editing objects

Solution designers with access to Alfabet Expand can define generic operations for editing, creating, and deleting objects, adding objects to a parent object and detaching objects from a parent object. This capability enhances the possibilities for solution designers to provide button interactions for custom object classes or custom object class properties.

*The documentation of generic operation design* has been added to the documentation of Alfabet Expand.

## User management via SCIM integration

SCIM is a standard for cross-domain identity management and an important part of identification management in cloud environments. Alfabet 11.9 supports SCIM management with Okta® or AzureAD® as identity provider.

SCIM integration will push user data from the identity provider to the Alfabet database, updating the user information regularly. New users are added to the Alfabet database as anonymous users. The user profile assignment is updated based on identity provider group affiliation. A web-based SCIM management application is available for administration.

*SCIM management has been added to the documentation for system administrators.*

## Object class for AI Portfolio Management

A new `UseCase` object class has been added to the standard meta-model in preparation of the new AI Portfolio Management feature that will be published as part of the preconfigured Alfabet IT Transformation server solution in the near future.

## Improvements

The following enhancements to the Alfabet platform are available with this release.

### Standard folder for shared favorites

To make it easy for users to distinguish between own and shared favorites, a standard **Alfabet Shared Favorites** folder is now available for shared favorites. The folder is only visible if at least one favorite is shared. On migration to Alfabet 11.9.0, all already existing shared favorites will be moved automatically to this folder for all users.

*End users can read about this in the documentation about favorites.*

### Improved usability of search field in masthead

- When the user selects an option from the search results, the search field is no longer emptied but the selection made by the user will still be visible for reference.
- The search field is hidden if searchability has been disabled for all object classes. This is a resolution for ticket ASD-6401.

### Help button removed if no help is available

If no help link is defined in the server alias of the Alfabet Web Application the search button will be hidden completely instead of showing unavailability messages.

### Deleted content items removed from content pages

When a user deleted a content item in the **My Content Items** view, it was still persistent in content pages it has been added to and only removed when hidden. This behavior has been changed, and the content item is immediately removed from all content pages when deleted. This is a resolution for ticket ASD-498.

### Enhanced configuration and merging of content area defaults

In Alfabet 11.8, the way content areas were configured was complex, with different versions of content areas being stored on different level of configuration. The content of the content area was defined in Alfabet Expand while the layout was done in an administrative user profile in the Alfabet user interface. That configuration was then stored as default view. When a user opened the content area, the way the content area was rendered depended on three configurations. If the user had changed the configuration, the user's configuration was used. If the user did not change the configuration, the default configuration done by the administrative user was used. If neither a user configuration nor a default configuration was available, the configuration in Alfabet Expand was used.

This method also caused issues when merging the configurations. When a configuration was changed, the user was informed about it and needed to actively reload the default configuration, thus overwriting the user configuration and reverting the content page to default.

This approach is substituted with a new method with enhanced usability for both the solution designer and the end user. The content areas are easier to configure, and users experience an improved merging behavior between the configuration in Alfabet Expand and the configuration of the user.

- Solution designers can create pages, add content items, and arrange these content items in a content area designer accessible from Alfabet Expand. The content area designer is based on a private user profile which need to be assigned to a user with access to Alfabet Expand to access the content area designer. The configuration done by the solution designer in the web-based content area designer directly changes the configuration stored in Alfabet Expand
- If a solution designer changes an already existing and used content area either directly in Alfabet Expand or in the content area designer, all changes are directly merged to the individual user's configuration without reverting changes done by the user. New content items are added as last elements to the content page without removing content items added by the user or changing re-ordering of the existing pages performed by the user. New content items and new content pages are marked with a dot in the upper right corner the first time the user accesses the changed content area. If content items or content pages are removed in Alfabet Expand, they are also removed from the content areas changed by the users without affecting the rest of the user configuration.

On migration from Alfabet 11.8 to Alfabet 11.9, the default configuration for the content area will overwrite the configuration in Alfabet Expand. The default configuration is the last configuration effort done by the solution designer and is therefore regarded as the final version of the content area. Content available in Alfabet Expand but hidden from the default configuration will still persist in Alfabet Expand. A new **Default Visibility** attribute is introduced to inform the solution designer whether the content was hidden or visible in the Alfabet 11.8 default configuration. The attribute can be used any time to hide default content from the content area. The content will then be available via the content repository only.

The *documentation about creating content areas* as been adapted to the new methods and includes detailed information about the merging behavior for update of content areas already in use and about migration to Alfabet 11.9.

## Guided data views available in workflows and as part of content areas

The way users can access guided data views in the Alfabet user interface has been enhanced:

- Solution designers can configure in class settings whether the guided data entry view or the content area opens on navigation to an object. Users can then switch between the **Analysis View**, which is the content area, and the **Data Entry View**, which is the guided data entry view, via a toggle button. In addition, the **Edit** button can be hidden from the content area via an attribute in the class settings to provide the **Data Entry View** instead of the editor.
- Guided data entries can be added to a workflow step.
    - For workflow steps about an existing object, the guided data entry can be added as view to open. If a user shall be able to switch from the data entry view to a content area via a toggle button, the content area needs to be defined as additional target in the workflow step. The workflow step content area pages are available via tabs of the guided data view.
    - For workflow steps for creating a new object, the guided data view can be defined as additional target for the editor assigned to the workflow step. The user can then switch to the guided data view after having created the new object in the editor.
- On automatic refresh due to inline editing of a guided data view the user currently working with the view is scrolled back to the content item the user was working with prior to refresh.

*The documentation of guided data view design including the new methods for providing access to the guided data views* has been added to the documentation of Alfabet Expand.

## Content repository content per content page

For default content added to the content repository via the **System Content Repository** subnode of class settings solution designers can define whether the content shall be available in the content repository for all content areas, in a defined content area only, or for a defined content page in the specified content area only.

## Business functions can be added to content areas

Business functions can be added as content items to content areas. A preview icon or preview report must be defined for the business function content item as rendering of the complete business function in the content area is not supported.

The new option has been added to the documentation of *adding content items to content pages in a content area*.

## Enhancements for range definitions in drop-down lists

The drop-down lists for reference properties and string properties can be defined via a query. This method has been enhanced:

- For reference or reference array properties, the list of objects to be selected might be longer than the 300 objects which can be loaded into the drop-down list. Users can access all other objects via the **Advanced Search** button beneath the drop-down list. This opens the selector for the target object class. If a range has been defined, the selector is not restricted to the range definition in the field. Selectors are configured with own queries for content restriction. In the object class property overwrite in the class settings solution designers can now change the selector that opens to a selector which is better suitable to the configured restrictions. If users nevertheless select an object not included in the range definition in a selector, they are informed about the range mismatch.
- Different ranges can be defined per object or group of objects depending on the settings of another object class property. For example, the range of values for the importance of an application can be reduced to values for high importance if the application is a strategic application while for non-strategic applications users can specify either high, medium, or low importance.

The documentation about range definitions in Alfabet Expand has been adapted accordingly for *string ranges*and *reference ranges*.

## Improved usability of hierarchical data workbenches

The collapse and expand behavior of levels in hierarchical data workbenches has been improved:

- Hierarchical data workbenches only show an expand icon if the level contains sub-objects and can be expanded.
- If a user opens a panel, for example using the **Structure** button, after having expanded a level of the hierarchical data workbench, the expanded level stays expanded.

## Data capture in multiple languages

For on-prem customers having configured the Alfabet meta-model to provide object data translation capabilities, language specific capturing of object data has been re-implemented with enhanced usability in Alfabet 11.9. Users can capture the data directly in the language they have currently selected to render the user interface. The data will be included in the database table of the object class in the column for the respective language. If data already exists in the primary language, the value of the original value will be displayed in the respective field in data workbenches, content areas, and editors as a placeholder text which can optionally be overwritten with the language version by the user. When new objects are created, the primary key data or mandatory data specified for the object will be written to the database columns for both the primary language and the current language.

For the new way of managing translations, the XML attribute `EnableTranslationToPrimaryLanguage` in the **`SolutionOptions`** XML object is deprecated and not evaluated any longer. The **Capture Translations in Language of User Interface** is also not used any longer and has not been re-established.

Data Quality Rules and Data Triggered Actions will currently not work yet for multi-language data captures.

## Usability enhancements for Alfabet Expand

Handling of attribute setting in Alfabet Expand has been improved:

- Copy and paste of class operation adaptations from one class setting for an object class to another class setting for the same object class has been enhanced to enable copy and paste for single class operation modifications. This is a resolution for ticket ASD-6076.
- The drop-down list for selecting a graphic view target in a menu item of a user profile has been reduced to show only views which do not require a base object and can be added meaningfully to a left-hand navigation.
- When a button overwrite is created under the **Presentation** tab in the class settings, the **Caption** attribute is filled in automatically with the standard caption of the button. Thus, making it easy to identify the button if the technical name does not correspond to the button functionality. This is a resolution for ticket ASD-6009.
- When a new class setting was created, the **Editor Type** attribute was set to `undefined`. The default setting for an undefined editor type was `SimpleEditor`, which is one of the existing options to set the editor type. To make the default

behavior directly visible, the **Editor Type** for new class settings is now directly set to `SimpleEditor`. On upgrade to Alfabet 11.9.0, all `undefined` settings for the editor type in existing class settings will be changed to `SimpleEditor`.

- Marking of mandatory properties in the meta-model tab of Alfabet Expand has been adapted to the new concept in Alfabet 11 versions to regard only object class properties defined as unique key or which have a strict NULL handling as mandatory. All other checks whether data is correctly provided have been moved to data quality rules. Please note that this will also change the display of mandatory properties in newly created data capture templates. For existing data capture templates, the changes are not applied to ensure the integrity of data import and export. This is a resolution of ASD-5790.
- The **Review Selector** functionality has been re-implemented. This is a resolution for ticket ASD-6332.

## Restriction of role settings in data capture template

In data capture templates, the definition of roles can be limited to include users with a defined stereotype only. For example, you can exclude users with the stereotype `Contact` as role owner. To exclude a stereotype of the object class `PERSON` from the **Role Type** tab of a class-based data capture template, the new **Enable for Data Capture Templates** attribute in the class settings must be set to `False` for the class settings of that stereotype. Please note that the new attribute is not related to the **Enable for Data Capture Templates** attribute on the object class level. It is exclusively removing the object class stereotype from the **Role Type** tab of data capture templates of any object class while it is still possible to define a data capture template for the object class stereotype itself. Setting of this attribute in class settings of other object classes or stereotypes thereof has no effect.

This information has been added to the documentation of *class setting configuration* for Alfabet Expand.

## IT policy architecture import via data capture templates

In prior releases, it was not possible to import IT policy architecture objects of the type `IsAffected` via data capture templates. The `Type` object class property was completely managed by code. On import via data capture templates the type of the IT policy architecture objects was always set to the default `Implements` value. The `Type` object class property is now enabled for data capture templates with an in-build restriction to set the value to one of the two allowed values only.

## Enhanced button modification in class settings

- Custom buttons defined for content areas can be changed in class settings with the new **Modify Content Area Buttons** option for content area nodes under **Content Areas**. *This has been added to the documentation of the class settings.*
- The way button modifications change the rendering of the toolbar of graphic views has been improved. The check for button modifications is not limited any longer to modifications defined for the presentation object in the object class the view is defined for. The button definitions in the presentation object can include a class specification. If the button is not modified in the class setting of the base object class of the view, the class settings of the object class defined in the button definition is checked for modifications of the button interaction. In addition, presentation objects include configuration of object classes for which the standard editing buttons shall be added to the view. The generic operation overwrites in the class settings for the object class will also be checked for modifications. This is a resolution for ticket ASD-6878.
- Generic operations can be configured to display a prompt with a message that a user needs to confirm to activate the button operation. This is available for buttons for delete buttons, detach buttons, navigate buttons, buttons triggering an event, and buttons starting an ADIF job. A new **Confirm Message** attribute has been added to the attributes for customer defined generic operations to specify the message. *This has been included in the Alfabet Expand documentation for creating generic operations where applicable.*
- For buttons in presentation objects and for class operations, button overwrites in the class settings have two new attributes. A custom icon can be defined for the button via the **Icon** attribute. Whether the button is displayed in the three dots menu or directly in the toolbar can be changed via the **Button Location** attribute. This has been included in the Alfabet Expand documentation *for button modifications*.
- If stereotypes are defined for object classes, the menu option for creating the object should display the stereotype instead of the object class name. In previous releases, this required a button adaptation in all class settings for all stereotypes. To reduce configuration effort, in Alfabet 11.9, the caption of the private generic operations for creating an object is overwritten automatically with the name of the object class stereotype if only one stereotype can be created. If multiple stereotypes could be created, the caption is not overwritten, but a stereotype selector opens.

- Information about the modification of generic operations and buttons in presentation objects can be exported to CSV file format via the new **Get Class Buttons Report** option of the **Class Settings** node.

*This is documented in the class settings documentation.*

## Sort order and caption overwrites for property groups

The sort order of object class properties in object class property groups as well as the caption for the object class property group can be overwritten in a new **Property Groups** node in the class settings. While in a property group content item added to a content area only attributes added by default to the property group are available for sorting, sorting in the **Property Groups** node in the class settings takes object class property modifications defined in the class settings into account. *This information has been added to the documentation of object class property group configuration.*

## Configurable access to fields in private editors

Under the **Editor** subnode of class settings solution designers can now adapt the availability and editability of fields in private editors for a user profile.

## Enumeration style definition for configured reports

For configured reports displaying a gantt chart or a tabular dataset, enumeration styling can be defined via the new Alfabet instruction `ShowEnumeration(ColumnName, EnumerationName, icon:bool, color:bool);`. If `icon:bool` is set to `icon:true` then the icon defined in the enumeration is shown. If `color:bool` is defined as`color:true` then the coloring defined for the enumeration is taken over to display the value. Styling is done via style groups if defined in the enumeration.

## Restricted access to attributes of data quality management

The **Access in User Interface** attribute has been set to `ReadOnly` for several object class properties of the `DataQualityRule` object class. the properties are calculated by code and shall not be set by users.

## Deactivation of generic API Integration interfaces via server variables

An XML object specified with server variables was only valid if all server variables defined in the XML object were set in the server alias of the Alfabet Web Application. For XML objects including multiple integration connections, like the XML object **`GenericAPIIntegrationConfig`**, this led to error messages about missing server variables for all integration solutions in case the values for server variables for one integration solution were not set. This behavior has been changed to enable activation of a sub-set of configured integration solutions only. The server variables need still to be available for all integrations to work, but no value need to be specified for them. If no values are specified for the server variables used in one integration solution specification, all other integration solution specifications will still work and the integration solution with missing values will show error messages about data not being available instead of informing end users about server variable issues.

## Remote Alias Configurations Removed from Alfabet Administrator

If an existing alfabetms.xml configuration file is re-used from an Alfabet 10.15.x installation, remote server alias configurations existing in the file will not be available any longer via the explorer in the Alfabet Administrator.

## Control of number of database server connections

Alfabet 11.9 uses SQL server connection pooling for connections to the database server. With connection pooling, opening and closing a connection is no longer required for each database request, but all requests are handed over to a pool of persistent connections. Two new fields have been added to the **Database Settings** tab of the server alias configuration to change the minimum and maximum number of concurrent connections allowed in the connection pool. The default values provide optimum connection pool usage in most environments.

## Direct migration to Alfabet 11.9 from Alfabet 10.15.x and previous Alfabet 11 versions

Customers can upgrade to Alfabet 11.9 directly from one of the previous versions without any intermediate upgrades: Alfabet 10.15.x, Alfabet 11.7.2, and Alfabet 11.8.x. Please note however that customers migrating from Alfabet 11.7.0 or 11.7.1 must follow the stepwise migration within the Alfabet 11.7 versions first. *Details are given in the installation documentation.*

## AMM editor adapted to new functionalities

The AMM editor has been changed to remove deprecated content and provide a better usability for customers.

- The **Guide Pages** tab has been removed from the AMM file editor because guide pages are no longer supported.
- The default value for the **Include Configuration Name and Version** checkbox has been changed from `true` to `false`.
- User interface themes can be added to AMM files in the **Reference Data** tab and merged into the user interface theme configuration. When a user interface theme is added to an AMM file, all styles which are changed in the user interface

theme are also added to the AMM file. When the configuration in the AMM file is merged in the configuration of the target database, all styles and themes which do not exist in the target database will be added and styles and themes also available in the target database will be overwritten. Styles are overwritten on the level of the style. If a style group is available in both databases and has different styles, only the styles which are available in the AMM file will be overwritten.

- Categorized visual items are included in AMM files via the **Save Configuration** option. They can also be selected via the **Find Meta-Model Objects for Deployment** functionality.
- Content area default configurations are no longer used and have been removed from the content that can be added to AMM files.

## Reschedule of locked jobs via command line

Rescheduling of locked jobs that were not executed because they ran into a deadlock is available via the AlfaAdministratorConsole.exe command line tool. *Details are available in the documentation.*

## RESTful service client credential calls

Client credential call can be used for calls to the endpoint `token` of the Alfabet RESTful services. The body of the request must then be defined as: `grant_type=client_credentials&client_id=youralfabetRESTusername&client_secret=youralfabetRESTuserAPIpassword`. This is a resolution for ticket ASD-6159.

*The description of the call has been added to the documentation.*

## Fixes

- Performance problems have been resolved:
    - The load time for portfolio charts, matrices, and gantt charts has been reduced.
    - Loading of content area items has been improved to speed up rendering of content areas.
- The error which occurred in global search on attempt to view all search results or filter the results has been fixed.This fix is a resolution for ticket ASD-7032 and ASD-7326.
- After a favorite has been assigned to a favorites folder, it was not possible to change its location to the root level of the **Favorites** menu. The editor has been corrected to enable detaching of favorites from favorites folders.
- Email addresses of users are correctly displayed in data workbenches. This fix is a resolution for ticket ASD-6828.
- In data workbenches, the information about the number of returned objects has been corrected for date filters and shows the correct returned number of objects. This fix is a resolution for ticket ASD-6853.
- Rearrangement of columns in data workbenches via the **Structure** button has been fixed and filter values are maintained after restructuring. This fix is a resolution for ticket ASD-6587.
- Content areas are no longer double refreshed when matrix reports are embedded as content items. This fix is a resolution for ticket ASD-6863.
- User created content pages are deleted if they are hidden by the user prior to adding them to the content repository. The names of these deleted content pages are no longer stored in the background and users can later create a new page with the same name. This fix is a resolution for ticket ASD-7241.
- String array properties are now stored when set via inline editing. This fix is a resolution for ticket ASD-6402.
- The sort order in drop-down lists has been fixed to follow the sort order defined in the enumeration or in the range query defined for the object class property. This fix is a resolution for ticket ASD-6785.
- The missing operation to detach components from applications has been added. The button is visible in data workbenches for assigning components to applications. While the assignment of a component to an application creates a

local component object in the background, the detach operation will delete the local component. The component itself is not deleted but detached by this activity. This fix is a resolution for ticket ASD-6919.

- The standard functionality to block out calendar days which is available in Alfabet 10.15.x has been re-established for Alfabet 11. It is now possible to create, edit and delete blocked out dates one day at a time.
- The object class icon for standard platforms is shown correctly in map reports. This fix is a resolution for ticket ASD-6283.
- The rendering issue in the assignment editor has been fixed. This fix is a resolution for ticket ASD-6949.
- In explorers, the option to open a view in a new tab has been deactivated for explorer nodes with no view behind them. This fix is a resolution for ticket ASD-7157.
- The legend of bar charts has been corrected to show only coloring of bars which represent values other than zero and are therefore displayed in the chart. The coloring is consistent in bar chart and legend. This fix is a resolution for ticket ASD-7163.
- Rendering of doughnut charts in content areas has been fixed to show the total number within the doughnut chart reliably in all doughnut charts it is implemented for. This fix is a resolution for ticket ASD-7222.
- Issues with data quality rules have been resolved:
  - Issues with rescan of data quality rules have been corrected.
  - The execution of the `RescanDataQualityRules` ADIF job will no longer lock the database during generation of temporary tables. The asynchronous execution will no longer prevent users from working with Alfabet when rescanning complex data quality rules.
  - Export of the **My Data Quality Issues** view to Microsoft Excel® file has been fixed. This fix is a resolution for ticket ASD-6711.
  - The limit of maximum 50 objects in bar charts derived from content areas is no longer applied to the data quality bar chart displayed in the **Data Quality** visualization of the data workbench. The limit is only applied to user generated visualizations.
- The issue with filtering of the **Organizations** view (`ORGG_Organizations`) has been fixed. This fix is a resolution for ticket ASD-6163.

- Filters for diagram reports were displayed in the **Global Filters** filter section of the diagram filters and the **Page Filters** section of content areas including the diagram report. This has been corrected, and filters are shown in the **Report Filters** section where they belong. This will change the filter behavior on the content page containing the diagram, as these filter fields are no longer available on the page level.
- The issue with the search for providing applications in the **Business Architecture** content page of business capabilities has been fixed. This fix is a resolution for ticket ASD-6559.
- In the data capture editor, generic attributes with a range definition are added correctly to the **Extended Attributes** tab. This fix is a resolution for ticket ASD-6041.
- The editor for creating workflow releated job schedules has been fixed to allow all types of actions to be selected. This fix is a resolution for ticket ASD-7234.
- Registering of user login and logout for the Alfabet components has been corrected to enhance the security with better GUID Id generation mechanisms and to fix the problems with wrong user information in audit tables. This fix is a resolution for ticket ASD-7027.
- The Technopedia® product selector was fixed to show complete captions for all filter fields. This fix is a resolution for ticket ASD-6782.
- Information about the Technopedia® ID has been added to all selectors used to import objects from Technopedia.
- Breadcrumbs shown in Alfabet have been enhanced to process browser back button interactions. If a user is going back to the previous view via the browser back button, the previous view is no longer added again to the breadcrumbs, but the breadcrumbs are reduced to end with the view the user has gone back to. This fix is a resolution for ticket ASD-7156.
- Opening of the workflow explorer from a link in a workflow notification email was fixed to set the focus on the correct workflow step in the explorer. This fix is a resolution for ticket ASD-7121.
- The way changes are saved in Alfabet Expand has been corrected to speed up the saving process and avoid database connection outages for other Alfabet components. Instead of saving the complete meta-model each time, only the changed part of the configuration is saved. This fix is a resolution for tickets ASD-6871 and ASD-6955.

- The copy and paste functionality for the menu items in user profile configuration has been adjusted to accept the new option to create a three-level menu. This fix is a resolution for ticket ASD-6083.
- Skype® has been deprecated. The `SkypeId` and `SkypeDomain` object class properties have been removed from the object class `Person` and from all relevant **User** editors.
- In the class settings, the object selection for the **Modify Button** functionality of the **Presentation Object** node and all functionalities to add items to the **System Content Repository** has been corrected to show only items for selection which have not been selected previously. This fix is a resolution for ticket ASD-6074.
- The **Access in User Interface** attribute of object class properties related to Technopedia® integration has been set to `ReadAccess` to enable display of the information on the Alfabet user interface.
- The way **Edit** buttons are handled in content areas has been fixed:
  - If a custom editor is defined in the class settings of the object class, the custom editor opens as part of the editor.
  - Buttons defined directly in the content area are available in the content area.

This fix is a resolution for ticket ASD-6307.

- The new XML attribute `MaxLayoutCalculationDuration` has been added to the XML object **SolutionOptions**. The default value is 30 seconds. If rendering of a view requires more than the set layout calculation duration, rendering of the report is cancelled and the user is informed that the view cannot be rendered. The user is also informed that the problem can be fixed by reducing the number of objects to be displayed if filters are available. The maximum layout calculation is not applied to publications.
- The user interface style group **Error Level** has been corrected to also include border color and text color definitions.
- Drop-down fields for object selection in custom editors and wizards for which a lookahead query has been defined are corrected to show only values corresponding to the query results. This fix is a resolution for ticket ASD-6771.
- If the default stereotype set for an object class is not matching the defined stereotypes, it is nevertheless set for the object class if the user creating the object does not explicitly choose a stereotype. To avoid issues with wrong stereotype settings, the private meta-model has been scanned for correctness

and cleaned from offending default stereotype settings. In addition, a script will be run during update of the meta-model to the Alfabet 11.9 release which scans the customer defined settings in the meta-model. If a default value is incorrect, it will be removed, and a message is written about the removal in the log file generated during update of the meta-model. This fix is a resolution for ticket ASD-6520.

- If a default stereotype is set for an object class, the default will be added automatically to the stereotype selection fields in the user interface and users only need to select a stereotype if the stereotype shall deviate from the default. This fix is a resolution for ticket ASD-6963.
- Visibility conditions defined for content items in a data workbench are not only applied to the visibility of content items in the content pages, but also to content items in the content repository. This fix is a resolution for ticket ASD-6288.
- If a content item with the same content has been added to two content pages of the same content area, these content items can now be deleted independend of each other. If one content item is deleted, the content item showing the same content on the other content page is persistent. Addition of the same content item to the same content page twice has been disabled. This fix is a resolution for ticket ASD-6860.
- If an object profile is assigned to a class-based content area and an edit operation is also available in the generic operations of the object class, all edit buttons from the object profil were hidden and only the generic edit operation was displayed as button. This behavior has been adapted to hide only edit buttons for standard editing of the object. Buttons for editing specific attributes, like changing the object state of the object, are still added to the class-based content area. These buttons can be hidden via the class settings, if they should not be included. This fix is a resolution for ticket ASD-6172.
- The rendering of custom explorer root nodes has been fixed to display the view assigned to the root node. In addition, content areas can now also be added to custom explorer root nodes as views of the type `ObjectView`.
- The issue with the advanced mode of Alfabet Expand activating not only button creation for configured reports, but also for Alfabet standard graphic views has been resolved. The standard graphic views are private objects that must not be customized.
- The behavior of the Technopedia® selector opened from a configured report based on the `Capture_Components` template has been adapted to the behavior

of the selector in standard views. If the report is embedded in the content area of an ICT object, component category or vendor linked to Technopedia, the relevant data about this object is prefilled in the selector and cannot be changed by the user.

- During update of the meta-model with a customer configuration including ADIF scheme update failed if ADIF schemes existed in the AMM file and the target database with names only differing in case sensitivity. The update meta-model mechanism has been changed to identify ADIF scheme names case insensitive. ADIF schemes will be overwritten when the name of the ADIF scheme in the target database is the same as the name of the ADIF scheme in the AMM file except for the use of capital and non-capital letters. As ADIF scheme handling has been made case insensitive, it is no longer possible to create two ADIF schemes with names only differing in case sensitivity in one database. This fix is a resolution for ticket ASD-6624.
- Restore of databases from ADBZ file failed if the database contained exceedingly large tables, for example for audit information. To resolve this issue, the timeout for index generation for database tables has been made configurable. The timeout is now read from the **Database Settings** > **Command Details** > **Default Execution Timeout** setting of the server alias the restore is performed with. This setting also defines the timeout for any other server related processes.
- The link to the help center defined in the **Overview** > **Help Server** field of the server alias can be defined with or without a slash at the end.
- IT-Pedia® integration failed if server variables for other integration solutions were missing in the server alias of the Alfabet Web Application. The way server variables are evaluated for IT-Pedia has been corrected to exclude checks for other integration solutions.
- Restore of audit data has been fixed for database restore using the AlfabetAdministratorConsole.exe. Audit data is taken over from the source to the target database if `–IncludeAudit true` is added to the command line. This fix is a resolution for ticket ASD-7131.
- When the definition of workflow names in the command line of a job to start workflows with AlfaAdministratorConsole.exe is not correct, an error is returned, and the job will not start any workflows instead of starting all available workflow which are configured to start automatically.
- The **API Permissions** tab of the user editor is no longer deactivated when ADIF is not licensed by the customer. An ADIF license is no longer required to use the

Alfabet RESTful services, and the tab is required to set the RESTful service access permissions for users.

- Upload of documents to the internal document store of the Alfabet database via the endpoint `idocupload` of the Alfabet RESTful services has been corrected. If documents already exist, they can be updated. This fix is a resolution for ticket ASD-7560.
- If the saml.json file is missing in the SAML configuration, user access is blocked instead of opening the standard SAML login screen which offers insufficient authentication security.

## Known Limitations

- On import of the license file for upgrade to Alfabet 11.9.0 to the server alias editor, a message will pop up requesting a decision about workspace rescan. Customers using the preconfigured Alfabet IT Transformation Server (FastLane or Enterprise) databases must click **Yes**. Customers with a complete customized solution must click **No**.

For customers using the preconfigured Alfabet IT Transformation server, XML objects will show preconfigured versions of the XML configuration as workspace nodes. The workspaces have been introduced to provide preconfigured XML objects without overwriting existing customer configuration.

- Customers migrating from Alfabet 10.15.x to Alfabet 11 must substitute some standard business functions with content areas, as the underlying technology required for these standard business functions is no longer supported:
  - The **Personal Info** (`Personal_Items`) functionality that was available in Alfabet 10.15.x versions to inform users about their assignments to tasks and user groups is no longer supported and has been removed from the meta-model. It can be substituted by a class-based content area for a single object of the object class `PERSON`.
  - Business functions which display subordinate business functions as tiles are no longer supported. They can be substituted with a content area containing the subordinate business functions as content items.
- It is not possible to change the coloring of elements in Archimate® diagrams. The standard coloring corresponds to the coloring defined by Archimate.