



# ARIS - Alfabet Interoperability Interface

Alfabet Reference Manual

---

Documentation Version Alfabet 10.15.2

Copyright © 2013 - 2023 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and or/its affiliates and/or their licensors.

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.





The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

---

## Conventions used in the documentation

Convention	Meaning
<b>Bold</b>	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 <b>Finish</b> when setup is completed.
<i>Italics</i>	Used for emphasis, titles of chapters and manuals.  this  Example: see the <i>Administration</i> reference manual.
Initial Capitals	Used for attribute or property values.  Example: The object state <b>Active</b> 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

<b>Chapter 1: Introduction</b>	<b>5</b>
Related Documents	7
Documents Relevant for the Alfabet Application	7
Documents Relevant for the ARIS Application	8
Terminology in ARIS and Alfabet	8
<b>Chapter 2: Functionality of the ARIS - Alfabet Interoperability Interface</b>	<b>10</b>
Integrating ARIS Business Process Model Planning Data Into Alfabet	11
Creating an Alfabet Business Process Model on Basis of an ARIS Business Process Model	12
Integrating Changes To an ARIS Business Process Model into the Existing Dependent Alfabet Business Process Model	13
Navigation From the Alfabet to the ARIS User Interface	16
Defining a New Link to an ARIS Diagram	17
Opening a Link to an ARIS Diagram	17
Updating the Information About ARIS Diagrams Displayed in the Table	17
Integrating Links to ARIS Diagrams into Dashboards and Configured Reports	17
<b>Chapter 3: Activating and Configuring the ARIS - Alfabet Interoperability Interface</b>	<b>19</b>
Required Licenses	20
Configuring the Alfabet Web Application to Enable REST API calls	20
Optionally Configuring the Alfabet Web Application to Accept Self-Signed Certificates	21
Configuring the ARIS Connect Server to Read Data from Alfabet	21
Required Configuration on Alfabet Side	21
Required Configuration on ARIS Side	23
Configuring the Alfabet Web Application to Read Data from ARIS	23
Step 1: Configuring the Connection to ARIS Connect in the XML Object ArisApiConfig	25
Step 2: Configuring the Connection to the ARIS Database in the Integration Solutions Configuration Functionality	32
Making the Views for the ARIS - Alfabet Interoperability Interface Available on the Alfabet User Interface	34
<b>Chapter 4: Customer Configured Integration of Alfabet Data to ARIS</b>	<b>36</b>
Configuring Data Export from Alfabet	36
Configuring Data Import to ARIS	36
<b>Chapter 5: Customer Configured Integration of ARIS Data to Alfabet</b>	<b>38</b>
Configuration of the ARIS - Alfabet Interoperability Interface to Perform Customized Data Import	39
Configuration of the ADIF Import Scheme	43
Creating an ADIF Import Scheme With the ARISImport_Assistant	44
Temporary Tables Created During Import	45
Considerations for Data Import	53
<b>Index</b>	<b>55</b>

## Chapter 1: Introduction

Software AG provides two applications that support IT management and business process management:

- ARIS for business process analysis and management
- Alfabet for Enterprise Architecture Management, IT Planning and IT Portfolio Management

Together, these applications provide Software AG's customers with comprehensive and consistent planning and management support from the highest level of business assessment to the individual activities in IT planning and portfolio management.

Both ARIS and Alfabet maintain data about business process models and objects in the IT landscape in order to support planning and management capabilities. From the perspective of business process modelling, however, each tool provides unique functionalities. ARIS provides a wealth of views and methodologies to plan, develop, and enhance business processes on a detailed level. Entities in the IT landscape such as applications are included as modeling elements in order to map functions in the business process model to the application assets.

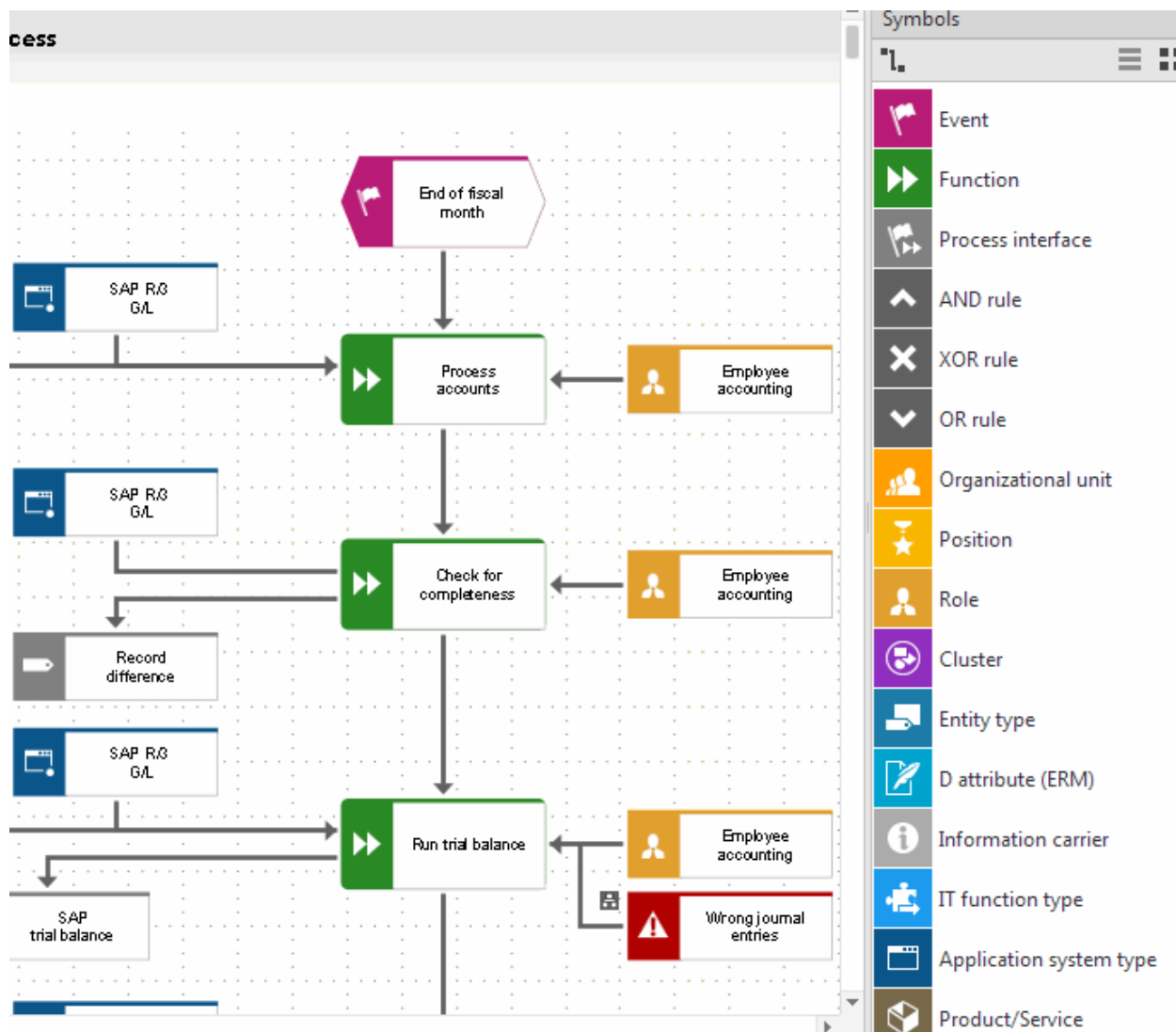
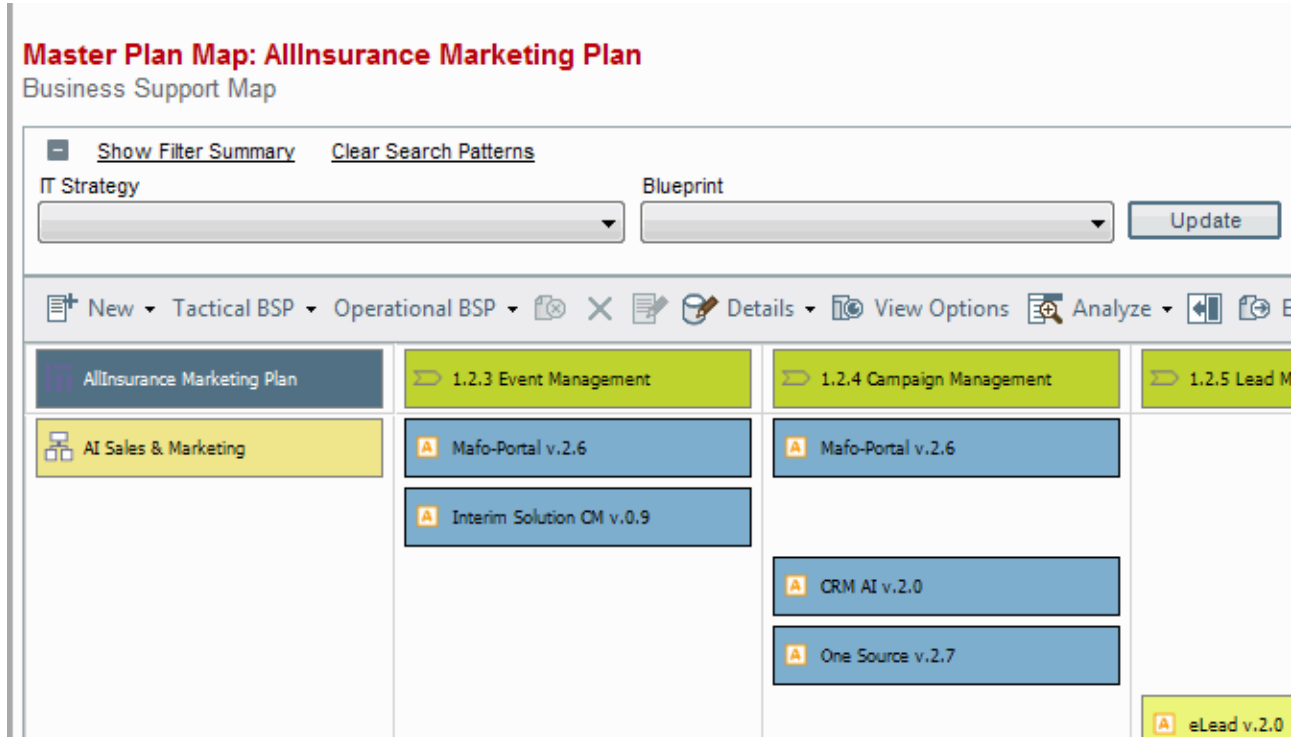


FIGURE: ARIS EPC diagram of business process including applications (blue) used to execute functions (green)

Alfabet focuses on modeling the IT landscape that supports the execution of business processes. This includes, for example, release and deployment planning as well as IT-centric risk and compliance management. The business process model is used to map the applications - existing today or potentially existing some time in the future - to the relevant business processes in order to align IT support with business demands and to plan the development and streamlining of the IT landscape in strategy and master planning.



*FIGURE: Master Planning in Alfabet with supporting applications (blue) displayed per organization (yellow) and business process (green)*

The available interoperability between the two solutions allows both ARIS and Alfabet to be used in synergy. Data maintained in either ARIS or Alfabet can be re-used in the other application to ease data maintenance and to prevent data inconsistencies between the complementary management tools. The ARIS - Alfabet Interoperability Interface allows for the regular synchronization of data between ARIS and Alfabet. The default configuration of the data synchronization is based on the following:

- Business process models are created and modelled in ARIS and the information about the business process model is transferred to the Alfabet database to be considered in enterprise architecture management.
- The enterprise architecture is managed in Alfabet. Only information about the IT landscape that is relevant for business process planning is transferred to the ARIS databases. In the standard configuration, information about applications as the central objects in the IT infrastructure is transferred to ARIS.

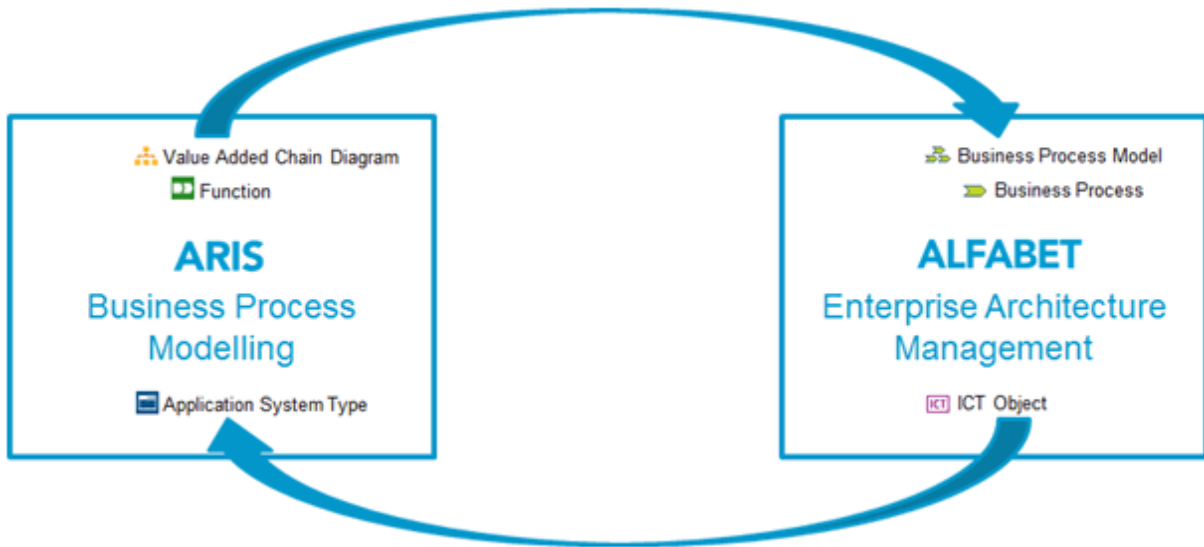


FIGURE: Standard Data exchange configured for the ARIS - Alfabet Interoperability Interface

The ARIS - Alfabet Interoperability Interface enables the user to easily traverse the user interfaces of ARIS and Alfabet. For example, when a user plans a business process model in ARIS, thereby leveraging application assets; the full 360° of the relevant application portfolios is just one click away.

## Related Documents

The following documents are relevant for the functionality or configuration of the ARIS - Alfabet Interoperability Interface and provide more information about the technologies and models used:

### Documents Relevant for the Alfabet Application

Consult:	For more information about:	Relevant for RESTful API based ARIS - Alfabet Interoperability Interface	Relevant for Web Service based ARIS - Alfabet Interoperability Interface
Reference Manual <i>Alfabet Data Integration Framework</i>	Configuring data import into and export from the Alfabet database with the Alfabet Data Integration Framework (ADIF).	Yes	Yes
Reference Manual <i>Web Services for Alfabet</i>	Working with the Alfabet Web Services. The document provides information about the structure of the WSDL and authentication for Web services.	-	Yes

Consult:	For more information about:	Relevant for RESTful API based ARIS - Alfabet Interoperability Interface	Relevant for Web Service based ARIS - Alfabet Interoperability Interface
Alfabet Online Help	Working with business process models and solution business process models in Alfabet.	Yes	Yes

## Documents Relevant for the ARIS Application

The documents relevant for the ARIS - Alfabet Interoperability Interface are part of the ARIS documentation for software administration that is located in the folder *Administration* of the ARIS documentation.

## Terminology in ARIS and Alfabet

In order to better understand this document, it is important to know about the differences regarding the terminology used in ARIS and Alfabet because different terms are sometimes used to describe similar concepts or the same term may be used differently in each application.

ARIS	Definition	Alfabet
Object Type	A classification for the object stored in the database. In Alfabet, each object class has a separate database table. In ARIS, model type information is stored as attribute of the object in the database table for model objects.	Object Class
Attribute	Information stored about an object in the database (for example, Name and Type). In ARIS, model type attributes are defined centrally and assigned to one or multiple model types. In Alfabet, each object class has an individual set of properties. Object class properties relevant for multiple object classes are configured for each object class individually and changes made to an object class property configured for one object class (for example, changing the Name of a property) are not automatically applied to other object classes.	Object Class Property
Fact Sheet	Central view for an object to view and edit information stored about the object. The view also provides links to other relevant views.  In Alfabet, the object view/object profile is the central workspace for an object and highly customizable. Object profiles can either present information about the object and a list of links to the relevant views only (object view) or also directly display the content of graphic reports about the object (object cockpits).	Object View / Object Profile / Object Cockpit



ARIS	Definition	Alfabet
Business Process Model	In Alfabet, a business process model describes a hierarchy of business processes in the enterprise. In ARIS, a business process model describes in detail the way business processes are executed in the enterprise. A model can have multiple hierarchical levels that start with a general view on the business process and provides navigation to detailed diagrams that allow the execution of functions involved in a business process to be planned.	Business Process Model
Function	A set of activities that represent work required to achieve a business objective. In ARIS this is a <i>function</i> within a business process model, while in Alfabet this is a business process.	Business Process
Application System Type	An application is a fully-functional integrated IT product that provides functionality to end users and/or to other applications. In ARIS business process models applications are included as <i>application system type</i> . In Alfabet, the object class Application represents applications.	Application

## Chapter 2: Functionality of the ARIS - Alfabet Interoperability Interface

The following functionality is provided by the standard RESTful service based ARIS - Alfabet Interoperability Interface:

- [Integrating ARIS Business Process Model Planning Data Into Alfabet:](#)

Alfabet business process models can be created on basis of business process models already designed in ARIS. If the base business process model in ARIS is later changed, Alfabet users responsible for maintaining the business process model data in Alfabet can view a report about the changes and apply them to the Alfabet business process model. These processes are triggered by a user on the Alfabet user interface.

The way Alfabet business process models are derived from the ARIS business process model data is configurable. Configuration is described in the section [Activating and Configuring the ARIS - Alfabet Interoperability Interface](#).

In addition to the standard integration of ARIS business process model data as business process models into Alfabet, data from ARIS business process models, other object classes of the Alfabet meta-model can be targeted by data integration from ARIS business process models. For example, capability maps or business supports can be based on planning performed in ARIS. Unlike the standard integration targeting business process models these integrations are not based on a concept and standard user interface based integration mechanisms provided by Alfabet. A highly configurable data import mechanisms based on the Alfabet Data Integration Framework (ADIF) allows customer specific integration logics to be implemented on demand. For more information, see the chapter [Customer Configured Integration of ARIS Data to Alfabet](#).

- *Integrating Alfabet enterprise architecture management data into ARIS :*

Applications are a central element of enterprise architecture management in Alfabet and correspond to the ARIS *application system type*. The information about the applications maintained in Alfabet can be taken over to ARIS either in a batch process or on a per application basis. This integration is done via functionality available on the ARIS Architect user interface. Integration includes setting of a link to the relevant Alfabet object on the Alfabet user interface. Once set, the links are also available in ARIS Connect.

The implementation of the ARIS - Alfabet Interoperability Interface and the handling of the per object and batch data integration is described in the Reference Manual *ARIS Server Installation and Administration Guide* that is part of the ARIS documentation.

- *Navigation between the user interfaces of ARIS and Alfabet :*

The navigation between the user interfaces of Alfabet and ARIS is based on links in one of the applications that open the other application on the linked target view. These links are set differently in Alfabet and ARIS:

- In Alfabet a solution designer can configure which object classes allow setting of links to the ARIS user interface and which ARIS diagram types can be targeted by the links. On basis of the configuration, users can set links to ARIS diagrams as needed on the page view **ARIS Diagrams** of any object of the object classes configured to allow link setting. Navigation to the ARIS user interface requires a login to the user interface prior to showing the linked view.

Once a link has been defined on the **ARIS Diagrams** page view, it can also be included into Dashboard diagrams to open the ARIS user interface directly from the Dashboard and it can be used in

configured reports including navigation directly to the ARIS diagram opening in the ARIS user interface in a new browser tab.

For information about the setting of links and navigation to the ARIS user interface, see [Navigation From the Alfabet to the ARIS User Interface](#) in this reference manual.

For information about the required configuration of Alfabet to enable the navigation feature, see [Activating and Configuring the ARIS - Alfabet Interoperability Interface](#) in this reference manual.

- In ARIS a link to the Alfabet user interface is set for each object of the *application system type* that is based on an Alfabet object of the object class application. The link is available in the attributes of the ARIS object. The links are also available as links in the ARIS Publisher if activated in the settings for the publisher export profile used.

Navigation requires login to the Alfabet user interface. Please note that Read/Write access to the object is only granted if the user has the required access permissions to the object in Alfabet. Otherwise the object data is ReadOnly. Mandates are not supported. That means that a user accessing the Alfabet user interface via a link from ARIS can see the object regardless of the mandate settings.

The implementation of the ARIS - Alfabet Interoperability Interface and the handling of the navigation to the Alfabet is described in the Reference Manual *ARIS Server Installation and Administration Guide* that is part of the ARIS documentation.

The following information is available:

- [Integrating ARIS Business Process Model Planning Data Into Alfabet](#)
- [Creating an Alfabet Business Process Model on Basis of an ARIS Business Process Model](#)
- [Integrating Changes To an ARIS Business Process Model into the Existing Dependent Alfabet Business Process Model](#)
  - [Directly Applying Changes to the Alfabet Business Process Model](#)
  - [Applying Changes Via Solution Business Process Planning](#)
- [Navigation From the Alfabet to the ARIS User Interface](#)
  - [Defining a New Link to an ARIS Diagram](#)
  - [Opening a Link to an ARIS Diagram](#)
  - [Updating the Information About ARIS Diagrams Displayed in the Table](#)
  - [Integrating Links to ARIS Diagrams into Dashboards and Configured Reports](#)

## Integrating ARIS Business Process Model Planning Data Into Alfabet

Alfabet business process models can be created on basis of business process models already designed in ARIS. If the base business process model in ARIS is later changed, Alfabet users responsible for maintaining the business process model data in Alfabet can view a report about the changes and apply them to the Alfabet business process model.

A business process model in Alfabet can be build automatically as copy of an existing business process model in ARIS. The process starts with a selectable ARIS business process model diagram and creates a business process hierarchy according to the configuration of the ARIS - Alfabet Interoperability Interface.

By default mapping is done as follows:

- For each *function* in the diagram a root business process is created regardless of any hierarchy within the diagram.
- If a *function* in the diagram is connected to other diagrams of the same type as the start diagram, the *functions* in the connected diagrams are added as child business processes to the business process representing the *function*. This is done for the complete hierarchy of *functions* and diagrams.
- For all imported *functions* a link to the diagrams that the object is assigned to is added automatically to the **ARIS Diagrams** page view of the business process in Alfabet.

Optionally, the ARIS - Alfabet Interoperability Interface can be configured to perform mapping based on connections between objects in ARIS:

- A hierarchy of business processes is build from the *functions* in the diagram according to the connections defined between the *functions*. Which type of connection is relevant for building the hierarchy is configurable in the ARIS - Alfabet Interoperability Interface. Optionally, your company may have configured the ARIS - Alfabet Interoperability Interface to import other ARIS object types as business processes.
- The process then follows the hierarchy of relevant ARIS connections of the objects in the diagram to other diagrams of the same type or of any other type configured as relevant in the ARIS - Alfabet Interoperability Interface and includes all connected objects in these diagrams to the hierarchy.
- For all imported *functions* a link to the diagrams that the object is assigned to is added to the **ARIS Diagrams** page view of the business process in Alfabet.




If the Alfabet business process model is build on basis of connections defined between objects in ARIS. Prior to starting import of an ARIS business process model make sure that connections are consistently set in the involved ARIS business process diagrams.

For all business processes defined during the integration process the user triggering the integration process is set as the authorized user of the business process. It is recommended to prevent modification of the business process model by Alfabet users to ensure that the business process model is kept identical to the ARIS source. For information about configuring access permission to objects, see *Configuring Access Permissions for Alfabet* in the reference manual *Configuring Alfabet with Alfabet Expand*.

## Creating an Alfabet Business Process Model on Basis of an ARIS Business Process Model

To create an Alfabet business process model on basis of an existing ARIS business process model, you must first create a business process model in the business processes explorer in Alfabet and then link the Alfabet business process model to the ARIS business process model:

- 1) Click the **Business Processes** node  at the top of the **Business Processes** explorer.
- 2) In the toolbar, click **New > Create New Business Process Model**. The **Business Process Model** editor opens.
- 3) Enter information into each field, as required.

### Basic Data tab:

- **ID:** Alfabet assigns a unique identification number to each object in the inventory. This number cannot be edited.

- **Name:** Enter a name for the business process model.
- **Description:** Enter a meaningful description that will clarify the purpose of the business process model.

**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 business process model.
  - 5) Click the new business process model in the explorer. The object profile of the business process model opens.



Depending on the configuration of your Alfabet solution, an object cockpit might open. You can change to the object profile via the link on the top left of the object cockpit.

- 6) In the object profile, click on **Business Processes**. The **Business Processes** page view opens.
- 7) In the toolbar, click **New > Get Business Process Model From ARIS....** An object selector opens.
- 8) In the **ARIS Database Connection** field, select the ARIS database that you want to connect to for import of business processes.



The current release status of the ARIS diagram connection is displayed in parentheses behind the name of the ARIS diagram connection. The release status informs you whether the connection is ready for use in a production environment or For example, in a test phase.

- 9) Enter search criteria into the **Search Pattern** field to find the relevant ARIS business process diagram and click the **Search** button.
- 10) Select the ARIS diagram from the list of search results and click **OK** to start the integration process. Business processes are created automatically from ARIS *functions* and structured according to relations defined in ARIS.

## Integrating Changes To an ARIS Business Process Model into the Existing Dependent Alfabet Business Process Model

A business process model diagram that is based on an ARIS business process diagram remains coupled with the ARIS parent diagram and if the ARIS diagram changes, these changes can be taken over to the Alfabet business process model. The Alfabet business process model derived from an ARIS business process model can be updated anytime with changes made to the ARIS business process model since creation of the dependent Alfabet business process model or since the last update of changes.

Update is performed on the **Business Process Model ARIS Source Report** page view of the business process model.

To open the view:

- 1) Click the new business process model in the **Business Processes** explorer. The object profile of the business process model opens.



Depending on the configuration of your Alfabet solution, an object cockpit might open. You can change to the object profile via the link on the top left of the object cockpit.

- 2) In the object profile, click on **Business Process Model ARIS Source Report**. The **Business Process Model ARIS Source Report** page view opens.

The **Business Process Model ARIS Source Report** page view lists all changes that were performed in ARIS on the relevant business process diagrams.

Each column in the table is defined below:

- **Info:** An icon informing about the complexity of the change. The icon can have the following colors:
  - **green:** The change is not critical. For example, a new function has been added to the ARIS diagram that would result in adding a new business process to the business process model in Alfabet, or the name or description of an existing business process has changed.
  - **yellow:** The change is not critical, but more complex. For example, the structure of the business process model has changed and existing business process within the existing business process model will be moved to a new parent process during change integration.
  - **red:** The change is critical. For example, a function has been removed from the ARIS diagram and the respective business process will be deleted from the Alfabet database during change integration.
- **Description:** A short description about which change was made to the ARIS diagram.

On basis of the information in the list you can decide which of the available import mechanisms to use:

- [Directly Applying Changes to the Alfabet Business Process Model:](#)

This mechanism directly changes the existing business process model in Alfabet. If a business process already exists, the changes in ARIS overwrite existing data in Alfabet but the business process is not completely overwritten. That means that a change in the description of the business process will overwrite the current description without affecting the other properties of the business process like the name or the relations defined to other objects in the Alfabet database. Business processes that are deleted in the ARIS business process model are deleted in the Alfabet business process model and depending sub-objects and relations defined in Alfabet are also deleted.
- [Applying Changes Via Solution Business Process Planning:](#)

This mechanism creates a solution business process model for the selected business process model including all the changes performed in ARIS. Within the solution business process model, required changes concerning Alfabet specific data can be performed. The current business process model is substituted with the solution business process model after data was adjusted. This is For example, relevant if a business process is deleted. The relations to Alfabet object that have been set for the business process can be redirected to other business processes in solution business process planning. Direct integration would delete the existing relations without notice.

Import is always performed on batch. It is not possible to select individual changes for import.

## Directly Applying Changes to the Alfabet Business Process Model

To directly integrate all changes that were made to the ARIS source business process model:

- 1) In the toolbar, click **New > Apply Changes**. All changes are applied automatically to the existing business process model. The report is empty after the action is performed.

## Applying Changes Via Solution Business Process Planning

Complex changes performed in ARIS should be applied via the business process planning functionality in Alfabet.



In business process planning, one or multiple copies of a business process model can be created as solution business process models. All business processes in the original business process model exist as solution business processes in the solution business process model.

At the start, each solution business process has the same property values as the business process it is based on. The information about which business process it is based on is stored in an additional property. The solution business processes in the solution business process model can then be edited, deleted and moved within the business process model while the original business process remains unchanged. New solution business processes (which are not based on an existing business process) can also be created within the business process model. A number of reports and planning tools are available about the potential impact to the IT landscape by the solution business process model. The person planning the business process model changes can then view the impact to the existing landscape without modifying an existing business process model.

Multiple solution business process models can be defined and compared to in order to select the best solution. Once a solution business process model has been approved, it can be checked in to the database. The original business process model will be overwritten with the solution business process model. A solution business process that is based on an existing business process overwrites that business process. Solution business processes that are new in the solution business process model are converted to new business process objects. Business processes that do not have a corresponding solution business process upon check in will be removed from the business process model and marked as no longer in use. The planned changes are now updated to the Alfabet business process model and the actual business process objects.

A new solution business process model is created for the business process model transferred from ARIS to Alfabet. If a solution business process model already exists for the business process model and the already existing business process model is not yet checked in, it will be overwritten by the newly transferred business process model. An Alfabet user has to check in the solution business process model manually via the Alfabet user interface in order to integrate the changes to the current Alfabet business process models.

Data integration is based on the ARIS GUID, which is transferred with the data about each object. The GUID is stored in the Alfabet database as the property `ARIS_GUID` for the object classes business process, business process model, solution business process and solution business process model. The import of a business process model includes the following changes to objects:

- 1) A solution business process model is created for the business process model with the `ARIS_GUID` identical to the GUID of the transferred business process model.



If a solution business process model from a prior import has not been checked in yet, this business process model will be overwritten with the currently imported data.

- 2) For all ARIS *functions* included in the transferred business process model, a solution business process with the `ARIS_GUID` set to the transferred GUID is created within the solution business process model.

- 3) For each solution business process within the solution business process model, a check is made to ensure that a business process with the same `ARIS_GUID` exists in the Alfabet database. If such a business process is found, all properties of the existing business process except for the Name and Description properties are copied to the new solution business process and the solution business process is assigned to overwrite the existing business process during check-in of the solution.
- 4) The business process model that will be overwritten by the solution business process model is then checked to see if there are any business processes that exist in the business process model but not in the solution business process model. In this case, the business process was removed from the business process model in ARIS. For each of these obsolete business processes, a solution business process will be created with a property **Deleted** set to True. When the business process model is overwritten by the solution business process model during check-in, the corresponding business process will be deleted.

After import, the user responsible for the maintenance of the business process model can consult reports available for the solution business process model to decide whether links from business processes to other objects in the Alfabet database require corrections prior to check in of the solution business process model. When the user has performed all required changes to the Alfabet -specific part of the business process properties, the solution business process model can be checked in manually by the responsible user. The changes performed in ARIS are mirrored in the Alfabet business process model structure only after the solution business process model has been checked-in.

To apply all changes that were made to the ARIS source business process model via a solution business process model:

- 1) In the toolbar, click **New > Create Business Process Solution...** A solution business process model is created for the current business process model with all changes applied to it. The report is empty after the action is performed.
- 2) Perform the required corrections to the solution business process model and check it in to substitute the current business process model. For more information about business process planning see *Business Process Planning Functionality*.

## Navigation From the Alfabet to the ARIS User Interface

Alfabet users can set links to ARIS diagrams as needed on the page view **ARIS Diagrams** of any object of the object classes configured to allow link setting in the configuration of the ARIS - Alfabet Interoperability Interface. Navigation to the ARIS user interface requires a login to the user interface prior to showing the linked view.

For each link to an ARIS diagram the following information is displayed on the page view **ARIS Diagrams**:

- **ARIS Diagram Name:** The name defined for the diagram in ARIS.
- **ARIS Diagram Description:** The description defined for the diagram in ARIS.
- **ARIS Diagram Type:** The ARIS diagram type of the diagram in ARIS.
- **Is Source:** This column shows a checkmark if the diagram has been used to create the content of the Alfabet object owning it.
- **ARIS DB Connection:** The name of the ARIS database connection used to access the ARIS database. Alfabet can connect to multiple ARIS databases concurrently. A diagram is therefore specified by a combination of the connection required to connect to the database and the name of the diagram.



## Defining a New Link to an ARIS Diagram

To create a new link to an ARIS diagram:

- 1) In the toolbar, click **Add Aris diagrams...**. An object selector opens.
- 2) In the **ARIS Database Connection** field, select the ARIS database containing the diagram.



The current release status of the ARIS diagram connection is displayed in parentheses behind the name of the ARIS diagram connection. The release status informs you whether the connection is ready for use in a production environment or For example, in a test phase.

- 3) Select a diagram. You can use the **Search Pattern** field to limit the display of search results to a subset of the available diagrams.




Which ARIS diagram types are displayed in the selector depends on the configuration of the ARIS - Alfabet Interoperability Interface.

- 4) Click **OK** to create the link. The information about the diagram is added to the table.

Once a link has been defined on the **ARIS Diagrams** page view, it can also be included into Dashboard diagrams to open the ARIS user interface directly from the Dashboard.

## Opening a Link to an ARIS Diagram

To open a link to an ARIS diagram:

- 1) Select the diagram in the table.
- 2) In the toolbar, click the **Navigate**  button.

The ARIS user interface opens in a new tab of your browser. After login, the selected business process diagram is directly displayed. Which view about the diagram is displayed is configured by your solution designer in the ARIS - Alfabet Interoperability Interface.

## Updating the Information About ARIS Diagrams Displayed in the Table

During business process modelling in ARIS, diagrams for that a link was created in Alfabet may be removed from the business process model. Also, the name and the description attribute of the ARIS business process model diagrams may be changed in ARIS after the link was set. To update the information on the **ARIS Diagrams** page view with the current information about the diagram in ARIS:

- 1) In the toolbar, click **Refresh Links** .

## Integrating Links to ARIS Diagrams into Dashboards and Configured Reports

Once a link to an ARIS diagram has been created, it can be used in dashboard diagrams and configured reports to enable navigation from Alfabet dashboards and configured reports to the ARIS user interface. The ARIS diagram opens in a new browser tab when accessed from dashboards or configured reports.

In Alfabet dashboard diagrams, the ARIS diagram link is available as toolbox item in the toolbox. Clicking on the toolbox item and then clicking into the diagram opens a selector for available ARIS diagram links. For detailed information about the design of dashboard diagrams see the reference manual *Designing IT Landscape Diagrams in Alfabet*.

In configured reports, ARIS diagram links can be added by adding the object class `ArisDiagramLink` to the query of the configured report. Creating a link to an ARIS diagram results in the creation of an object of the object class `ArisDiagramLink` in the Alfabet database. For detailed information about the design of configured reports and the rules for definition of queries, see the reference manual *Configuring Alfabet with Alfabet Expand*.

## Chapter 3: Activating and Configuring the ARIS - Alfabet Interoperability Interface

An API based on RESTful service architecture, in the following called RESTful API, is implemented in both the Alfabet Web Application and ARIS Connect. To perform data integration, both Alfabet and ARIS must be configured to connect to the RESTful API of one another and exchange data that is then automatically processed when a user performs an action involving data exchange on the respective user interface.



For example, a user working with business processes in Alfabet can open a view **ARIS Diagrams** to configure a link to a relevant ARIS diagram. This functionality is only working if a connection to the ARIS RESTful API is configured for Alfabet.

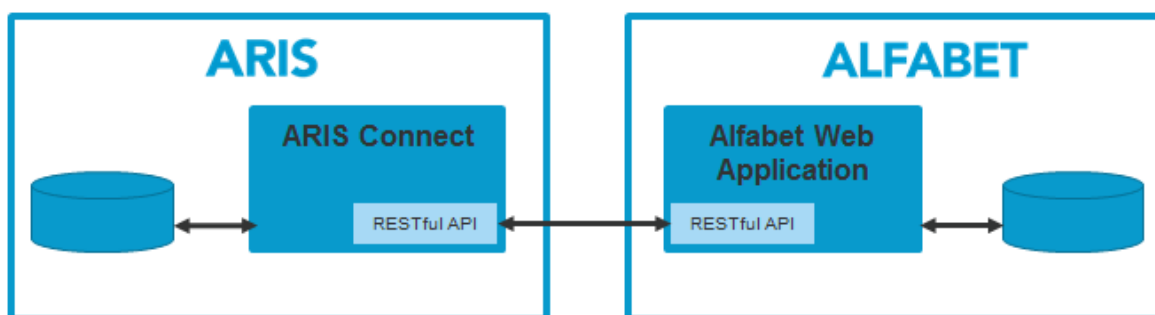


FIGURE: Overview of the components involved in data exchange via the ARIS - Alfabet Interoperability Interface

Alfabet can be configured to connect to multiple ARIS databases located via connections to one or multiple ARIS Connect servers. These individual database connections are the basis for each user interaction. The user must select the database that he/she wants to import data from prior to performing the action related with data import. If a user should have access to only a subset of the available data connections, the mandate concept in Alfabet can be used to hide database connections for the user.

The configuration of the ARIS application is described in the *Administration* folder of the ARIS documentation. This documentation is limited to the required configuration of the Alfabet application and a basic description of the required configuration of authorization of the ARIS RESTful services when connecting to the Alfabet RESTful API. Detailed information about the implementation and configuration on ARIS side is provided by the ARIS documentation.

The following configuration is required:

- [Required Licenses](#)
- [Configuring the Alfabet Web Application to Enable REST API calls](#)
- [Optionally Configuring the Alfabet Web Application to Accept Self-Signed Certificates](#)
- [Configuring the ARIS Connect Server to Read Data from Alfabet](#)
  - [Required Configuration on Alfabet Side](#)
  - [Required Configuration on ARIS Side](#)
- [Configuring the Alfabet Web Application to Read Data from ARIS](#)
  - [Step 1: Configuring the Connection to ARIS Connect in the XML Object ArisApiConfig](#)
    - [Configuring Access to the ARIS Connect Server](#)

- [Configuring the Mapping of Data for Business Process Model Integration](#)
- [Configuring the Navigation to the ARIS User Interface](#)
- [Configuring Connection via a Proxy Server](#)
- [Step 2: Configuring the Connection to the ARIS Database in the Integration Solutions Configuration Functionality](#)
- [Making the Views for the ARIS - Alfabet Interoperability Interface Available on the Alfabet User Interface](#)

## Required Licenses

A valid license for the ARIS - Alfabet Interoperability Interface need to be implemented on both Alfabet and ARIS side to use the ARIS - Alfabet Interoperability Interface.

The license requirements for ARIS depend on the release version used. Consult your Software AG sales representative for information about license requirements.

For Alfabet, a license for the Alfabet Data Integration Framework (ADIF) is also required.

## Configuring the Alfabet Web Application to Enable REST API calls

The Alfabet API for RESTful Web services must be enabled in the implementation of the Alfabet Web Application. This includes settings in the `web.config` file of the Alfabet Web Application as well as settings in the Microsoft® Internet Information Services® hosting the Alfabet Web Application. In most Alfabet installations, the RESTful services will already be activated for other purposes. The description of the required activation in this reference manual is therefore limited to the required permission settings. If the Alfabet API for RESTful Web services is not yet activated in your Alfabet installation, see *Activating the Alfabet RESTful API on Server Side* in the reference manual *Alfabet RESTful API* for information about the required configuration steps.

The Alfabet API for RESTful Web services must be enabled in the server alias of the Alfabet Web Application. The configuration is performed in the configuration tool Alfabet Administrator:



For information about how to access and use the Alfabet Administrator, see the section *Working with the Alfabet Administrator* in the reference manual *System Administration*.


- 1) In the explorer of the Alfabet Administrator, click the **Alfabet Aliases** node.
- 2) In the table, click the server alias that you want to configure.
- 3) In the toolbar, click the **Edit**  button. An editor opens.
- 4) Go to the **Server Settings > REST API** tab.
- 5) Set the checkmark for the **Enable REST API v2** attribute.
- 6) In the **API Access Options** field, set the checkmark for the following options:
  - **Has Meta-Model Access**
  - **Has GetObjectsByReport Access**

- **Has GetObjectsByRefs Access**
  - **Has GetObjectsByFilter Access**
- 7) Click **OK** to save your changes.

## Optionally Configuring the Alfabet Web Application to Accept Self-Signed Certificates

Self-signed certificate validation can optionally be used on HTTPS connections sending data from ARIS to Alfabet.

This requires the following configuration:

- 1) Copy the self-signed certificates from the third-party web service to a local folder that the Alfabet Web Application has access permissions to.
- 2) Open the Alfabet Administrator.
- 3) In the explorer, click **Alfabet Aliases**. The right pane displays a list of all available alias configurations.
- 4) In the table, select the alias configuration of the Alfabet Web Application.
- 5) In the toolbar, click the **Edit**  button. You will see the editor in which you can edit the alias configuration.
- 6) Open the **Server Settings > Security** tab.
- 7) Enter the path to the folder containing the self-signed certificates in the field **Path for Self-Signed Public Certificate Files**.
- 8) Click **OK** to save your changes.

## Configuring the ARIS Connect Server to Read Data from Alfabet

The authorization of the ARIS Connect Server to the RESTful API of the Alfabet Web Application is based on a single authorized Alfabet user. In Alfabet, a user must be configured to have the required access permissions. The ARIS database is then configured to use this user for authorization calls to the Alfabet RESTful services:

- [Required Configuration on Alfabet Side](#)
- [Required Configuration on ARIS Side](#)

### Required Configuration on Alfabet Side

Access to the Alfabet RESTful API must be explicitly granted to a named Alfabet user in the user configuration in the Alfabet database. User configuration is performed in the **User Administration** functionality in the user interface that is accessible via an administrative user profile or the **User Management** functionality that is available via the connected server alias node in the Alfabet Administrator.



For more information about how to create a named Alfabet user, see the chapter *Defining and Managing Users* in the reference manual *User and Solution Administration* or *Managing New and Existing Users* in the reference manual *System Administration*.

The named user must match the following conditions:


- The user profile assigned to the user must grant access to the private configured reports **Data Import to ARIS: Application Groups, Data Import to ARIS: Applications, Data Import to ARIS: Business Processes, Data Import to ARIS: Domains, Data Import to ARIS: Value Notes**. Access is granted by default for new user profiles.



For more information about user profile configuration, see *Configuring User Profiles for the User Community* in the reference manual *Configuring Alfabet with Alfabet Expand*. For more information about access permissions to configured reports, see *Making Configured Reports Available to a User Profile* in the reference manual *User and Solution Administration*.

- For security reasons it is recommended to create a specific Alfabet user for connection from ARIS to Alfabet in the Alfabet database that is not used for access via the Alfabet user interface or by other technical processes, like running ADIF jobs or accessing the Alfabet RESTful services from other client applications.

After having created a named user, the following settings must be performed to grant access to the Alfabet RESTful API:

- 1) In the table of the **User Administration** functionality, select the user that shall be used to send request to the Alfabet RESTful API.
- 2) In the toolbar, click the **Edit**  button. The **User** editor opens.
- 3) In the editor, go to the **Permissions** tab.
- 4) Select the checkbox **Has API V2 Access**.
- 5) Specify details about the access permissions with the following attributes:
  - **API Token Duration (minutes):** The RESTful service interface on client side must be implemented to send a request for authorization code prior to sending a data request. The authorization code received in the response of the authorization request can be used in data request calls to the Alfabet RESTful API posted within a limited amount of time, by default 20 minutes, after receiving the authorization code. The time period of the validity of the authorization code can be changed with this attribute. Enter the number of minutes the authorization code should be valid.
  - **API Access Options:** By default, most of the options are checked in this field to give the user access to all central functionality provided for the Alfabet RESTful services. Deselect all options that the user should not have permissions to perform. The following permissions are required for the ARIS - Alfabet Interoperability Interface and must be kept activated:
    - **Has Meta Model Access**
    - **Has GetObjectsByRefs Access**
    - **Has GetObjectsByReport Access**
    - **Has GetObjectsByFilter Access**
- 6) Click **Generate API Password**. A code is generated and stored in the Alfabet database.
- 7) Copy the code and store the information about user name and password for use on client side.

## Required Configuration on ARIS Side

Access to Alfabet must be activated for each database that shall have access to Alfabet in the configuration of the properties of the ARIS database:

- 1) Open the ARIS architect and login with an administrative user (e.g. "system") to your ARIS Connect server.
- 2) Log in to the database and open the **Properties** page.
- 3) Go to **Alfabet Integration** and enter the access data into the following fields:
  - **Connect to Alfabet:** Select the checkbox.
  - **Web service URL:** Enter the URL of the Alfabet Web Application.
  - **User interface URL:** Enter the URL of the Alfabet Web Application.
  - **User name (optional):** Enter the **API User Name** of the Alfabet user with the required access permissions.
  - **API password/Token:** Enter the **API Password** of the Alfabet user with the required access permissions.
  - **Use'Alfabet REST API v1':** Deselect the checkbox.
- 4) Click **Test connection** to test whether your settings are correct.

After configuration, disconnect all databases prior to testing the integration.

## Configuring the Alfabet Web Application to Read Data from ARIS

The configuration of the Alfabet application to connect to the RESTful API at ARIS is done in the configuration tool Alfabet Expand in the XML object **ArisApiConfig** and on the Alfabet user interface in the **Integration Solutions Configuration** functionality that is part of the **Configuration** module.

In the XML object ArisApiConfig, the connection to ARIS Connect is defined. It includes the following:

- Configuration of the connection to ARIS Connect, which requires a connection to an API for authentication and database access.
- Definition of the ARIS model types that a user can set a link to. This restriction can be individually defined for each individual object class.
- Definition of the ARIS model types that are allowed as start model for business process model integration.
- Optional mapping of custom properties defined for the Alfabet object classes business process model and business process to ARIS attributes to include these properties into integration.

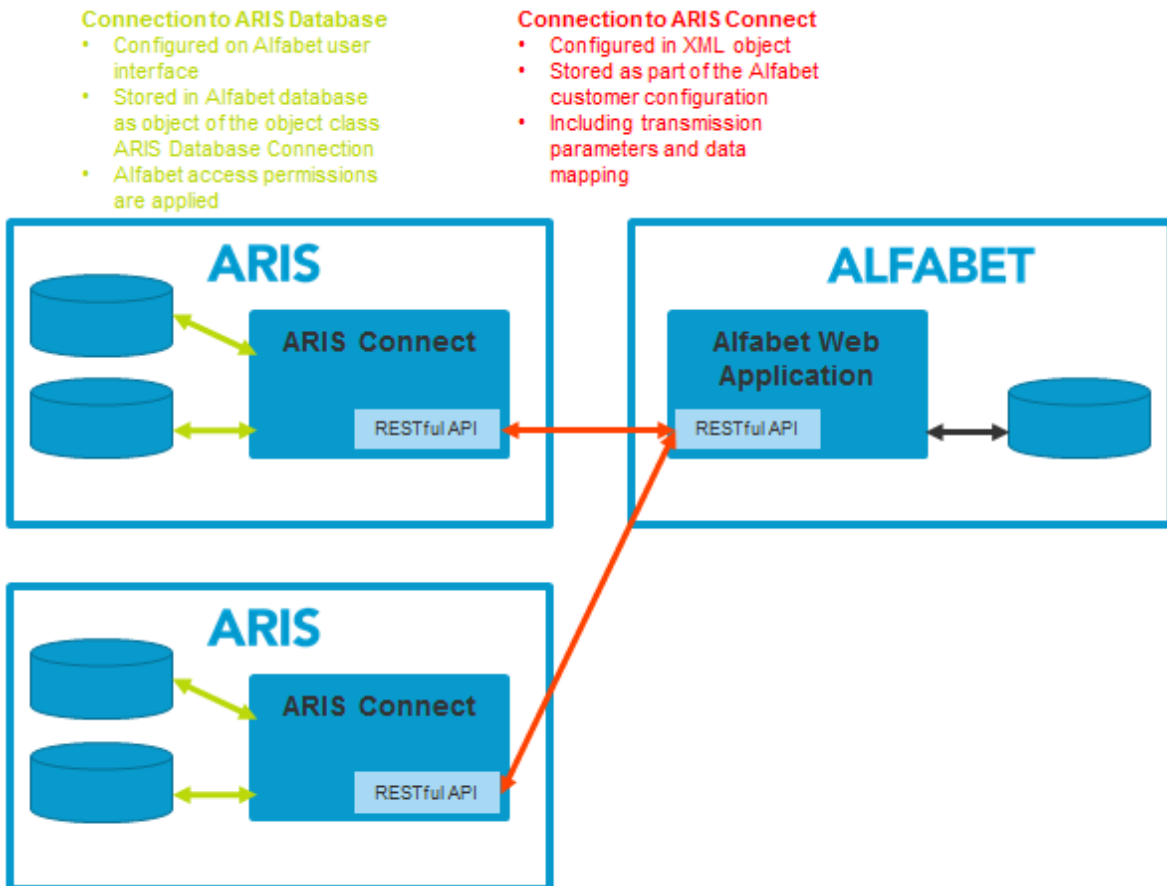
Connections to multiple ARIS Connect installations can be configured in the XML object **ArisApiConfig**. For each of the configured connections, access to individual databases that are accessible via the target ARIS Connect must then be defined in the **Administration** module on the Alfabet user interface. The configuration also includes specification of the database version that the connection targets. These individual database connections are the basis for each user interaction. The user must select the database that he/she wants to import data from prior to performing the action related with data import.

ARIS Database Connection is an Alfabet object class and ARIS database connections are stored as objects in the Alfabet database. Therefore Alfabet access permission concepts apply also to connections to ARIS databases. You can For example, hide ARIS database connections from a user via the mandate concept.



For detailed information about Alfabet access permission concepts and how to apply them to objects, see *Configuring Access Permissions for Alfabet* in the reference manual *Configuring Alfabet with Alfabet Expand*.

The following image gives an overview of the two configuration steps required:



Configuration includes the following:

- [Step 1: Configuring the Connection to ARIS Connect in the XML Object ArisApiConfig](#)
  - [Configuring Access to the ARIS Connect Server](#)
  - [Using Server Variables in the XML Object ArisApiConfig](#)
  - [Configuring the Mapping of Data for Business Process Model Integration](#)
  - [Configuring the Navigation to the ARIS User Interface](#)
  - [Configuring Connection via a Proxy Server](#)
- [Step 2: Configuring the Connection to the ARIS Database in the Integration Solutions Configuration Functionality](#)





The functionality of the ARIS - Alfabet Interoperability Interface is performed on special page views in the Alfabet user interface. The user profile of a user that shall use the functionality must be configured to show the view. For more information about the required views, see [Making the Views for the ARIS - Alfabet Interoperability Interface Available on the Alfabet User Interface](#).

## Step 1: Configuring the Connection to ARIS Connect in the XML Object ArisApiConfig

Connections to multiple ARIS connect servers can be configured in the XML object `ArisApiConfig`. Each connection is stored in an XML element `ArisApiConnection` within the root XML element `ArisApiConfig`. Within the XML element `ArisApiConnection`, the connection parameters for establishing a connection to ARIS Connect are defined with the XML attributes of the XML element `ArisApiConnection`, while mapping of data from ARIS to Alfabet is defined in child XML elements:

```
<ArisApiConfig>
  <ArisApiConnection name="IntegrationStandard" service="$ARIS_Service"
    api="/abs/api/" link_view="item" link_itemtype="default"
    auth_path="/umc/api/tokens" auth_tenant="$ARIS_Tenant"
    auth_name="$ARIS_UserName" auth_password="$ARIS_Password"
    languages="en_US,de_DE" data_portion="50" search_limit="200"
    import_empty_name="true">
    <ClassMapping class="ARIS_DiagramLink">
      <AttributeMapping aris_attr="status" alfa_attr="ARIS_STATUS" />
    </ClassMapping>
    ...
  </ArisApiConnection>
  <ArisApiConnection name="IntegrationSpecial"...>
    ...
  </ArisApiConnection>
  ...
</ArisApiConfig>
```

The required settings are described in the following:

- [Configuring Access to the ARIS Connect Server](#)
- [Using Server Variables in the XML Object ArisApiConfig](#)
- [Configuring the Mapping of Data for Business Process Model Integration](#)
- [Configuring the Navigation to the ARIS User Interface](#)
- [Configuring Connection via a Proxy Server](#)

Configuring Access to the ARIS Connect Server

- 1) In Alfabet Expand, go to the Presentation tab and expand the explorer nodes **XML Objects** > **IntegrationSolutions**.

- 2) Right-click on **ArisApiConfig** and select **Edit XML....** The XML object opens.
- 3) Set the XML attribute **active** of the **ArisApiConfig** XML element to `true` to activate the configuration.
- 4) Add a child XML element **ArisApiConnection** to the **ArisApiConfig** XML element and configure the connection to the ARIS RESTful API by setting the following XML attributes of the XML element **ArisApiConnection** to the values of your ARIS installation provided by Software AG:



In some of the XML attributes server variables can be used to read the value of the attribute at runtime from the server alias configuration of the Alfabet Web Application when a connection to ARIS is established. For information about server variables, see [Using Server Variables in the XML Object ArisApiConfig](#).

- **name:** Enter a unique name for the connection. This attribute is mandatory. The name is used to identify the **ArisApiConnection** in the configuration of ARIS database connection objects in the **Integration Solution Configuration** functionality required to specify the database that shall be opened via the connection.
- **service:** Enter the URL to ARIS Connect. This attribute is mandatory.
- **api:** Enter `/abs/api`. This attribute specifies the path to the ARIS RESTful API starting from the URL specified with the attribute **service**. This attribute is mandatory.
- **auth\_path:** Enter the path to the ARIS API. This is usually `/umc/api/tokens`. This attribute is mandatory.
- **auth\_name:** Enter the user name for authorization with the ARIS API.
- **auth\_password:** Enter the password for authorization with the ARIS API.
- **auth\_key:** Enter the certificate for authorization with the ARIS API. This entry is optional. For a standard implementation, this XML attribute should not be set and should be removed from the XML element to make sure that the built-in standard license key is used.
- **data\_portion:** Enter the maximum number of data sets that shall be exchanged within one transmission during the call. The default value is 100.
- **link\_view:** Enter `"item"`, if the view shall open in the Portal of ARIS Connect and `"thinclient"` if the link shall open in the ARIS Connect Designer.
- **link\_itemtype:** Enter `"default"`. This entry provides a mandatory technical term for that currently no other choices are implemented.
- **methodlanguage:** Define the method language that shall be used for the data transmission as locale ID with language and country code (For example, `de_DE` or `en_US`).
- **languages:** Define the languages for that data shall be transmitted as locale ID with language and country code (For example, `de_DE` or `en_US`) in a comma separated format.
- **import\_empty\_name:** Setting the name of an object in a primary language is not mandatory in ARIS. All language versions are equal. The user can define the name of an object in any language. The unique identifier for the object is the GUID. If this XML attribute is set to `true`, ARIS objects without a name defined in the primary language are imported and the name of these objects is set to `<no name defined for [ARISGUID]>` where `[ARISGUID]` is substituted with the GUID of the object in ARIS. If the XML attribute is set to `false`, objects with empty names in the primary language are not imported. By default, this attribute is set to `true` to ensure complete import of data.

The following XML attributes specify settings that are used as default values for the values defined for the ARIS database connection objects. If a value is defined here, it is used as fallback in case the value is not defined in the ARIS database connection object.

- **auth\_db:** Enter the name of the ARIS database that includes the data relevant for data integration.
- **auth\_tenant:** Enter the tenant for authorization with the ARIS API (For example, "default").
- **method\_filter:** Enter the name of the filter that shall be applied when accessing the ARIS database (For example, "Entire Method"). Alternatively, you can use the attribute `method_filter_guid` to specify the filter via its ARIS GUID instead of its name.
- **method\_filter\_guid:** Enter the ARIS GUID of the filter that shall be applied when accessing the ARIS database (For example, "Entire Method"). Alternatively, you can use the attribute `method_filter` to specify the filter via its name instead of its ARIS GUID.

The following XML attributes are interconnected and manage the performance on the data connection. For very huge amount of data to be retrieved, timeout errors might occur. In this case, adaptation of these attribute values can solve the problem. Tests with ARIS databases containing more than 5000 group nodes and 100000 objects have been passed successfully after modification of the parameter settings.

- **search\_limit:** Enter the maximum number of relevant objects to be retrieved from the ARIS database for a request as a positive integer. The default value is 100.
  - **timeout:** Enter the maximum number of seconds that a connection for a request is kept open for data transfer. The default value is 30 seconds.
  - **buffer\_size:** Enter the buffer size for the connection in kbytes. The default value is 200 kbytes.
- 5) Make sure that the XML element `ArisApiConnection` contains the following child element:

```
<ClassMapping class="ARIS_DiagramLink">
  <AttributeMapping aris_attr="status" alfa_attr="ARIS_STATUS" />
</ClassMapping>
```

- 6) Activate the navigation to the ARIS user interface for the relevant Alfabet object classes. There is no default navigation behaviour and navigation can only be performed from object classes included in the configuration. The configuration is described below in the section [Configuring the Mapping of Data for Business Process Model Integration](#).
- 7) Configure the mapping of object data between ARIS and Alfabet for business process models. A default behaviour is implemented and configuration is optional.

The configuration is described below in the section [Configuring the Navigation to the ARIS User Interface](#).

- 8) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.

#### Using Server Variables in the XML Object `ArisApiConfig`


The definition of server variables allows values for some XML attributes of the XML object `ArisApiConfig` to be defined in the server alias configuration. Defining the values in the server alias configuration instead of directly defining them in the XML object eases the propagation of changes.



For example, the configuration of the ARIS - Alfabet Interoperability Interface is done first in a test environment, and the test environment is an exact copy of the production environment except that the components are installed on different servers. Therefore the specification of the ARIS - Alfabet Interoperability Interface will be identical in the test and the production environment except for the data for the connection to the involved ARIS components. When migrating to the production environment, this would require a change to the `ArisApiConfig` XML object in Alfabet Expand. But if the connection parameters are defined as server variables in the server alias, and the configurations done in Alfabet Expand refer to the server variables in the XML attributes that define the connection to the ARIS RESTful API, the XML definition can be reused without changes in the production environment. Only the variable definition in the server alias of the production environment must be set to the current data for establishing the connection to the ARIS RESTful API of the production environment. A system administrator without access to Alfabet Expand can handle the whole migration procedure.

A value for an XML attribute can either be defined as a string directly in the XML element or via a server variable. It is not possible to define a value with a server variable written as part of a string. The complete value of the XML element in the XML object must be substituted with a server variable. In an XML object, the server variable definition must be specified with the equal symbol followed by the value to be assigned to the XML attribute in double quotes. For example: `<XML attribute>="SQLSERVER"`. The server variable is referenced in the relevant XML attribute as: `$(server variable name)`. For example, a server variable called `SQLSERVER` is referenced as `$(SQLSERVER)` in the XML object.

Server variables are defined in the Alfabet Administrator. To define a server variable to use in XML objects:

- 1) In the Alfabet Administrator, click the **Alfabet Aliases** node in the **Administrator** explorer.
- 2) In the table on the right, select the server alias that you want to define a server variable for and click the **Edit**  button. The alias editor opens.
- 3) Go to the **Variables** tab and click the **New** button. A dialog box opens.
- 4) In the **Variable Name** field, enter a unique name for the server variable.



The server variable name may only contain letters (English alphabet), numbers, and the underscore symbol.

- 5) In the **Variable Value** field, specify the value to use for the server variable.



If the variable value contains a special character according to XML standards (for example: `&`, `%`, `;`, `<`, `>`), these characters must be replaced by with their respective XML compliant code (for example: `&amp;` for `&`)

- 6) Click **OK** to save your changes. The server variable definition appears in the list of server variables.



To edit or delete the server variable, select the server variable in the table and click the **Edit** or **Delete** button below the table.

- 7) Click **OK** to save your changes and close the editor. The server variable definition is now available in the server alias configuration and can be used in the relevant XML objects available in the **Integration Solutions** folder in Alfabet Expand.

### Configuring the Mapping of Data for Business Process Model Integration

For the integration of ARIS business process models into Alfabet, a default behavior is configured that can be changed in the following ways:


- By default, business process models in Alfabet include all ARIS functions in a selectable source ARIS business process model diagram as root business processes and all functions in diagrams which are linked to the functions in the source diagram and are of the same type as the source business process model diagram as subordinate business functions. The hierarchy of diagrams linked to each other is used to build the hierarchy.

This default behaviour can be changed by adding one or multiple XML elements **ImportMapping** to the configuration of the connection to ARIS connect in the XML object **ArisApiConfig**. The integration processes is then changed in the following way:

- The hierarchy of business processes is build on connections between objects configured in ARIS. This methods offers the advantage that a hierarchy can also be build for objects in the same diagram if they are interconnected with the connection type used for building the hierarchy. Which connection type is used for building the connection is configurable per diagram type.
- Optionally other object types than functions can be regarded as business processes during integration. This functionality is only available for the integration process based on connections.
- By default, standard attributes of ARIS objects like name and description are taken over when an Alfabet business process is created. Both in ARIS and in Alfabet, customers can define custom attributes/properties to store company specific information. These customer defined information can be included in the integration process and taken over from ARIS to Alfabet when the business process model is build or updated.

To build the Alfabet business process model hierarchy on connections between ARIS objects:

- 1) In Alfabet Expand, go to the **Presentation** tab and expand the explorer nodes **XML Objects > IntegrationSolutions**.
- 2) Right-click on **ArisApiConfig** and select **Edit XML....** The XML object opens.
- 3) Add the following child element to the **ArisApiConnection** XML element defining the connection for that you want to reconfigure data mapping:
 

```
<ClassMapping class="BusinessProcessModel">
</ClassMapping>
```
- 4) Add one child element **ImportMapping** to the **ClassMapping** XML element for each ARIS diagram type that you would like to use as source business process model diagram during integration.
- 5) Set the following XML attributes for the **ImportMapping** XML element:
  - **model\_type**: Enter the ARIS API name of the ARIS model type that shall be allowed to be used as starting point for integration. This XML attribute is mandatory.
  - **parent\_rel**: Enter the ARIS API name of the ARIS connection type that shall be allowed to be used as starting point for integration. This XML attribute is mandatory.
  - **proc\_type**: Define the object classes that shall be regarded as business processes during integration in a comma separated list of ARIS API names of the object types. This definition is optional. If the XML attribute is empty or not set, only ARIS functions are regarded as business processes. Please note that the list of classes is overwriting the default and the *function* object type (OT\_FUNC) must be included into the specification to be regarded as business process if the **proc\_type** is defined.
- 6) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.




The following configuration allows Value-Added Chain Diagrams (MT\_VAL\_ADD\_CHN\_DGM) and Function Tree Diagrams (MT\_FUNC\_TREE) as starting point for integration of business process models into Alfabet. For Value-Added Chain Diagram the business process hierarchy is build on the connection "is process oriented-superior" (CT\_IS\_PRCES\_ORNT\_SUPER) and *ARISfunctions* (OT\_FUNC) and *processes* (OT\_PROC) are regarded as business processes in Alfabet. For Function Tree Diagram the business process hierarchy is build on the connection "is component of" (CT\_CONS\_OF\_1) and the default behaviour converting *ARISfunctions* to Alfabet business processes applies:

```
</ArisApiConfig active="true">
  <ArisApiConnection...>
    <ClassMapping class="BusinessProcessModel">
      <ImportMapping model_type="MT_VAL_ADD_CHN_DGM"
        parent_rel="CT_IS_PRCES_ORNT_SUPER"
        proc_type="OT_FUNC,OT_PROC"></ImportMapping>
      <ImportMapping model_type="MT_FUNC_TREE"
        parent_rel="CT_CONS_OF_1"></ImportMapping>
    </ClassMapping>
  </ArisApiConnection>
</ArisApiConfig>
```

To import ARIS attributes as Alfabet properties into the Alfabet business processes during integration:

- 1) In Alfabet Expand, go to the **Presentation** tab and expand the explorer nodes **XML Objects > IntegrationSolutions**.
- 2) Right-click on **ArisApiConfig** and select **Edit XML....** The XML object opens.
- 3) Add the following child element to the **ArisApiConnection** XML element:

```
<ClassMapping class="BusinessProcess">
</ClassMapping>
```

- 4) Add one child element **AttributeMapping** to the **ClassMapping** XML element for each attribute that you want to include into the integration process. Please note that the Name and Description object class properties in Alfabet are mapped by default. Any mapping defined for these object class properties in an XML element **AttributeMapping** will be ignored.
- 5) Set the following XML attributes for the **AttributeMapping** XML element:
  - **aris\_attr**: Enter the ARIS API name of the source ARIS attribute. This XML attribute is mandatory.
  - **alfa\_attr**: The value of the Name attribute of the target Alfabet object class property. This XML attribute is mandatory.
- 6) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.



The following configuration imports an ARIS status value to a custom property ARIS\_STATUS in Alfabet:

```
<ArisApiConfig active="true">
  <ArisApiConnection...>
    <ClassMapping class="BusinessProcess">
```

```


        <AttributeMapping aris_attr="status"
        alfa_attr="ARIS_STATUS" />
    </ClassMapping>
</ArisApiConnection>
</ArisApiConfig>

```

### Configuring the Navigation to the ARIS User Interface

An Alfabet user can only set links to diagrams in the ARIS user interface if the diagram types that are defined in the **ArisApiConfig** XML object in Alfabet Expand for the respective **ArisApiConnection**.

To activate and configure navigation from the Alfabet to the ARIS user interface:

- 1) In Alfabet Expand, go to the **Presentation** tab and expand the explorer nodes **XML Objects > IntegrationSolutions**.
- 2) Right-click on **ArisApiConfig** and select **Edit XML...**. The XML object opens.
- 3) For each object class for that navigation to the ARIS shall be configured, add a child element **ClassMapping** to the **ArisApiConnection** XML element that you want to configure. The XML element **ClassMapping** must have an XML attribute **class** set to the value of the Name attribute of the Alfabet object class.
- 4) Add one child element **ModelLink** to the **ClassMapping** XML element for each ARIS business process model diagram that shall be allowed as link target.
- 5) Set the following XML attributes for the **ModelLink** XML element:
  - **type**: Enter the ARIS API name of the ARIS model type that shall be allowed as link target. This XML attribute is mandatory.
  - **status**: Enter the ARIS status of the ARIS model type that shall be allowed as link target. If all status values shall be included, enter "all". This XML attribute is mandatory.
- 6) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.



The following configuration allows users to navigate from the Alfabet object class business process to four different diagram types in ARIS:

```


<ArisApiConfig active="true">
  <ArisApiConnection...>
    <ClassMapping class="BusinessProcess">
      <ModelLink type="MT_VAL_ADD_CHN_DGM"
      status="all"></ModelLink>
      <ModelLink type="MT_EEPC" status="all"></ModelLink>
      <ModelLink type="MT_BPMN_COLLABORATION_DIAGRAM"
      status="all"></ModelLink>
      <ModelLink type="MT_BPMN_PROCESS_DIAGRAM"
      status="all"></ModelLink>
    </ClassMapping>
  </ArisApiConnection>

```

```
</ArisApiConfig>
```

### Configuring Connection via a Proxy Server

Optionally, you can configure the ARIS - Alfabet Interoperability Interface to send requests to the ARIS RESTful API via a proxy server. This configuration can be done individually for each connection. This requires the following additional configuration in the XML element **ArisDataConnection** for that the proxy server shall be used:

- 1) In Alfabet Expand, go to the **Presentation** tab and expand the explorer nodes **XML Objects > IntegrationSolutions**.
- 2) Right-click on **ArisApiConfig** and select **Edit XML....** The XML object opens.
- 3) Add a child XML element `Proxy` to the XML element `ArisApiConnection` for that you want to configured use of a proxy server.
- 4) Define the following XML attributes for the XML element `Proxy`:
  - `url`: Define the URL of the proxy server.
  - `user`: If required, enter the user name for access to the proxy server. The domain name for authentication is defined separately with the XML attribute `domain` and must not be specified as part of the user name.
  - `psw`: If required, enter the password for access to the proxy server.
  - `domain`: If required, define the domain name that shall be used as part of the user name for authentication at the proxy server.
- 5) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.

### Step 2: Configuring the Connection to the ARIS Database in the Integration Solutions Configuration Functionality

After having configured one or multiple connections to ARIS Connect in the XML object **ArisApiConfig** in Alfabet Expand, the connections to different databases that are available via the same ARIS Connect must be specified as Alfabet database objects. This step is mandatory, even if only a single connection to ARIS Connect is defined and only one ARIS database is available.

Configuration of ARIS database connection data is performed in the **Integration Solutions Configuration** functionality on the Alfabet user interface. The resulting connection definitions are Alfabet database object, which provides the following advantages:

- Alfabet access permission concepts can be applied to ARIS database connections. That means that only users authorized by For example, right rules or authorized user or authorized user group definition of the object may change the data for the ARIS database connection in the the **Integration Solutions Configuration** functionality.
- ARIS database connections have a release status. The release status is displayed to users selecting an ARIS database connection in a selector, For example, when setting a link to an ARIS diagram. The user can see from this information, whether the ARIS database connection is released for productive use or still under construction or in a test phase. If the ARIS database connection is retired, it is not



displayed any longer to users in the drop-down lists for selection of an ARIS database connection in selectors.

- 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.

## Making the Views for the ARIS - Alfabet Interoperability Interface Available on the Alfabet User Interface

The user can execute the functionalities that require a connection to ARIS on the Alfabet user interface in the views listed in the following table. The Alfabet solution must be configured to provide access to the views for the relevant users. The required configuration is listed in the table.

View Name	View Caption	Available for Alfabet Object Class	Functionality for ARIS - Alfabet Interoperability Interface	
PRMD_Root-Processes	Business Processes	Business Process Model	Button interaction to start integration of an ARIS business process model into the current, empty business process model.	<p>The page view is a standard view that is also used for building business process models directly in Alfabet. The button is automatically available if</p> <ul style="list-style-type: none"> <li>The XML attribute <b>active</b> of the <b>ArisApiConfig</b> XML element is set to "true" in the XML object <b>ArisApiConfig</b> and</li> <li>the business process model does not have business processes assigned.</li> </ul>
ARIS_BMP-Source_Report	Business Process Model ARIS Source Report	Business Process Model	Check and update of changes for Alfabet business process models based on an ARIS business process model.	<p>The user profile of a user that shall use the functionality must be configured to show the view. This requires that the view is added to the object profile of the business process model used for the user profile and the view is configured to be visible.</p> <p>For information about the configuration of user profiles consult the reference manual <i>Configuring Alfabet with Alfabet Expand</i>.</p>
ARIS_DiagramLinks	ARIS Diagrams	any object class for that it is configured	Definition of links to the ARIS user interface and navigation via the links.	<p>The user profile of a user that shall use the functionality must be configured to show the view. This requires that the view is added to the object profile of the business process model used for the user profile and the view is configured to be visible.</p> <p>For information about the configuration of user profiles consult the reference manual <i>Configuring Alfabet with Alfabet Expand</i>.</p>

## Chapter 4: Customer Configured Integration of Alfabet Data to ARIS

By default, both Alfabet and ARIS are preconfigured to provide the standard import of applications described in the section [Functionality of the ARIS - Alfabet Interoperability Interface](#). The configuration can be changed in the following ways:

- Additional data about the standard objects can be included into the import.
- Import of data can be filtered both on Alfabet and on ARIS side.
- Import can be configured to include objects of object classes not included in the standard import.

To change the standard import, the following configuration is required:

- [Configuring Data Export from Alfabet](#)
- [Configuring Data Import to ARIS](#)

### Configuring Data Export from Alfabet

In Alfabet, data to be exported to ARIS is not read directly from the Alfabet database, but read from configured tabular reports that return the sub-set of data about objects of an object class that are relevant for the export.

For the standard export of data to ARIS, a set of preconfigured reports is available. These reports are visible in Alfabet Expand in the **Reports** explorer in the report folder **DataImportIntoARIS**. The reports are private and cannot be changed. To change the import of objects into ARIS, new configured reports of the type `Query` or `NativeSQL` must be created that provide the data.

For the standard object classes, the available private reports can be copied into the new configured report and adapted. For additional object classes to import into ARIS, the reports must be defined from scratch.

Please note the following about the required output of the configured report:

- The data types that are returned must match the data types expected on ARIS side and all data required by ARIS must be included.
- Columns returning Alfabet object class properties storing references must return the `REFSTR` value of the referenced object.
- Columns returning Alfabet object class properties storing reference arrays must return a comma separated list of `REFSTR` values of the referenced object.

### Configuring Data Import to ARIS

In ARIS, the standard mapping of Alfabet objects and object properties to ARIS objects and ARIS object attributes must be changed to include the changes in the Alfabet reports for export of data to ARIS:

- 1) Open the ARIS architect and login with an administrative user (e.g. "system") to your ARIS Connect server.
- 2) Open the **Administration** tab.
- 3) In the explorer, navigate to **Evaluation > Reports > IT Architecture**.

- 4) Open the XML report **ARIS-ALFABET-mapping-V2.xml**.
- 5) For each object class for that object data is imported from Alfabet to ARIS, an XML element **class** is available in the configuration file. Do the following to change the configuration:
  - To adapt one of the existing XML elements **class** to import from a customer configured report instead of the standard Alfabet report for ARIS integration:
    - In the XML element **class**, set the XML attribute `report` to the name of your customized report.
    - In the mapping section change and extend the mapping between data imported from Alfabet to ARIS attributes. For each XML element **attr**, the XML attribute `alfabet` must return the name of the column in the dataset of the configured Alfabet report returning the data, and the XML attribute `aris` must return the name of the corresponding ARIS attribute.
    - Optionally, filters, connections and detail settings about the import can be changed. For details see the ARIS documentation.
  - To add a new object class to the mapping, add a new XML element **class** to the mapping table and set all required XML attributes and child XML elements. Details about all available settings are provided in the ARIS documentation.

## Chapter 5: Customer Configured Integration of ARIS Data to Alfabet

As part of the ARIS - Alfabet Interoperability Interface, Alfabet provides an import mechanism that can read any current data from ARIS diagrams into temporary tables in the Alfabet database. This import mechanism can be further configured by customers to import data from the temporary tables into database tables that are part of the Alfabet meta-model.

The import is based on the RESTful API architecture of the ARIS - Alfabet Interoperability Interface in combination with the Alfabet Data Integration Framework (ADIF). Configuration is partially performed in the XML object **ArisGenericDataConnectivity** but also requires configuration of an ADIF import scheme.

Information about the configuration of ADIF import schemes is provided in the reference manual *Alfabet Data Integration Framework*. In the following description, basic knowledge about the configuration is assumed.

The data import requires a consistent conceptualization and modelling of ARIS diagrams. The import mechanism simply imports all data about defined objects and connections included in defined diagram types. This information can then be used to update information in the Alfabet database. For example, a connection of an application system type to a function in a specific ARIS diagram can be interpreted as identical to an Alfabet database business support. As part of the import of data from ARIS to Alfabet, an object of the object class Business Support is created in the Alfabet database for each new connection between a function and an application system type in one of the specific ARIS diagrams and the references to other objects in the Alfabet database that can be consistently read from the ARIS data, like the reference to the application providing the business support are also created during import.

Customer defined import mechanisms require a valid configuration of the ARIS - Alfabet Interoperability Interface to be available, that means that the configurations described in the chapter [Activating and Configuring the ARIS - Alfabet Interoperability Interface](#) are all available. For each data import, the relevant **ArisApiConnection**, as defined in the XML object **ArisApiConfig**, must be selected. One or a subset of ARIS database connections defined for the ArisApiConnection can also be selected to limit data import to one or a set of databases accessed with defined method filter settings. If no subset of ARIS database connections is defined, the import mechanism imports all data from all ARIS database connections that are defined to use the relevant **ArisApiConnection**, if the ARIS database connections are not explicitly specified in the ArisGenericDataConnectivity. That means that data from different databases may be included into the import and consistency in conceptualization and modelling between the ARIS diagram in the different databases must be ensured.

The import is configured in two steps:

- The model diagram types, object types, connection types and groups that are relevant for the data integration must be defined in the XML object **ArisGenericDataConnectivity** to limit data import to the temporary tables to the relevant data. Only the data specified in the configuration is requested from ARIS via the RESTful services of the ARIS - Alfabet Interoperability Interface.
- Import of data from ARIS is managed via an ADIF import scheme created via an assistant that implements the required basic configuration. The assistant creates a number of ADIF import sets in the ADIF import scheme. The ADIF import sets are configured to import data from ARIS to temporary database tables in the Alfabet database. This ADIF import scheme must be amended with the configuration for reading the data from the temporary database tables in a defined way into the database tables of the class model of the Alfabet database.

After having configured the import mechanism, the ADIF import scheme must be executed to perform update of data in the Alfabet database. The ADIF import scheme both triggers the execution of the RESTful service request and changes the data in the Alfabet database according to the provided information.



For ADIF imports of large data sets, the execution of the RESTful service requests may take some time. If the ADIF job is started with the command line option `-logverbose`, information about the execution of the RESTful service requests is written into the log file at runtime which makes it possible

to monitor whether the process is still executed correctly. If `-logverbose` is not used, no log information is written into the log file during the time of the execution of background processes that are not integral part of ADIF but triggered by the ADIF job.

Information about the execution of ADIF schemes is provided in the reference manual *Alfabet Data Integration Framework*.


## Configuration of the ARIS - Alfabet Interoperability Interface to Perform Customized Data Import

For performance reasons it is essential to limit the data imported from ARIS into temporary tables in the Alfabet database. The ADIF scheme for data import from ARIS triggers a request via the RESTful services between ARIS and Alfabet. All data returned by the request are imported into the temporary tables. The configuration of object types that shall be imported is therefore performed on the level of the RESTful service configuration in Alfabet Expand.

Configuration of data import is done in the XML object **ArisGenericDataConnectivity**. The **GenericConnectivity** root XML element of this XML object can have multiple child elements **DataConnection**, each including a filter for import of specific data. All data that is imported is written into the same temporary table, but the name of the data connection defined with the XML attribute **DataConnectionName** is added as column to inform about which data connection delivered the data. The configuration of multiple data connections can be used For example, to import data for multiple different data updates. If For example, data from the same diagram types shall be used to generate business supports and capability maps in Alfabet, the connection type can be used to provide the information whether the data is relevant for the generation of business supports or for the generation of capability maps. A later decision not to integrate For example, the capability map data from ARIS diagrams can then easily be implemented by deactivating the data connection for the capability maps without affecting the other integrations.

Filtering is done in multiple levels which is also represented in the structure of the XML elements in the filter, that are hierarchically structured. First, a relevant Arisrelevant models or model types are to be specified. Within the relevant models, a filter for relevant objects can be defined. On the last level of the filter, ARIS groups can be defined to limit import to models and objects belonging to the defined ARIS groups.

To define customized data import from ARIS via the Alfabet RESTful API :

- 1) In Alfabet Expand, go to the **Presentation** tab and expand the explorer nodes **XML Objects > IntegrationSolutions**.
- 2) Right-click on **ArisApiConfig** and select **Edit XML....** The XML object opens.
- 3) For each object class for that navigation to the ARIS user interface shall be configured, add a child element **ClassMapping** to the **ArisApiConfig** XML element with the XML attribute **class** set to the value of the Name attribute of the Alfabet object class.
- 4) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.
- 5) In the explorer, right-click on the **ArisGenericDataConnectivity** XML object and select **Edit XML....** The XML object opens.
- 6) Add the following attribute to the root element **GenericConnectivity**:
  - **MaxThreadCount**: Enter the maximum number of parallel threads on the data connection. This parameter is mandatory. The recommended value is 16.

- 7) Add at least one child element **DataConnection** to the **GenericConnectivity** XML element and add the following XML attributes to the **DataConnection** XML element:
- **DataConnectionName:** Enter a unique name for the data connection. The name is written into the temporary tables created via the ADIF import scheme **Alfabet\_Standard\_Import\_from\_ARIS** into the column `DATACONNECTIONNAME`. This XML attribute is mandatory.
  - **IsActive:** Enter "true" to activate data import for this data connection. Enter "false" if you do not want to import the data defined for this connection. This XML attribute is mandatory.
  - **NetChange:** Enter `false`. The import of changed data only will be available with Alfabet 9.12 and shall not be activated with Alfabet 9.10.
  - **ArisConnections:** Enter a comma separated list of ARIS database connections to limit data import to data from the listed ARIS database connections.
  - **ArisDBConnectionStatusSet:** Enter a comma separated list of release status values defined for ARIS database connections to limit data import to data from ARIS database connections with one of the defined release status values.



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*. The restriction of import to ARIS database connections with a defined release status allows, for example, to exclude ARIS database connections currently defined for testing purposes or retired because of changes in the ARIS database configuration.

- 8) Enter a child XML element **ModelFilter** to the XML element **DataConnection**. The XML element **DataConnection** can only have one child element **ModelFilter**.
- 9) Add the following XML attributes to the **ModelFilter** XML element.

Add either:

- **Type:** Enter a comma separated list of ARIS model types to limit data import to data from the listed model types. The model type is defined by its API name.
- **ATFilter:** Enter a search expression for ARIS attribute settings to limit data import to models with the defined attribute settings. the search expressions have the following syntax:
  - A simple search expression consists of `<attribute type> <operator> <value>`, for example:  
`AT_NAME=Jones.`
  - The attribute type can be defined by the API name, integer type number or type GUID of the attribute type.
  - Allowed operators are:
    - for numbers or dates: `=, !=, >, <, >=, <=`
    - for boolean values: `=, !=`
    - for text: `=`
  - independent from value data type `+` can be used for maintained, `-` for not maintained. `+` and `-` are used without value specification. For example, if the attribute `AT_AUTH` shall be set, the expression must be set as:



AT\_AUTH+

- expressions can be combined with an AND or OR operator. If expressions are combined, each of the combined expressions must be written into curly brackets. Curly brackets can also be used for grouping within the expression. For example:

```
{{AT_NAME = Jones} OR {AT_DESC = Human Resources}} AND {AT_AUTH -}
```

If curly brackets are used, the values that you are searching for must not contain "or". If the search value contains a curly bracket, use the backslash as escape parameter to mark it as part of the search value and not the search expression.

- When searching for text, any attribute value containing the defined string is matching the search. Whitespaces and dashes are separating strings and the search will find all results that contain the two separated strings in any order. For example, the search for

```
AT_NAME=holder certificate
```

will find "holder certificate" as well as "certificate holder" or "the Holder Certificate".

- When searching for text, an asterisk can be used as placeholder. For example:
  - A\* will find all strings starting with A and containing at least one character following the A.
  - A\*B will find all strings starting with A and ending with B with at least one character between.
- QueryFilter:** Define a native SQL query to select the diagrams that shall be imported. The query must return a dataset with one column `ARIS_GUID` and find one or multiple `ARIS_GUID` values of ARIS models. For example:

```
SELECT 'a105a310-3eb6-11e6-022b-0050568c1d14' AS ARIS_GUID
```

10) Enter a child XML element **ObjectFilter** to the XML element **ModelFilter**. The XML element **ModelFilter** can only have one child element **ObjectFilter**.

11) Add the following XML attributes to the **ObjectFilter** XML element:

- Type:** Enter a comma separated list of ARIS object types to limit data import to data from the listed object types. The object type is defined by its API name.
- ATFilter:** Enter a search expression for ARIS attribute settings to limit data import to objects with the defined attribute settings. the search expressions have the following syntax:
  - A simple search expression consists of <attribute type> <operator> <value>, for example:  
AT\_NAME=Jones.
  - The attribute type can be defined by the API name, integer type number or type GUID of the attribute type.
  - Allowed operators are:
    - for numbers or dates: =, !=, >, <, >=, <=
    - for boolean values: =, !=
    - for text: =
  - independent from value data type + can be used for maintained, - for not maintained. + and - are used without value specification. For example, if the attribute AT\_AUTH shall be set, the expression must be set as:

AT\_AUTH+

- expressions can be combined with an AND or OR operator. If expressions are combined, each of the combined expressions must be written into curly brackets. Curly brackets can also be used for grouping within the expression. For example:

```
{{AT_NAME = Jones} OR {AT_DESC = Human Resources}} AND {AT_AUTH -}
```

If curly brackets are used, the values that you are searching for must not contain "or". If the search value contains a curly bracket, use the backslash as escape parameter to mark it as part of the search value and not the search expression.

- When searching for text, any attribute value containing the defined string is matching the search. Whitespaces and dashes are separating strings and the search will find all results that contain the two separated strings in any order. For example, the search for

```
AT_NAME=holder certificate
```

will find "holder certificate" as well as "certificate holder" or "the Holder Certificate".

- When searching for text, an asterisk can be used as placeholder. For example:
    - A\* will find all strings starting with A and containing at least one character following the A.
    - A\*B will find all strings starting with A and ending with B with at least one character between.
- 12) Optionally, enter a child XML element **ModelGroupNames** to the XML element **ModelFilter**. The XML element **ModelFilter** can only have one child element **ModelGroupNames**.
- 13) Enter one or multiple child XML elements **Group** to the XML element **ModelGroupNames** and add the following XML attribute to the **Group** XML element:

- Name:** Enter the name of an ARIS model group.

The data import will be limited to the models that belong to groups defined with the XML elements **Group**. Please note that the filter will find all groups with the given name defined for any of the languages that are specified in the XML object **ArisApiConfig** with the XML attribute `language` of the XML element `ArisApiConfig`.

The definition of the group filters is optional. If no group filter is defined, models are imported independent from the group they belong to.

- 14) Optionally, enter a child XML element **ObjectGroupNames** to the XML element **ObjectFilter**. The XML element **ObjectFilter** can only have one child element **ObjectGroupNames**. The data import will be limited to the objects that belong to groups defined with the XML elements **Group** that are part of the **ObjectFilter**. The definition of the group filter is optional. If no group filter is defined, models are imported independent from the group they belong to.



The **ObjectGroupNames** filter is only applied if XML attributes of the XML element **ModelFilter** are empty. If the **ModelFilter** definition limits import to a subset of models, all objects that are used in the selected models and that are matching the filter defined in the XML element **ObjectFilter** are imported independent from their group assignment.

- 15) Enter a child XML elements **Group** to the XML element **ObjectGroupNames** and add the following XML attribute to the **Group** XML element:

- Name:** Enter the name of an ARIS object group.

Please note that the filter will find all groups with the given name defined for any of the languages that are specified in the XML object **ArisApiConfig** with the XML attribute `language` of the XML element `ArisApiConfig`.

- 16) Optionally, you can change the value of the XML attribute `MaxThreadCount` of the XML root element `GenericConnectivity`. This setting defines, how many parallel threads are used to access the ARIS server when reading the data. A low value will slow down the process of data integration. A high value will fasten the process of data integration but if the value is too high, this will lead to performance issues for users concurrently working with ARIS. The default value for the XML attribute `MaxThreadCount` is 8, which has been tested to provide good results with regard to speed and performance.

- 17) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.

## Configuration of the ADIF Import Scheme

Running an ADIF import scheme configured via the assistant `ARISImport_Assistant` triggers the following import actions:

- A RESTful service request is sent to the RESTful API of the ARIS connect server. The data is returned in JSON format.
- The JSON data is imported into predefined temporary tables.
- Data from the temporary tables is taken over into the database tables of the Alfabet meta-model. This part of the import is not predefined by Software AG but must be configured by the customer. Running the assistant based ADIF import scheme without prior configuration of the import logic will not lead to any changes in the Alfabet database.
- The temporary database tables are deleted.

The ADIF import scheme must be created and configured as described below. It must then be executed in the required intervals.

The way data import is configured and executed via ADIF schemes is described in the reference manual *Alfabet Data Integration Framework*. This documentation is limited to a description of the temporary tables that are written during import of the ARIS data to ease the implementation of the conversion into correct Alfabet data.

The following information is available:

- [Creating an ADIF Import Scheme With the ARISImport Assistant](#)
- [Temporary Tables Created During Import](#)
  - [Temporary Table Containing Models](#)
  - [Temporary Table Containing Objects](#)
  - [Temporary Table Containing Connections](#)
  - [Temporary Table Containing Assignments](#)
  - [Temporary Table Containing Attributes](#)
- [Considerations for Data Import](#)

## Creating an ADIF Import Scheme With the ARISImport\_Assistant

After having defined a valid configuration for data import in the XML objects **ArisApiConfig** and **ArisGenericDataConnectivity**, you must define a new ADIF import scheme to configure data integration using the ARIS import assistant:

- 1) In the **ADIF** tab in Alfabet Expand, right-click the **ADIF Schemes** root node in the explorer or any sub-folder and select **Create Import Scheme**. The new import scheme is added to the explorer. The attribute window of the new import scheme is displayed on the right.
- 2) In the attribute window, set the following attributes for the ADIF import scheme:
  - **Name:** Enter a unique name. The name is used to identify the ADIF import scheme in technical processes. It must be unique and should not contain white spaces or special characters.
  - **Assistant:** Select `ARISImport_Assistant` from the drop-down list.
  - **Import File Required :** Select `False`.
  - **Commit After Run :** If set to `True`, the result of the data import is written persistently to the Alfabet database. If set to `False`, the import process will be rolled back after execution and no changes will be written to the database. Configuration of the automatic start of workflows during import is ignored when **Commit After Run** is set to `False`. It is recommended that you set **Commit After Run** to `False` for a new import scheme to allow debugging without the risk of corrupting the database. After the successful testing of the data import and verification that the resulting changes to the Alfabet database are as expected, you can reset the **Commit After Run** attribute to `True` to perform regular data import.



Please note the following:

- Setting the **Commit After Run** attribute rolls back all changes to data records in existing tables caused by DML statements. The creation or deletion of tables is not included in the roll back. For example, if you test an ADIF scheme that is configured to persistently write temporary tables to the database, these temporary tables will be created persistently even if **Commit After Run** is set to `False`. SQL commands of the type **OnActivate** are also excluded from roll back.
- When new objects are created during an ADIF import job, the data bind mechanism assigns `REFSTR` values for the new objects. When **Commit After Run** is set to `False`, the objects are not created in the database, but nevertheless the `REFSTR` values are regarded as in use and will not be used for data bind in the next ADIF run unless the Alfabet Server or Alfabet Expand application used to process the ADIF jobs is restarted.
- Changes triggered by **OnActivate** commands are not rolled back if the option **Commit After Run** is set to `False` for the import scheme.
- **Drop Temp Tables :** If set to `True`, all temporary tables are dropped after import. Only the changes to the Alfabet database are stored persistently. If set to `False`, the temporary tables are kept in the database after import is finished. Storing temporary tables persistently is only required for special import/export cycles designed for data manipulation that require input from the temporary tables of a previously set import. In most cases, setting this attribute to `True` is recommended to clean the database of data that is not part of the Alfabet meta-model.

- 3) In the explorer, right-click the node of the new ADIF import scheme and select **Create ADIF Scheme Details Using the ARIS Import Assistant**. A warning dialog is displayed.



Using the assistant with an existing, already configured ADIF import scheme will overwrite all automatically generated parts of the ADIF scheme. If changes have been made to these import sets/entries, these changes will get lost.

- 4) Click **Yes**. The assistant opens in your standard Web browser.
- 5) In the field **Select Data Connections to Include**, all XML elements **DataConnection** in your **ArisGenericDataConnectivity** XML object will be displayed. Click on each data connection for that data shall be integrated with this ADIF import scheme.
- 6) After having selected all relevant **DataConnection** XML elements, click the button **OK** below the field. A new page is displayed that reminds you to rescan the ADIF tree in Alfabet Expand.
- 7) Close the browser window and return to the **ADIF** tab of Alfabet Expand.
- 8) Right-click the ADIF import scheme and select **Rescan Tree** in the context menu. You will then see all automatically generated ADIF elements in the ADIF scheme. Each ADIF import entry will have data import to temporary tables defined via the attribute elements in the **Attributes** folder. The import to the standard Alfabet database tables is not included into the configuration.
- 9) Configure the data integration to the standard Alfabet database tables according to your demands. For information about the configuration options available via ADIF import schemes see the reference manual *Alfabet Data Integration Framework*.

## Temporary Tables Created During Import

The assistant `ArisImport_Assistant` for an ADIF import scheme creates one temporary table per ARIS data type. For example, data about ARIS objects are written into one temporary table independent from the model the data is derived from. For each object listed in the table the information about the model that the object is used in and the data connection the data is fetched for is included into the information written into the table.

The tables listing models, objects, and the relations and connections linking objects to other objects in the same model diagram or to other model diagrams contain a subset of attributes that are considered important for import. Independent from that, the attributes defined for the models, objects and relations are also included into the table that lists information about attributes if not specified otherwise in the configuration of the data connection in the XML object **ArisGenericDataConnectivity**.

It is important to note that there is a size restriction configured for many of the database table columns of the temporary tables. For some columns, the values are truncated to the defined maximum allowed data size. If this is the case, it is mentioned in the following overview of table data.

The attribute values defined in the temporary table for attributes are never truncated. Therefore if For example, very long names are defined for objects, the attribute Name for the object should be included in the import of attributes although it is also listed in the table for objects to make sure that the complete value is available.

The following temporary tables are created during import:

- [Temporary Table Containing Models](#)
- [Temporary Table Containing Objects](#)
- [Temporary Table Containing Connections](#)
- [Temporary Table Containing Assignments](#)

- [Temporary Table Containing Attributes](#)

### Temporary Table Containing Models

**Table name :** TMP\_MODEL

The table has the following columns:

Database column Name	Content	Truncated to:	Data Type
DATACONNECTIONNAME	<p>The name of the data connection to an ARIS Connect server configured in the <b>ArisApiConfig</b> XML element of the ARIS - Alfabet Interoperability Interface in Alfabet Expand. Multiple data connections can be configured to distinguish For example, for data relevant for the integration of different data types.</p> <p>For more information about the configuration of data connections see <a href="#">Step 1: Configuring the Connection to ARIS Connect in the XML Object ArisApiConfig</a> in the section <a href="#">Configuring the Alfabet Web Application to Read Data from ARIS</a>.</p>	50 characters	String
ARIS_DBCONNECTION	<p>The name of the ARIS database connection for access to the ARIS database the data is imported from. Multiple ARIS database connections can be created for one data connection to an ARIS Connect server to import data from multiple ARIS databases.</p> <p>For more information about the configuration of ARIS database connections see <a href="#">Step 1: Configuring the Connection to ARIS Connect in the XML Object ArisApiConfig</a> in the section <a href="#">Configuring the Alfabet Web Application to Read Data from ARIS</a>.</p>	50 characters	String
GUID	The ARIS GUID of the model. The GUID unambiguously identifies the object in the ARIS database.	40 characters	String
APINAME	The ARIS API name of the model.	36 characters	String
TYPENAME	The API name of the ARIS model type.	128 characters	String
GUILINK	The URL of the model diagram on the ARIS Connect server.	1024 characters	String
NAME	The name of the model.	128 characters	String

Database column Name	Content	Truncated to:	Data Type
DESCRIPTION	The description defined for the model.	1024 characters	String
CREATOR	The ARIS user that created the model.	50 characters	String
PERSONRESPONSIBLE	The ARIS user that is responsible for the model.	50 characters	String
LASTUSER	The ARIS user that last changed the model.	50 characters	String
TIMEOFGENERATION	The generation date of the model.		DateTime
LASTCHANGE	The date when the model was last changed in ARIS.		DateTime

#### Temporary Table Containing Objects

**Table name :** TMP\_OBJECT

The table has the following columns:

Database column Name	Content	Truncated to:	Data Type
DATACONNECTIONNAME	<p>The name of the data connection configured in the <code>Arise-ApiConfig</code> XML element of the ARIS - Alfabet Interoperability Interface in Alfabet Expand. Multiple data connections can be configured to distinguish For example, for data relevant for the integration of different data types.</p> <p>For more information about the configuration of data connections see the section <a href="#">Configuration of the ARIS - Alfabet Interoperability Interface to Perform Customized Data Import</a>.</p>	50 characters	String
ARIS_DBCONNECTION	<p>The name of the ARIS database connection for access to the ARIS database the data is imported from. Multiple ARIS database connections can be created for one data connection to an ARIS Connect server to import data from multiple ARIS databases.</p> <p>For more information about the configuration of ARIS database connections see <a href="#">Step 1: Configuring the Connection</a></p>	50 characters	String

Database column Name	Content	Truncated to:	Data Type
	<a href="#">to ARIS Connect in the XML Object ArisApiConfig</a> in the section <a href="#">Configuring the Alfabet Web Application to Read Data from ARIS</a> .		
MODELGUID	The ARIS GUID of the model the object data is taken over from.	40 characters	String
GUID	The ARIS GUID of the ARIS object. The GUID unambiguously identifies the object in the ARIS database.	40 characters	String
OCCID	The model specific ARIS ID of the object.	40 characters	String
APINAME	The ARIS API name of the object.	36 characters	String
TYPENAME	The API name of the ARIS object type the object belongs to.	128 characters	String
APILINK	The RESTful API URL of the object.	1024 characters	String
SYMBOLNAME	The name of the symbol that is defined for the object.	128 characters	String
SYMBOL_APINAME	The API name of the symbol that is defined for the object.	36 characters	String
NAME	The name of the object.	128 characters	String
DESCRIPTION	The description defined for the object.	1024 characters	String
CREATOR	The ARIS user that created the object.	50 characters	String
PERSONRESPONSIBLE	The ARIS user that is responsible for the object.	50 characters	String
LASTUSER	The ARIS user that last changed the object.	50 characters	String



Database column Name	Content	Truncated to:	Data Type
TIMEOFGENERATION	The generation date of the object.		DateTime
LASTCHANGE	The date when the object was last changed in ARIS.		DateTime

#### Temporary Table Containing Connections

**Table name :** TMP\_RELATIONSHIP

The table has the following columns:

Database column Name	Content	Truncated to:	Data Type
DATACONNECTIONNAME	<p>The name of the data connection configured in the <b>ArisApiConfig</b> XML element of the ARIS - Alfabet Interoperability Interface in Alfabet Expand. Multiple data connections can be configured to distinguish For example, for data relevant for the integration of different data types.</p> <p>For more information about the configuration of data connections see the section <a href="#">Configuration of the ARIS - Alfabet Interoperability Interface to Perform Customized Data Import</a>.</p>	50 characters	String
ARIS_DBCONNECTION	<p>The name of the ARIS database connection for access to the ARIS database the data is imported from. Multiple ARIS database connections can be created for one data connection to an ARIS Connect server to import data from multiple ARIS databases.</p> <p>For more information about the configuration of ARIS database connections see <a href="#">Step 1: Configuring the Connection to ARIS Connect in the XML Object ArisApiConfig</a> in the section <a href="#">Configuring the Alfabet Web Application to Read Data from ARIS</a>.</p>	50 characters	String
MODELGUID	The ARIS GUID of the model the connection data is taken over from.	40 characters	String
OCCID	The model specific ARIS ID of the connection.	40 characters	String

Database column Name	Content	Truncated to:	Data Type
APINAME	The ARIS API name of the connection.	36 characters	String
TYPENAME	The API name of the ARIS connection type the connection belongs to.	128 characters	String
SOURCE_GUID	The GUID of the source object of the connection.	36 characters	String
TARGET_GUID	The GUID of the target object of the connection.	36 characters	String
SOURCE_LINK	The RESTful API URL of the source object of the connection.	1024 characters	String
TARGET_LINK	The RESTful API URL of the target object of the connection.	1024 characters	String
SOURCE_OCCID	The model specific ARIS ID of the source object of the connection.	40 characters	String
TARGET_OCCID	The model specific ARIS ID of the target object of the connection.	40 characters	String

#### Temporary Table Containing Assignments

**Table name :** TMP\_ASSIGNMENT

The table has the following columns:

Database column Name	Content	Truncated to:	Data Type
DATACONNECTIONNAME	<p>The name of the data connection configured in the <b>ArisApiConfig</b> XML element of the ARIS - Alfabet Interoperability Interface in Alfabet Expand. Multiple data connections can be configured to distinguish For example, for data relevant for the integration of different data types.</p> <p>For more information about the configuration of data connections see the section <a href="#">Configuration of the ARIS - Alfabet Interoperability Interface to Perform Customized Data Import</a>.</p>	50 characters	String

Database column Name	Content	Truncated to:	Data Type
ARIS_DBCONNECTION	<p>The name of the ARIS database connection for access to the ARIS database the data is imported from. Multiple ARIS database connections can be created for one data connection to an ARIS Connect server to import data from multiple ARIS databases.</p> <p>For more information about the configuration of ARIS database connections see <a href="#">Step 1: Configuring the Connection to ARIS Connect in the XML Object ArisApiConfig</a> in the section <a href="#">Configuring the Alfabet Web Application to Read Data from ARIS</a>.</p>	50 characters	String
MODELGUID	The ARIS GUID of the model the contains the source object of the assignment.	40 characters	String
PARENTGUID	The ARIS GUID of the source object the assignment.	40 characters	String
APINAME	The ARIS API name of the object.	36 characters	String
TYPENAME	The API name of the ARIS object type the object belongs to.	128 characters	String
TARGET_GUID	The ARIS GUID of the target model of the assignment.	40 characters	String
APILINK	The RESTful API URL of the source object of the assignment.	128 characters	String
GUILINK	The URL of the assignment on the ARIS Connect server.	128 characters	String
TARGETMODELNAME	The name of the target model of the assignment.	128 characters	String

#### Temporary Table Containing Attributes

**Table name :** TMP\_ATTRIBUTE

The table has the following columns:

Database column Name	Content	Truncated to:	Data Type
DATACONNECTIONNAME	<p>The name of the data connection configured in the <b>AriseConfig</b> XML element of the ARIS - Alfabet Interoperability Interface in Alfabet Expand. Multiple data connections can be configured to distinguish For example, for data relevant for the integration of different data types.</p> <p>For more information about the configuration of data connections see the section <a href="#">Configuration of the ARIS - Alfabet Interoperability Interface to Perform Customized Data Import</a>.</p>	50 characters	String
ARIS_DECONNECTION	<p>The name of the ARIS database connection for access to the ARIS database the data is imported from. Multiple ARIS database connections can be created for one data connection to an ARIS Connect server to import data from multiple ARIS databases.</p> <p>For more information about the configuration of ARIS database connections see <a href="#">Step 1: Configuring the Connection to ARIS Connect in the XML Object AriseConfig</a> in the section <a href="#">Configuring the Alfabet Web Application to Read Data from ARIS</a>.</p>	50 characters	String
PARENTGUID	The ARIS GUID of the object the attribute value is defined for. If the attribute is defined for a connection, the PARENTGUID is the OCCID of the connection.	40 characters	String
ID	The ARIS ID number of the attribute.	30 characters	String
TYPENAME	The API name of the ARIS attribute type the attribute belongs to.	128 characters	String
VALUE	The value defined for the attribute for the object defined with PARENTGUID.		Text
APINAME	The ARIS API name of the attribute.	36 characters	String
LANGUAGE	The language for that the value in the column VALUE is defined. If a value is defined in multiple languages, the database table contains one row for each language.	10 characters	String
KIND	specifies which kind of parent object the attribute is assigned to. KIND can be Assignment, Object or Model.	15 characters	String

## Considerations for Data Import

The integration of ARIS data into the Alfabet database is usually a constant process that is performed in regular intervals. Therefore it is important that information about which database the object is imported from (that means which ARIS database connection was executed) and which ARIS source object the Alfabet object is based on is stored in the Alfabet database to be able to map the objects in both databases during import.

It is further recommended to store the date of the last update of the Alfabet object via an integration object in the database table for the Alfabet object class that is the target of the integration. The integration can then be limited to import of current changes that are not already integrated.

For the Alfabet object classes Business Process Model and Business Process which are updated with ARIS data during standard integration, the ARIS GUID of the ARIS object the Alfabet object is derived from is stored in a standard object class property `ARIS_GUID`. The name of the ARIS database connection is stored in a standard property `ARIS_DBCONNECTION`. The time of the last integration is stored in a standard property `ARIS_LAST_CHANGE` of the object class business process model.

It is recommended to define corresponding custom object class properties for each object class that shall be included in the customized data integration process and store the information about the ARIS GUID of the ARIS source object and the last integration date in these custom properties.



For information about the definition of custom object class properties see the chapter *Configuring the Class Model* of the reference manual *Configuring Alfabet with Alfabet Expand*.



## Index

ADIF import scheme	
ARISIntegration_ImportFromARIS	44
Alfabet configured report	37
Alfabet database	
temporary database tables	46
Alfabet Integration	
ARIS properties	23
Alfabet user interface	
configuring	35
Alfabet Web Application	
configuring RESTful services	23
AMM file	20
API password	
generating for Alfabet user	22
ARIS Connect Server	
configuring authorization	23
configuring RESTful services	21
ARIS database version	34
ARIS diagram link	
creating	17
in configured report	18
in dashboard	18
updating	18
ARIS Diagrams page view	11
ARIS function	
creating Alfabet business process	11
ARIS model diagram	
integrating changes to Alfabet	14
integrating in Alfabet business process model	11
ARIS-ALFABET-mapping-V2.XML	37
ArisApiConfig	
connection to ARIS	23
data mapping	29
navigation	31
ArisGenericDataConnectivity	40
ARISIntegration_ImportFromARIS	
generated temporary tables	46
attribute group	
Alfabet Integration	23
attribute mapping	

---

Alfabet import to ARIS	37
authorization	
from Alfabet to ARIS	23
from ARIS to Alfabet	22
business process model	
ARIS Diagrams page view	11
building from ARIS data	12
configuring integration	29
integrating current changes from ARIS	14
integrating from ARIS	11
standard mapping of ARIS data	11
Business Process Model ARIS Source Report	14
change list	
ARIS database	34
class mapping	
Alfabet import to ARIS	37
configured report	
Alfabet	37
navigation to ARIS diagram	18
customization	
data import from ARIS	39
Import of Alfabet data to ARIS	37
dashboard	
navigation to ARIS diagram	18
data exchange	
from Alfabet to ARIS	10
from ARIS to Alfabet	10
data mapping	
from ARIS to Alfabet	29
database table	
temporary for ARIS assignments	52
temporary for ARIS attributes	53
temporary for ARIS connections	50
temporary for ARIS models	47
temporary for ARIS objects	48
documentation	7
Enable REST API v2	
for Alfabet user	22
in the server alias	20
Enterprise Architecture Extension	20
functionality	
RESTful service based	10
implementation	
RESTful service based ARIS - Alfabet Interoperability	19
licenses	20
link	



---

configuring	31
to the ARIS user interface	17
navigating	
from Alfabet to ARIS	17
navigation	
configuring	31
preconditions	
RESTful service based ARIS - Alfabet Interoperability	20
RESTful service	
customization	39
RESTful services	
Alfabet authentication data	22
configuring connection from Alfabet to ARIS	23, 29
configuring connection from ARIS to Alfabet	21
configuring navigation from Alfabet to ARIS	31
Self-signed Certificates	21
solution business process planning	14
temporary database table	46
ARIS assignments	52
ARIS attributes	53
ARIS connections	50
ARIS models	47
ARIS objects	48
terminology	8
version	
ARIS database	34