



## **ALFABET REFERENCE MANUAL**

# **ARIS - Alfabet Interoperability Interface**

---

Documentation Version Alfabet 9.10.1

Copyright © 2013 - 2016 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).

---

## 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>6</b>
Related Documents	9
Documents Relevant for the Alfabet Application	10
Documents Relevant for the ARIS Application	10
Terminology in ARIS and Alfabet	11
<b>Chapter 2: Functionality Provided By the RESTful Service Based Interface</b>	<b>13</b>
Integrating ARIS Business Process Model Planning Data Into Alfabet	14
Creating an Alfabet Business Process Model on Basis of an ARIS Business Process Model	15
Integrating Changes To an ARIS Business Process Model into the Existing Dependent Alfabet Business Process Model	16
Navigation From the Alfabet to the ARIS User Interface	19
Defining a New Link to an ARIS Diagram	19
Opening a Link to an ARIS Diagram	19
Updating the Information About ARIS Diagrams Displayed in the Table	20
<b>Chapter 3: Activating and Configuring the RESTful Service Based Interface</b>	<b>21</b>
Required Licenses	22
Configuring the ARIS Connect Server to Read Data from Alfabet	22
Required Configuration on Alfabet Side	22
Required Configuration on ARIS Side	23
Configuring the Alfabet Web Application to Read Data from ARIS	24
Using Server Variables in the XML Object ArisApiConfig	25
Configuring the Mapping of Data for Business Process Model Integration	26
Configuring the Navigation to the ARIS User Interface	28
Making the Views for the ARIS - Alfabet Interoperability Interface Available on the Alfabet User Interface	29
<b>Chapter 4: Customer Configured Integration of ARIS Data into Alfabet</b>	<b>31</b>
Configuration of the ARIS - Alfabet Interoperability Interface	32
Configuration of the ADIF Import Scheme	35
Temporary Tables Created During Import	36
Considerations for Data Import	43
<b>Chapter 5: The Web Service Based Version of the ARIS - Alfabet Interoperability Interface</b>	<b>44</b>
Functionality Provided by the ARIS - Alfabet Interoperability Interface	44
Data Exchange between the ARIS and Alfabet Databases	45
Navigation between the ARIS and Alfabet User Interfaces	48
Interface Architecture	50
Basic Initial Set-Up of the Web Service based ARIS - Alfabet Interoperability Interface	51
Installation Requirements	51
Installation	52
Configuration	53
Running the Web Service Based ARIS - Alfabet Interoperability Interface	59
Command Line Parameters for Logging	61
Advanced Set-Up: Customizing Data Transfer from Alfabet to ARIS	62
The XML for Data Transfer From Alfabet to ARIS	64

---

The XML for Data Transfer From ARIS to Alfabet	67
<b>Index</b>	<b>71</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.

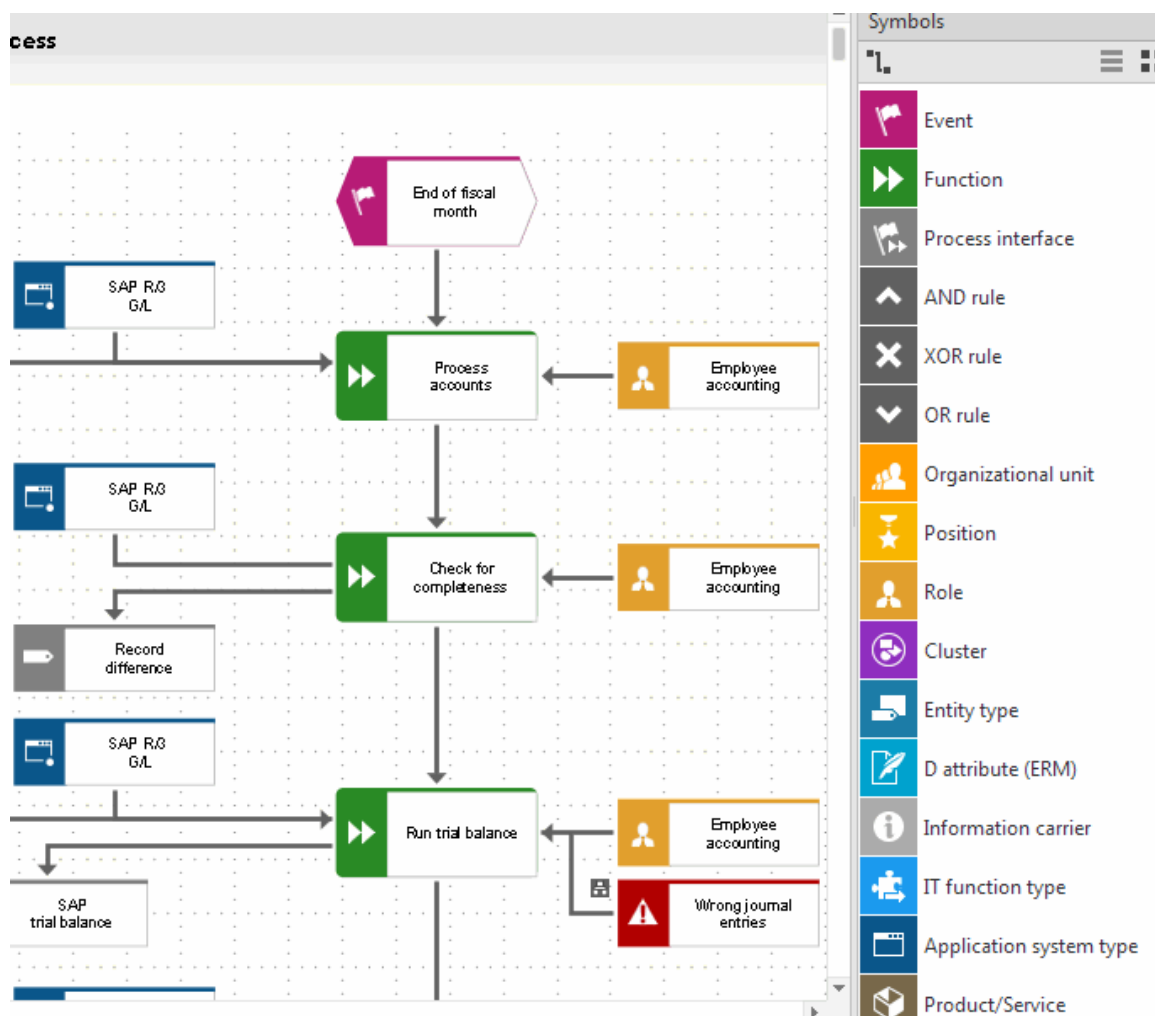


FIGURE 1: ARIS EPC diagram of business process including applications (blue) used to execute functions (green)

Alfabet focuses on the role of the IT landscape to support 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.

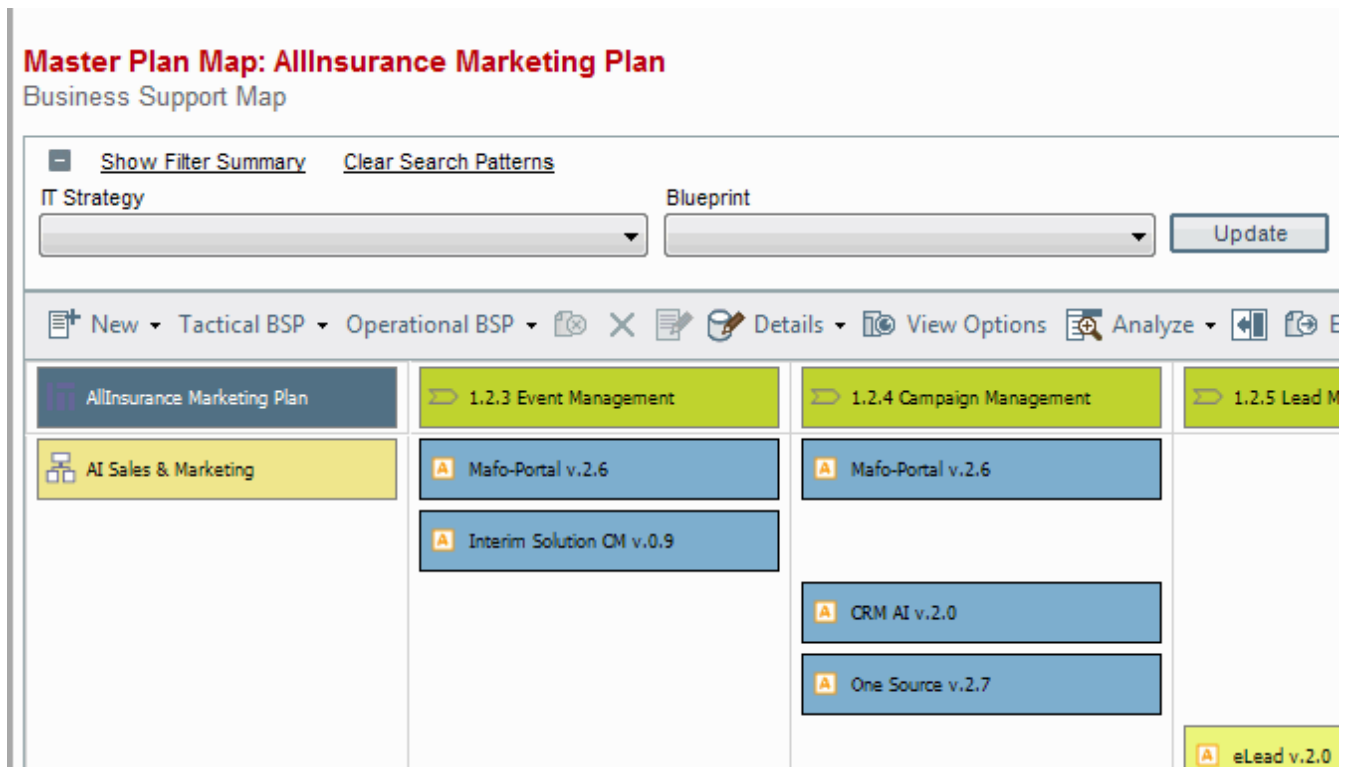


FIGURE 2: Master Planning in Alfabet with supporting applications (blue) displayed per organization (yellow) and business process (green)

Interoperability between the two solutions is now available that 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.

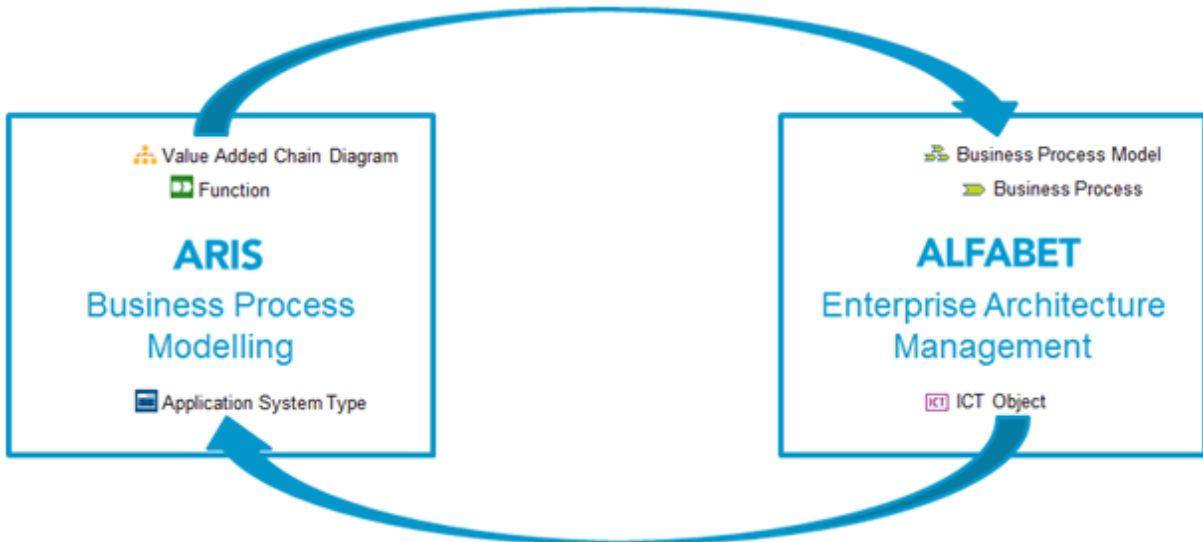


FIGURE 3: Standard Data exchange configured for the ARIS - Alfabet Interoperability Interface

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.

Two versions of the ARIS - Alfabet Interoperability Interface are currently available. One is based on Web Services and the other, more advanced version, is based on a RESTful service architecture. The following table lists the main differences between the two versions of the ARIS - Alfabet Interoperability Interface as a basis for decision which version to use. In general, it is recommended to use the version based on a RESTful service architecture, because it is easier to implement and offers advanced handling of data exchange via the user interfaces of the applications.

	RESTful API based ARIS - Alfabet Interoperability Interface	Web Service based ARIS - Alfabet Interoperability Interface
Availability	Alfabet 9.9	Alfabet 9.6 and higher
Involved components	RESTful API at both ARIS and Alfabet	ARIS reports ARIS Web Services ARIS - Alfabet Interoperability Interface batch tool Alfabet Web Services Alfabet Data Integration Framework (ADIF)
Data integration mode	Batch and per object at runtime via the user interfaces	Batch update via a batch tool
Business pro-	Existing ARIS business process mod-	Existing ARIS business process models



	<b>RESTful API based ARIS - Alfabet Interoperability Interface</b>	<b>Web Service based ARIS - Alfabet Interoperability Interface</b>
Process model integration from ARIS to Alfabet	<p>Models are directly integrated to Alfabet as business process models.</p> <p>The hierarchy of business processes is derived from relations set in ARIS or alternatively from the hierarchy of diagrams in ARIS.</p> <p>Any process model can be used for integration.</p> <p>Changes to the ARIS business process model already integrated to Alfabet are displayed to the Alfabet user on the Alfabet user interface and the user can directly take over changes for the Alfabet business process model.</p>	<p>Models are integrated via solution business process models into Alfabet. After manual control the solution business process model is then checked in to the Alfabet database as business process model.</p> <p>The hierarchy of business processes is derived from the hierarchy of diagrams in ARIS.</p> <p>Only business process models of the type Value Chain Diagram can be used for integration.</p> <p>Changes to the ARIS business process model already integrated to Alfabet are integrated to Alfabet via solution business process planning in parallel to the methods used for new business process models.</p>
Architecture element integration from Alfabet to ARIS	Integration is performed either by pushing data of individual objects via the Alfabet user interface to ARIS or by performing update per object or batch update via the ARIS Connect interface.	By default, Alfabet applications are integrated via a batch job. Integration of data for other object classes requires configuration.
Links between the user interface	Users can navigate between the user interfaces of Alfabet and ARIS via links that are created per object by the user.	Users can navigate between the user interfaces of Alfabet and ARIS via links that are created during integration of object data for all integrated objects.

This document describes the functionality provided by the integration interface between the ARIS and Alfabet applications, the installation of the interface including the required configuration, and the handling of the interface components. The document is focused on the new RESTful service architecture based ARIS - Alfabet Interoperability Interface. Nevertheless, the complete documentation of the Web Service based ARIS - Alfabet Interoperability Interface is included in a separate chapter [The Web Service Based Version of the ARIS - Alfabet Interoperability Interface](#).

## 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
Alfabet Online Help	Working with business process models and solution business process models in Alfabet.	Yes	Yes

## Documents Relevant for the ARIS 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>ARIS Server Installation and Administration Guide</i>	Includes the ARIS part of the RESTful API based ARIS - Alfabet Interoperability Interface	Yes	-
Online Help <i>ARIS</i> and <i>ARISScript</i>	Configuring reports in ARIS.	-	Yes
Reference Manual <i>ARISMethods</i>	Business Process Planning in ARIS.	Yes	Yes

## 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
Report	In ARIS, reports are JavaScript based and either export data in a customized format to, for example, Excel, Word or XML, or perform actions on the object data. A set of preconfigured reports exists that can be customized. Customers can define their own reports as needed.  In Alfabet, reports are either tabular or graphic representations of data displayed in the Alfabet user interface. Customers can define their own reports as needed. In terms of the ability to batch export and import data, the Alfabet functionality that corresponds to ARIS Reports is the Alfabet Data Integration Interface (ADIF). ADIF allows import and export of data to be performed from/to Excel, Word or XML files, or from/to external databases. The export and import is based on customer -defined native SQL queries. These are combined with ADIF mechanisms that, for example, ensure database integrity during import.	ADIF
Business Process	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	Business Process Model

ARIS	Definition	Alfabet
Model	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.	
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 Provided By the RESTful Service Based 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 RESTful Service Based 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 into 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.

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 RESTful Service Based 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 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) Enter search criteria into the **Search Pattern** field to find the relevant ARIS business process diagram and click the **Search** button.
- 9) Select the ARIS diagram from the list of search results and click **OK** to start the integration process. Business processes are created automatically from *ARISfunctions* 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.

## 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) 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.

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

## 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**.

## Chapter 3: Activating and Configuring the RESTful Service Based 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.

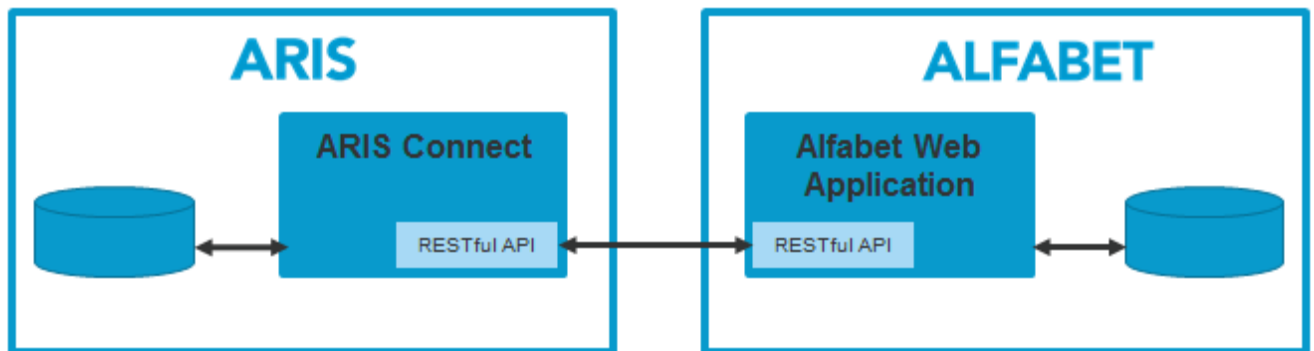


FIGURE 4: Overview of the components involved in data exchange via the ARIS - Alfabet Interoperability Interface

The configuration of the ARIS application is described in the reference manual *ARIS Server Installation and Administration Guide* provided with 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 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](#)
  - [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](#)
- [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.

In addition, the Enterprise Architecture Extension for ARIS has to be licensed and installed.

Authorization of the requests between the ARIS and Alfabet components require a license key that includes certificates for all involved components. The license keys are delivered by Software AG Support.

## 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 unique key provided by Software AG. The unique key can be used in two different ways for authorization against the Alfabet Web Application:

- The key obtained from Software AG is used as is for each API call to the Alfabet Web Application.
- Authorization is performed on a per user basis with a different key per user.

## Required Configuration on Alfabet Side

The unique key provided by Software AG must be included into the web.config file of the Alfabet Web Application into the alfaSection XML element:

- 1) Go to the physical directory of the Alfabet Web Application.
- 2) Open the web.config file in a text editor.
- 3) Add the following as child element of the alfaSection XML element in the file:

```
<add key="rest_token_ro" value="UniqueKeyValue" />
```



For example:

```
<alfaSection>
  <add key="rest_token_ro" value="72GY6SWS5LSUH3OAVDH4HDYQQQRWZPGU" />
  ...
</alfaSection>
```

The implementation of a key per user authorization requires an additional configuration to be performed with the configuration tool Alfabet Expand:

- 1) In the **ADIF** tab of Alfabet Expand, right-click the root node and select **Generate User REST API V.1 Token**. A new window opens.



The same functionality is available in the main menu of Alfabet Expand under **Managers > ARIS Interoperability > Generate User Token**.

- 2) Select the user name from the drop down list in the field **Select User**.

- 3) Click **Generate Token**. A code is generated, stored in the Alfabet database and displayed in the field **Generated Token**.
- 4) Copy the code and store the information about user name and code for use on ARIS side. You can use the **Copy to Clipboard** button to copy the token to the clipboard.

The generated code must be used for authorization in combination with the user name.

## Required Configuration on ARIS Side

The unique key for authorization against the Alfabet Web Application or the user based code in combination with the user name must be added to the definition of the ARIS - Alfabet Interoperability Interface on ARIS side:

- 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-Integration.xml**.
- 5) Replace the content with the following code:

```
<mapping>
  <user name="AlfabetUserName" token="KeyValue" accessmode="httpclient"/>
  <object aristype="OT_APPL_SYS_TYPE" table="APPLICATION" >
    <attr aris="AT_NAME" />
    <attr aris="AT_DESC" />
    <attr aris="AT_REL_3" />
    <link attribute="AT_ALFA_LINK_1" title="AT_ALFA_NAVI1" defaultvalue="Navigate
to Application in ALFABET" >
    <title locale="1033" value="Navigate to Application in ALFABET" />
  </link>
</object>
</mapping>
```

- 6) In the user XML element, set the following values for the name and token XML attributes:

- For authorization via the unique key provided by Software AG, the name XML attribute must be empty and the unique key must be entered into the token XML attribute.



```
<user name="" token="72GY6SWS5LSUH3OAVDH4HDYQQQRWZPGU"
accessmode="httpclient"/>
```

- For authorization via the user based key, enter the user name into the name XML attribute and the user key into the token XML attribute.



```
<user name="Customer" token="POWQCVE5BOJJPWZTJ4F3ARRP7O454ESB"
accessmode="httpclient"/>
```

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

- 1) Go to the attribute group **ALFABET**.

- 2) Enter the URL of the Alfabet Web Application into the following properties:
  - **ALFABET – Web service URL**
  - **ALFABET – User interface URL**

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. It includes the following, as well as configuration of mapping of


- Configuration of the connection to ARIS, 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.
- 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) Configure the connection to the ARIS RESTful API by setting the following XML attributes of the XML element **ArisApiConfig** 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](#).

- **active**: Set to true to activate the connection to the defined RESTful API.
- **service**: Enter the URL to ARIS Connect.
- **api**: Enter /abs/api. This attribute specifies the path to the ARIS RESTful API starting from the URL specified with the attribute **service**.
- **auth\_path**: Enter the path to the ARIS API. This is usually /umc/api/tokens.
- **auth\_tenant**: Enter the tenant for authorization with the ARIS API (for example "default").
- **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 build-in standard license key is used.
- **aris\_db**: Enter the name of the ARIS database that includes the data relevant for data integration.
- **method\_filter**: Enter the name of the filter that shall be applied when accessing the ARIS database (for example "Entire Method").



- **data\_portion**: Enter the maximum number of data sets that shall be exchanged within one transmission during the call. The default value is 50.
  - **search\_limit**: Enter the maximum number of relevant objects to be retrieved from the ARIS database for a request. The default value is 200.
  - **link\_view**: Enter "item", if the view shall open in ARIS Connect and "thinclient" if the link shall open in ARIS Architect.
  - **link\_itemtype**: Enter "default". This entry provides a mandatory technical term for that currently no other choices are implemented.
  - **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).
- 4) Make sure that the XML element ArisApiConfig contains the following child element:
- ```
<ClassMapping class="ARIS_DiagramLink">
  <AttributeMapping aris_attr="status" alfa_attr="ARIS_STATUS" />
</ClassMapping>
```
- 5) 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](#).
- 6) 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](#).
- 7) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.



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](#).

## 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.

Only complete attribute values can be substituted with a server variable.

Server variables are defined in the Alfabet Administrator:

- 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, enter the value that shall be defined with the server variable.



If the variable value contains a special character according to XML standards (for example: &, %, ;, <, > ), these characters must be replaced by their XML conformant variant (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 XML object ArisApiConfig specified in Alfabet Expand.

A server variable is referenced in the XML attribute as \$<server variable name>. For example, a server variable called SQLSERVER is referenced as \$SQLSERVER.

A value for an XML attribute can either be defined as 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. .

## 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 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 **ArisApiConfig** XML element:
 

```
<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\_PRCS\_ORNT\_SUPER) and ARIS *functions* (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 ARIS *functions* to Alfabet business processes applies:

```
<ArisApiConfig ...>
  <ClassMapping class="BusinessProcessModel">
    <ImportMapping model_type="MT_VAL_ADD_CHN_DGM"
      parent_rel="CT_IS_PRCS_ORNT_SUPER"
      proc_type="OT_FUNC,OT_PROC"></ImportMapping>
    <ImportMapping model_type="MT_FUNC_TREE"
      parent_rel="CT_CONS_OF_1"></ImportMapping>
  </ClassMapping>
</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 **ArisApiConfig** 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.
- 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 ...>
  <ClassMapping class="BusinessProcess">
    <AttributeMapping aris_attr="status" alfa_attr="ARIS_STATUS" />
  </ClassMapping>
</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.

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 **ArisApiConfig** XML element with the 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 ...>
  <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>
</ArisApiConfig>
```

## 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_RootProcesses	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</li> </ul>

View Name	View Caption	Available for Alfabet Object Class	Functionality for ARIS - Alfabet Interoperability Interface	
				assigned.
ARIS_BMPSource_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 ARIS Data into 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 *ArisApiConfig* 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.

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.
- The predefined ADIF scheme *Alfabet\_Standard\_Import\_from\_ARIS* that is part of the predefined ADIF configuration in Alfabet Expand and located in the sub-folder **Alfabet Standard Imports** writes the data returned by the RESTful services into temporary database tables in the Alfabet database. The predefined ADIF scheme *Alfabet\_Standard\_Import\_from\_ARIS* 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 predefined ADIF scheme *Alfabet\_Standard\_Import\_from\_ARIS* must be executed to perform update of data in the Alfabet database. *ARISIntegration\_ImportFromARIS* both triggers the execution of the RESTful service request and changes the data in the Alfabet database according to the provided information.

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




## Configuration of the ARIS - Alfabet Interoperability Interface

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, relevant 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 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.
- 7) Enter a child XML element **ModelFilter** to the XML element **DataConnection**. The XML element **DataConnection** can only have one child element **ModelFilter**.
- 8) 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
```

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

10) 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.

- 11) Optionally, enter a child XML element **ModelGroupNames** to the XML element **ModelFilter**. The XML element **ModelFilter** can only have one child element **ModelGroupNames**.
- 12) 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**. The definition of the group filters is optional. If no group filter is defined, models are imported independent from the group they belong to.

- 13) 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.

- 14) 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.
- 15) 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.
- 16) In the toolbar of Alfabet Expand, click the **Save**  button to save your changes.

## Configuration of the ADIF Import Scheme

The preconfigured ADIF scheme **Alfabet\_Standard\_Import\_from\_ARIS** 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 import scheme **Alfabet\_Standard\_Import\_from\_ARIS** without prior configuration of the import logic will not lead to any changes in the Alfabet database.
- The temporary database tables are deleted.

To trigger data integration from ARIS diagrams to defined object class tables in the Alfabet database the ADIF import scheme **Alfabet\_Standard\_Import\_from\_ARIS** must be amended with the customer configured import of data from the temporary database tables to the standard database tables of the Alfabet meta-model. The ADIF import scheme **Alfabet\_Standard\_Import\_from\_ARIS** 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:

- [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](#)

## Temporary Tables Created During Import

The ADIF import scheme **Alfabet\_Standard\_Import\_from\_ARIS** 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 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</a></p>	50 characters	String

Database column Name	Content	Truncated to:	Data Type
	<a href="#">Interface</a> .		
GUID	The ARIS GUID of the model. The GUID unambiguously identifies the object in the ARIS database.	36 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
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 ArisApiConfig 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</a>.</p>	50 characters	String
MODELGUID	The ARIS GUID of the model the object data is taken over from.	36 characters	String
GUID	The ARIS GUID of the ARIS object. The GUID unambiguously identifies the object in the ARIS database.	36 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

Database column Name	Content	Truncated to:	Data Type
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 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 Connections****Table name:** TMP\_RELATIONSHIP

The table has the following columns:

Database column Name	Content	Truncated to:	Data Type
DATACONNECTIONNAME	The name of the data connection configured in the <b>AriseApiConfig</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.  For more information about the configuration of data connections see the section <a href="#">Configuration of the ARIS - Alfabet Interoperability Interface</a> .	50 characters	String
MODELGUID	The ARIS GUID of the model the connection data is taken over from.	36 characters	String
OCCID	The model specific ARIS ID of the connection.	40 characters	String
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



Database column Name	Content	Truncated to:	Data Type
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	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.  For more information about the configuration of data connections see the section <a href="#">Configuration of the ARIS - Alfabet Interoperability Interface</a> .	50 characters	String
MODELGUID	The ARIS GUID of the model the contains the source object of the assignment.	36 characters	String
PARENTGUID	The ARIS GUID of the source object the assignment.	36 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.	36 characters	String

Database column Name	Content	Truncated to:	Data Type
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>AriseApiConfig</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</a>.</p>	50 characters	String
PARENTGUID	The ARIS GUID of the object the attribute value is defined for.	36 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

Database column Name	Content	Truncated to:	Data Type
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 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 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*.

## Chapter 5: The Web Service Based Version of the ARIS - Alfabet Interoperability Interface

The Web Service based ARIS - Alfabet Interoperability Interface was developed prior to the ARIS - Alfabet Interoperability Interface based on a RESTful service architecture. It is available for Alfabet versions 9.6 and higher and requires a more complex configuration and handling as the successor version based on a RESTful service architecture. Data integration is done by a batch job independent from user interaction on the user interfaces.

The following information provides an overview of the functionality and technical architecture of the ARIS - Alfabet Interoperability Interface:

- [Functionality Provided by the ARIS - Alfabet Interoperability Interface](#)
  - [Data Exchange between the ARIS and Alfabet Databases](#)
    - [Application Data Integration from Alfabet to ARIS](#)
    - [Business Process Model Integration from ARIS to Alfabet](#)
  - [Navigation between the ARIS and Alfabet User Interfaces](#)
- [Interface Architecture](#)
- [Basic Initial Set-Up of the Web Service based ARIS - Alfabet Interoperability Interface](#)
  - [Installation Requirements](#)
  - [Installation](#)
  - [Configuration](#)
    - [Configuring the ARIS Data for Import to Alfabet](#)
    - [Configuring the Web Service based ARIS - Alfabet Interoperability Interface](#)
- [Running the Web Service Based ARIS - Alfabet Interoperability Interface](#)
  - [Command Line Parameters for Logging](#)
- [Advanced Set-Up: Customizing Data Transfer from Alfabet to ARIS](#)
  - [The XML for Data Transfer From Alfabet to ARIS](#)
    - [Links from ARIS to Alfabet](#)
  - [The XML for Data Transfer From ARIS to Alfabet](#)
    - [Links from Alfabet to ARIS](#)

### Functionality Provided by the ARIS - Alfabet Interoperability Interface

The way data exchange and switching between the user interfaces of ARIS and Alfabet at runtime are described in the following:

- [Data Exchange between the ARIS and Alfabet Databases](#)
  - [Application Data Integration from Alfabet to ARIS](#)
  - [Business Process Model Integration from ARIS to Alfabet](#)

- [Navigation between the ARIS and Alfabet User Interfaces](#)

## Data Exchange between the ARIS and Alfabet Databases

The Web Service based ARIS - Alfabet Interoperability Interface integrates changes to business process models performed in ARIS with the Alfabet database as well as changes made to the application architecture in Alfabet with the ARIS database. Data integration is performed via a batch job that is run by a system administrator and therefore decoupled from user interaction on the ARIS or Alfabet user interface.



The data integration does not include any mechanisms that prevent changes to application data in ARIS or the business process model in Alfabet. Any changes performed manually in ARIS to objects of the application system type or in Alfabet to objects of the object class Business Process Model or Business Process will be lost during data import via the Web Service based ARIS - Alfabet Interoperability Interface.

For example, a business process added to a business process model in Alfabet will be deleted at the next import if the business process does not correspond to a *function* in a Value Added Chain diagram in ARIS. The same applies to an object of the *application system type* defined in ARIS used in a business process model with no corresponding application in Alfabet.

To address the problems that may result by objects being created or edited in the wrong place, the import mechanisms on both ARIS and Alfabet sides includes mechanisms to check and correct the use of these objects prior to deletion.

### Application Data Integration from Alfabet to ARIS

The Web Service based ARIS - Alfabet Interoperability Interface reads data about all applications from the Alfabet database.

For each application, the following object class properties are transferred to ARIS:

- Name
- Description
- GUID
- The Alfabet view that opens if a user directly accesses the Alfabet interface via the object.



For more information about navigation between the ARIS and Alfabet user interfaces, see the section [Navigation between the ARIS and Alfabet User Interfaces](#).

On the ARIS side, the data is merged with the existing data in the database. The merge is based on two interface-specific ARIS attributes:

- **Alfabet - GUID:** Stores the GUID of the application in the Alfabet database as an attribute of the *application system type* object.
- **Alfabet - Deletion Candidate:** This boolean attribute is set to False per default. It is set to True for all objects that are deleted in Alfabet and therefore should also be deleted in ARIS.

During import, the existence of an object of the *application system type* with an attribute ALFA\_GUID identical to the GUID of the imported application is checked. The action that will be performed during the import depends on the result of the check:

ALFA_GUID in Import Data	ALFA_GUID in Database	Action in the ARIS database
yes	yes	The attributes of the object are updated with the import data. The object is assigned to the sub-group <b>active</b> of the <b>Main group</b> .
yes	no	A new object is created with the attributes from the import data and the object is assigned to the sub-group <b>active</b> of the <b>Main group</b> .
no	yes	The attribute <b>Alfabet- Deletion Candidate</b> is set to True and the object is assigned to the sub-group <b>delete</b> of the <b>Main group</b> . It is no longer assigned to the sub-group <b>active</b> .

Two groups were added to the **Main group** for the management of objects managed via import from Alfabet:

- **active**: This group contains all objects of the *application system type* that are available both in the Alfabet and ARIS database.
- **delete**: This group contains all objects of the *application system type* that are no longer available in the Alfabet database but are still part of the ARIS database.

Both groups contain an Application System Type diagram that include all *application system type* objects in the group. In ARIS, each object needs to be included in at least one model or diagram to be editable. The Application System Type diagram ensures editability of all objects imported via Alfabet.

After data import, an ARIS user can check the **delete** group to check whether the applications in the group are used in any models and substitute them with another *application system type* object in each of the impacted models. After that, the object should be manually deleted from the ARIS database via the ARIS user interface.

### Business Process Model Integration from ARIS to Alfabet

The Web Service based ARIS - Alfabet Interoperability Interface reads data about business process models from a selected ARIS database that match the following preconditions:

- The model is a Value Added Chain diagram
- The attribute **Alfabet Root Type** is set to Alfabet - Reference Model.

Each Value Added Chain Diagram matching the preconditions is imported as a separate business process model. All *functions* included in the Value Added Chain diagram are imported to the Alfabet business process model as first level business processes.

All *functions* included in Value Added Chain diagrams that are subordinate to the Value Added Chain diagram are imported to the Alfabet business process model in the hierarchical level corresponding to

the level in the hierarchy of diagrams in ARIS. For example, a function in a Value Added Chain diagram directly subordinate to the root Value Added Chain diagram is imported as second level business process. The number of levels that may be imported to Alfabet is configurable. The default is set to 4 levels.

All Value Added Chain diagrams and *functions* with the attribute **Alfabet - Not to be exported** set to True are excluded from export.

For each exported object, the following attributes are imported to the Alfabet business process model:

- Name
- Description
- GUID
- The ARIS view that opens if a user directly accesses the ARIS interface via the object.



For more information about navigation between the ARIS and Alfabet user interfaces, see the section [Navigation between the ARIS and Alfabet User Interfaces](#).

In Alfabet, the update of business process models based on the data imported from ARIS is performed 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.

The Web Service based ARIS - Alfabet Interoperability Interface creates a solution business process model for each business process model transferred from ARIS to Alfabet if no solution business process model exists that has not yet been checked in. If a solution business process model which is not yet checked in exists, 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 check is performed whether a business process model with an ARIS\_GUID exists that is identical to the GUID of the transferred business process model. If this is not the case, a new business process model with properties transmitted via the interface will be created.
- 2) 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.

- 3) 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.
- 4) For each solution business process within the solution business process model, a check is made to ensure that a business process with the same ALFA\_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.
- 5) 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. 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.

## Navigation between the ARIS and Alfabet User Interfaces

The data transfer between ARIS and Alfabet also activates the direct navigation between the ARIS and Alfabet user interfaces. Multiple links can be defined per object to enable navigation to different views that are relevant for different contexts. For example, a user can navigate from a business process in Alfabet to the fact sheet of the corresponding *function* in ARIS or to a diagram that the *function* is part of.

If a user opens a fact sheet of an *application system type* in ARIS, the attribute section includes the links to the Alfabet views of the application:



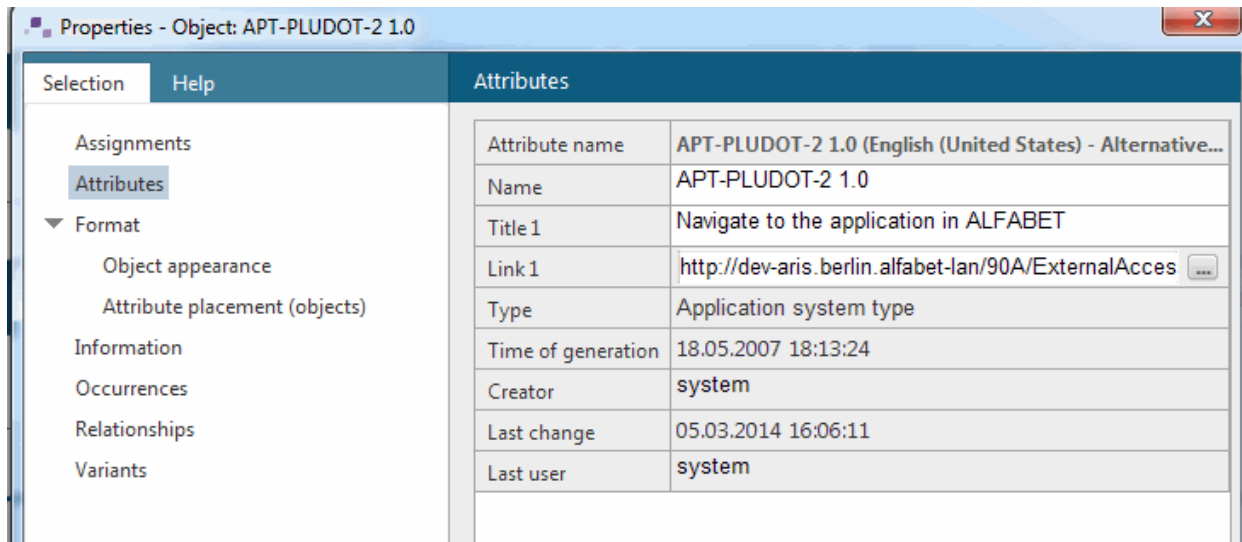


FIGURE 5: Link to the Alfabet user interface from an ARIS application system type fact sheet

Clicking on the link opens the Alfabet user interface in a new tab of the browser, displaying the data about the application.



If the user is not currently logged in to Alfabet, the user may have to enter his/her user name and password in a login window prior to viewing the data. Whether login with user name and password is required depends on the configuration of the access to the Alfabet user interface. For example, if Windows Sign On is performed, no login screen will be displayed and login credentials will be checked automatically.

If a user works with a business process or solution business process in Alfabet, a link to the ARIS user interface is included as a web link to the **Attachments** view of the business process.

## Solution Business Process: Customer services 1

### Attachments

Name	Change Date	Category	Document Type
1 ARIS Architect: Customer services			Web Link
2 ARIS Connect: Customer services			Web Link

FIGURE 6: Link to the ARIS user interface from a business process in Alfabet

Clicking on the link opens a small pane that provides information about the link. The button **Show Details** in the information pane opens the ARIS user interface in a new tab of the browser, displaying the relevant data about the business process.



If the user is not currently logged in to ARIS, the user may have to enter his/her user name and password in a login window prior to viewing the data.

## Interface Architecture

The interface is based on regular data exchange between the components. Data exchange is triggered by a separate software component, the A2A Integrator, that can either be called via a command line on demand or executed in regular intervals via a job scheduling engine. The A2A Integrator calls web services to connect to both the Alfabet Server and the ARIS server.

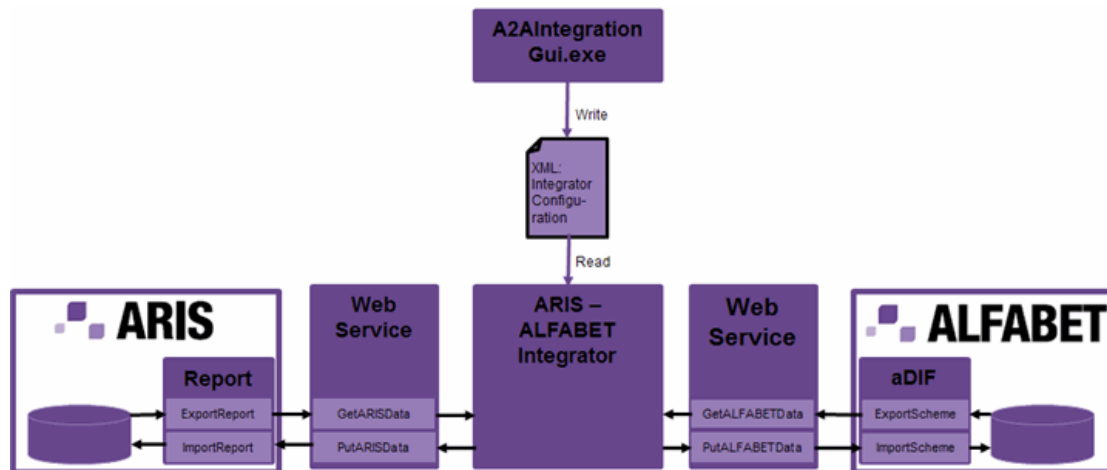


FIGURE 7: Overview of the technical components of the Web Service based ARIS - Alfabet Interoperability Interface

Data processing also involves the following mechanisms to ensure data integrity in the target databases and to provide a means for customer configuration of the interface:

- On Alfabet side:
  - The Alfabet Data Integration Framework (ADIF).
 

ADIF enables data import to or export from the Alfabet database based on ADIF internal mechanisms in combination with native SQL commands. The import or export definition is stored as an ADIF scheme in the Alfabet database. For the Web Service based ARIS - Alfabet Interoperability Interface, the web services trigger the execution of preconfigured, interface-specific ADIF schemes.
- ON ARIS side:
  - JavaScript-based ARIS evaluation reports:
 

The reports enable data import to or export from the ARIS databases. For the Web Service based ARIS - Alfabet Interoperability Interface, the web services trigger the execution of preconfigured, interface-specific reports.

The Web Service based ARIS - Alfabet Interoperability Interface not only transmits data directly between the ARIS and Alfabet databases. Data can be exported to an XML file and imported from an XML file. This implementation is useful for testing and for indirect data integration in a scenario where data export and import shall be executed at different times or in disconnected environments.

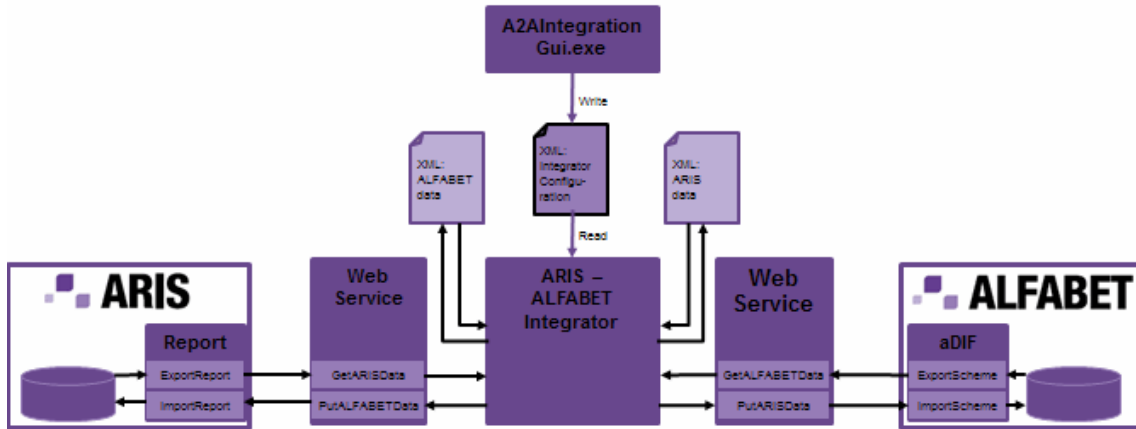


FIGURE 8: Overview of the technical components including all import sources and export targets

## Basic Initial Set-Up of the Web Service based ARIS - Alfabet Interoperability Interface

In order to use the Web Service based ARIS - Alfabet Interoperability Interface the following pre-configuration of the involved software components is required:

- [Installation Requirements](#)
- [Installation](#)
- [Configuration](#)
  - [Configuring the ARIS Data for Import to Alfabet](#)
  - [Configuring the Web Service based ARIS - Alfabet Interoperability Interface](#)

## Installation Requirements

The following components of the ARIS and Alfabet application are required for execution of the functionalities of the ARIS - Alfabet integration interface:

Components required for	ARIS	Interface	Alfabet
Basic set-up, data transfer	ARIS database on database server (third party component) ARIS Connect Server ARIS Design Server	ARIS Web Service on Web Server (third party component) ARIS - Alfabet Integrator Alfabet Web Service on Web Server (third party component)	Alfabet Server Alfabet database on database server (third party component)
Configuration	ARIS Architect (optional, only for changes to the stand-	A2IntegrationGui.exe	Alfabet Expand (optional, only for changes to the

Components required for	ARIS	Interface	Alfabet
	ard)s		standard)
Navigation between the user interfaces	ARIS Architect or ARIS Connect Viewer or ARIS Design Client		Alfabet Web Application Alfabet Web Client

The figure provides an overview of all components involved in the interface functionality.

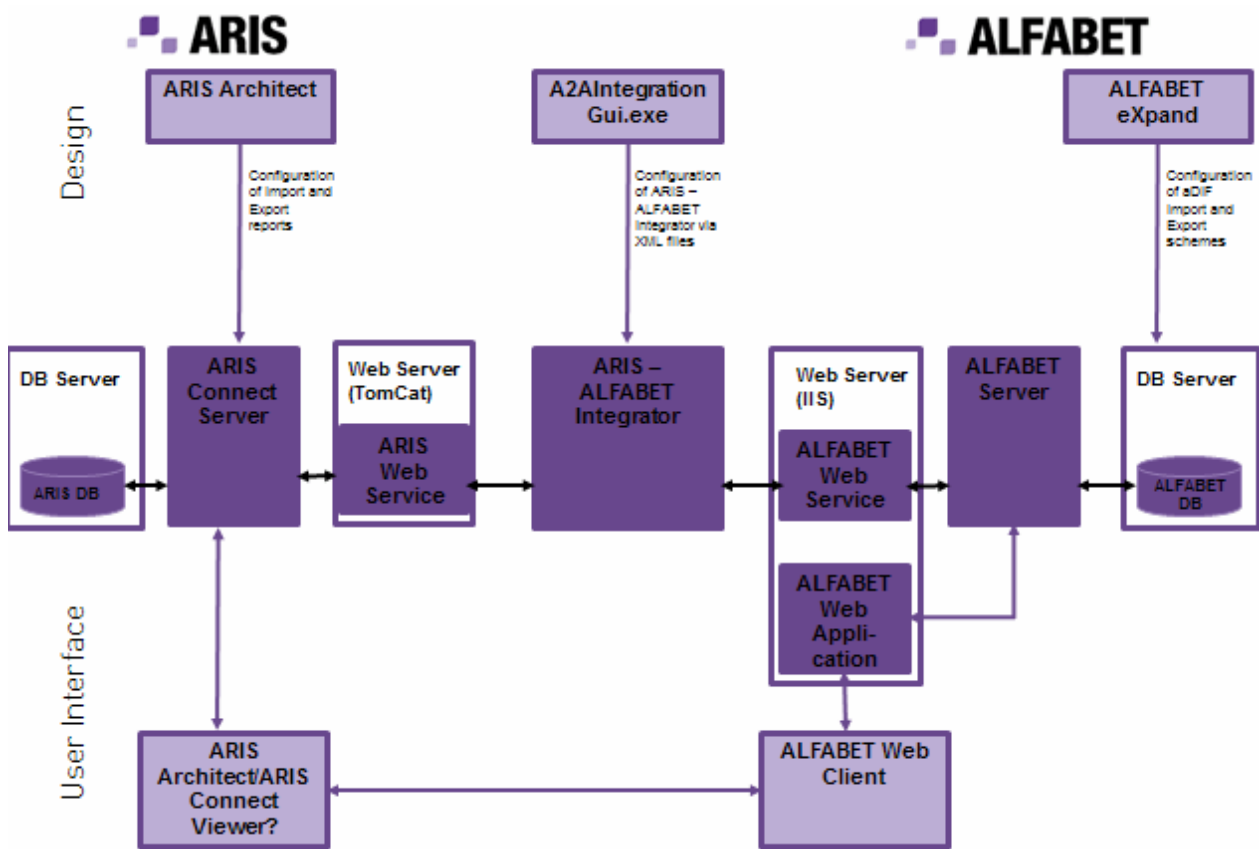


FIGURE 9: Components involved in the Web Service based ARIS - Alfabet Interoperability Interface

The technical requirements for the relevant ARIS and Alfabet components apply. For an overview of the technical requirements of ARIS and Alfabet components, consult the documentation of the respective application. The interface does not have any additional requirements.

### Installation

The following parts of the interface are standard components of ARIS and Alfabet and do not require any further installation:

- The Alfabet ADIF schemes.
- The ARIS reports

- The Alfabet -related attributes in ARIS.

The following overview of the installation procedures required to install the Web Service based ARIS - Alfabet Interoperability Interface presuppose that both ARIS and Alfabet are already installed and is limited to the procedure required to install the interface only:



Installation required on the ARIS side:

- Implement the ARIS web services as a Web Application.

The setup procedure depends on the Web server used. Consult the documentation of your Web server for information about how to set up a Web service. For more information about web service implementation for ARIS, consult the ARIS documentation.

Installation required on the Alfabet side:

- Set up the Alfabet web services on a Web server as a Web Application.

The setup procedure depends on the Web server used. Consult the documentation of your Web server for information about how to set up a Web service. For more information about Alfabet web services, consult the Alfabet documentation.

Installation of the ARIS - Alfabet Integration Interface application:

- For customers with a license for the Web Service based ARIS - Alfabet Interoperability Interface, the required executables are located in the **programs** directory of the Alfabet components. If the interface is licensed for a running version of Alfabet 9.10, the executables are delivered separately and must be copied to the **programs** directory of the Alfabet components.
- [Configure the integration tool.](#) The required configuration is described in the section [Configuration](#).

## Configuration

The following information is available about the configuration required for the Web Service based ARIS - Alfabet Interoperability Interface:

- [Configuring the ARIS Data for Import to Alfabet](#)
- [Configuring the Web Service based ARIS - Alfabet Interoperability Interface](#)

### Configuring the ARIS Data for Import to Alfabet


Data export from the ARIS database requires the following configuration of data before implementing the Web Service based ARIS - Alfabet Interoperability Interface for data transmission:

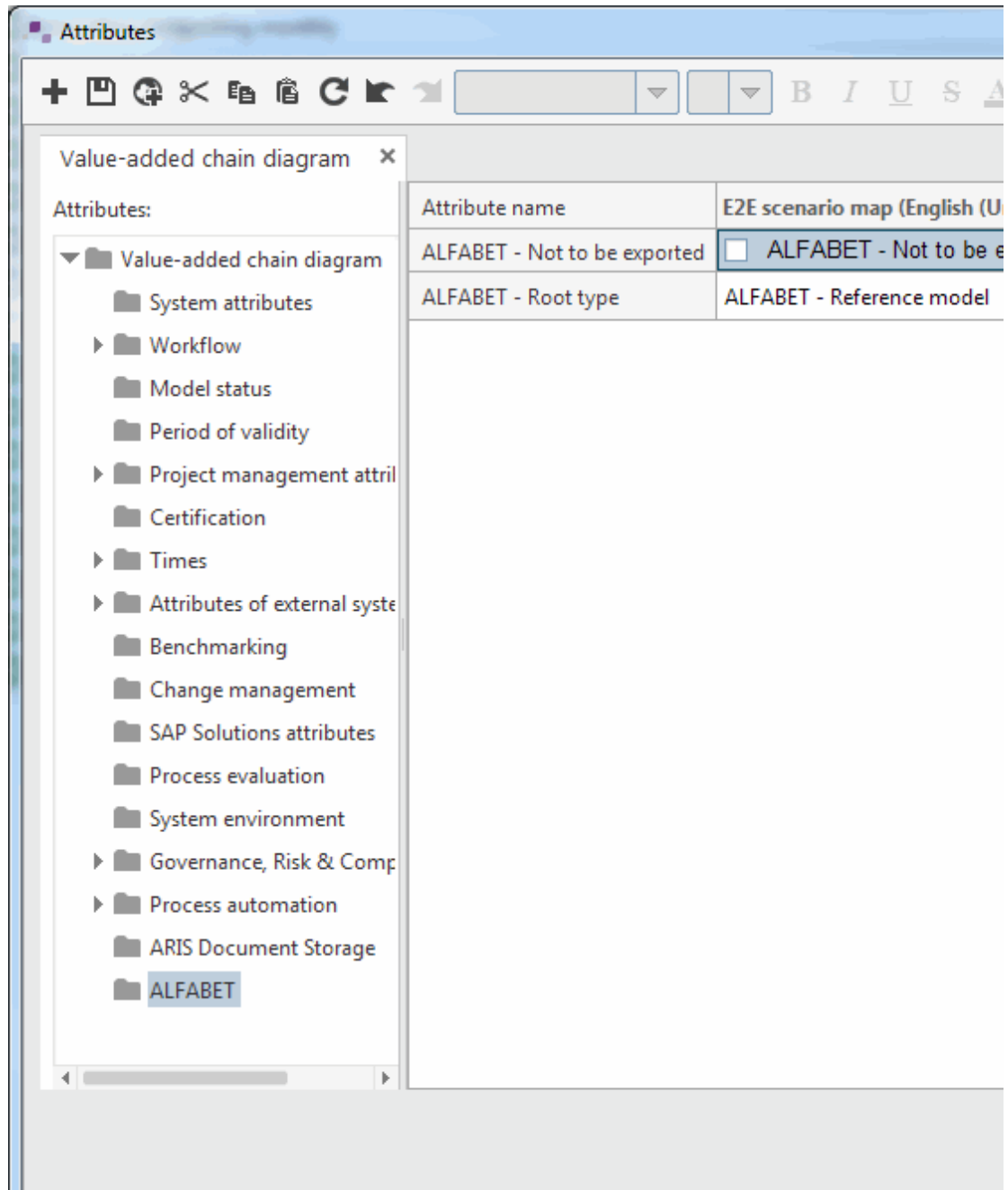


Ensure that you use a method filter that shows the required attributes for the Web Service based ARIS - Alfabet Interoperability Interface. The method filter "Entire Method" enables setting of the required attributes. The attributes are members of the attribute group Alfabet.

- The business process models that are to be re-used in Alfabet must be modelled in Value Added Chain diagrams.

- For each Value Added Chain diagram that is to be imported as a business process model in Alfabet, set the attribute **Alfabet Root Type** to **Alfabet - Reference Model**.

 To set the attributes of a Value Added Chain diagram or any other object in ARIS, right-click the object in the explorer and select, for example, **Attributes**. In the **Attributes** window that opens, click Alfabet in the **Attributes** explorer. The attributes related to the Web Service based ARIS - Alfabet Interoperability Interface are displayed.



- Functions* that are part of the root Value Added Chain diagram and all subordinate Value Added Chain diagrams are exported and imported to Alfabet as business processes. To exclude a *function* or all *functions* of a subordinate Value Added Chain diagram from export, select the check box of the attribute **Alfabet - Not to be exported** in the Attribute window of the function or Value Added Chain diagram.


## Configuring the Web Service based ARIS - Alfabet Interoperability Interface

The connection to the Alfabet and ARIS components must be configured in the configuration file for the Web Service based ARIS - Alfabet Interoperability Interface. Multiple configuration files can be created and stored in parallel.

To configure the Web Service based ARIS - Alfabet Interoperability Interface:

- 1) In the in the **programs** directory of the Alfabet components, double click A2AIntegrationGui.exe. An interface opens.
- 2) In the interface, open the **ARISAccess Configuration** tab and configure the call to the ARIS web services:
  - **ARISWEB Service:** Enter the URL of the ARIS web services for the integration of Alfabet data.
  - **ARISaccess configuration:**
    - **User Name:** Enter the user name for access to the ARIS connect server.
    - **Password:** Enter the password for access to the ARIS server.
  - **ARISdata access configuration:**
    - **ARISDatabase:** Enter the name of the ARIS database that is target for the data import.
    - **ARISTenant:** Enter the name of the tenant for accessing the database.
  - **Export:**
    - **ARISExport Report:** Optionally enter the name of the ARIS report that triggers data export for the ARIS - Alfabet Interoperability Interface. If this field is left blank, the standard report **ARIS2Alfabet- Export processes** is executed.
    - **ZIP export:** Select the check box if the XML export data shall be compressed in a ZIP file.
  - **Import:**
    - **ARISImport Report:** Optionally enter the name of the ARIS report that triggers data import for the ARIS - Alfabet Interoperability Interface. If this field is left blank, the standard report **Alfabet2ARIS- Import application systems** is executed.
    - **ARISImport Expects ZIP:** Select the check box if XML data compressed in a ZIP file shall be imported.
- 3) Open the **Alfabet Access Configuration** tab and configure the call to the Alfabet web services:
  - **Alfabet WEB Service:** Enter the URL of the Alfabet web services for the integration of ARIS data. To test the connection to the web services, click **Check Connection**. A log message regarding the success of the connection is displayed in the field in the bottom of the pane.
  - **Alfabet Application Server Configuration:**
    - **Alfabet Application Server URI (tcp protocol):** The Alfabet Web Service connects to the Alfabet Server via TCP. Enter the path to the Alfabet Server as:
 

```
tcp://<host>:<port>/<alfabetServerAlias>
```

 For example:

```
tcp://127.0.0.1:1880/Alfabet
```
    - **Ensure Security:** Select the check box if the .NET EnsureSecurity shall be used to secure the communication channel to the Alfabet Server.



The settings for host name or IP address, port number, and ensure security must be identical to the specifications in the server alias configuration of the Alfabet Server. For more information about the server alias configuration, see the reference manual Sys-

*tem Administration* in the Alfabet documentation.

- **Alfabet DB access Configuration:**



The Alfabet Web services connect to the Alfabet database via the Alfabet Server. Nevertheless, authentication occurs directly on the database level by providing a user name and password for login to the database on the database server.

- **User Name:** Enter the name of the database user for log in to the Alfabet database on the database server.
- **Password:** Enter the password for log in to the Alfabet database on the database server.

- **Export:**

- **ADIF Export Job:** Enter the name of the ADIF export scheme that shall be executed for data export from Alfabet. To use the standard functionality of the ARIS - Alfabet integration interface, enter ARIS\_Export.
- **Parameters:** For use of the standard Web Service based ARIS - Alfabet Interoperability Interface, enter the URL of the ARIS user interface as:

```
Html5UI=<URLForAccessToalfabetHTML5WebClient>,ObjectView=<ObjectViewToUse>,UserType=<Named|Anonymous>
```

The URL is used to generate the links for access to the Alfabet user interface that are stored in ARIS as attributes of the imported object of the *application system type*. The link is built by connecting the URL defined here with the information about the individual object's view that shall be opened. This is generated during export by the ADIF export scheme. Both a link to the new HTML5 based user interface and the classic user interface can be included in parallel in a comma-separated string.

Additional parameters can be entered into the field if the export is based on a customer-defined ADIF scheme that is configured to perform data export based on parameters given in the command line call for ADIF scheme execution. Define the parameters and their values in a comma-separated string. The format of each parameter definition must be <parameter name>=<value>. The listed parameters can be used in all SQL commands of the ADIF import scheme using the syntax @<parameter name>.



The fields **Assembly**, **Class** and **Method** are not required for the standard export of data via ADIF. They are provided in case an export scenario that cannot be handled via ADIF requires a custom code delivered by Alfabet for the specific export scenario.

- **Import:**


- **ADIF Import Job:** Enter the name of the ADIF import scheme that shall be executed for data import to Alfabet. To use the standard functionality of the Web Service based ARIS - Alfabet Interoperability Interface, enter ARIS\_Import.
- **Parameters:** This field is not required for the standard import functionality of the Web Service based ARIS - Alfabet Interoperability Interface. It is required if the import is based on a customer-defined ADIF scheme that is configured to perform data import based on parameters given in the command line call for ADIF scheme execution. Define the parameters and their values in a comma-separated string. The format of each parameter definition must be <parameter name>=<value>. The listed parameters can be used in all SQL commands of the ADIF import scheme using the syntax @<parameter name>.






The fields **Assembly**, **Class** and **Method** are not required for the standard export of data via ADIF. They are provided in case an export scenario that cannot be handled via




ADIF requires custom code delivered by Alfabet for the specific export scenario.

- 4) In the toolbar, click the **Save**  button. A browser window opens.
- 5) Select the name and location for the configuration file and click **Save** to store the file at the defined location.

 To edit an existing configuration file, click the **Open**  button in the toolbar and select the configuration file in the browser window that opens.

To clear all fields and start a new configuration, click the **New**  button in the toolbar.


Configurations can be tested via the tool A2AIntegrationGui.exe. The tool allows you to configure settings for the export and import and to execute the import or export directly via the interface.




 The settings for source and target of the import or export may be stored in the configuration file, for example if the configuration is saved after testing without removing the settings. Nevertheless they are ignored when running the Web Service based ARIS - Alfabet Interoperability Interface. The action performed by the Web Service based ARIS - Alfabet Interoperability Interface is exclusively defined by the command line parameters.

Export can be directed to an XML file to test the export only. Import can be performed from an XML file containing the import data to test import only.

The following table lists the settings required in the A2AIntegrationGui.exe interface in order to perform different tests for import and/or export and the action required to perform the test.

The window below displays the configuration tabs. Error messages are displayed during test execution.

Test Action to Perform	Required Settings	Starting the Test
Export from ARIS to XML file	<p>In the <b>ARISAccess Configuration</b> tab, go to the <b>Export</b> section and perform the following settings:</p> <ul style="list-style-type: none"> <li>• Select the check box <b>Send Export To File</b>.</li> <li>• Enter the path to and name of the target XML file in the <b>Export File</b> field. You can click the <b>Browse</b>  button to select a location in the browser window. If the specified file does not exist, it is created during the export process.</li> </ul>	Click the <b>Export</b> button in the <b>Export</b> section of the <b>ARISAccess Configuration</b> tab.
Import to Alfabet from XML file	<p>In the <b>Alfabet Access Configuration</b> tab, go to the <b>Import</b> section and perform the following settings:</p> <ul style="list-style-type: none"> <li>• Select the <b>Import From File</b> check box.</li> <li>• Enter the path to and name of the source XML file in the <b>Import File</b> field. You can</li> </ul>	Click the <b>Import</b> button in the <b>Import</b> section of the <b>Alfabet Access Configuration</b> tab.

Test Action to Perform	Required Settings	Starting the Test
	<p>click the <b>Browse</b>  button to select the file in the browser window.</p>	
Export from Alfabet to XML file	<p>In the <b>Alfabet Access Configuration</b> tab, go to the <b>Export</b> section and perform the following settings:</p> <ul style="list-style-type: none"> <li>• Select the <b>Send Export To File</b> check box.</li> <li>• Enter the path to and name of the target XML file in the <b>Export File</b> field. You can click the <b>Browse</b>  button to select a location in the browser window. If the specified file does not exist, it is created during the export process.</li> </ul>	Click the <b>Export</b> button in the <b>Export</b> section of the <b>Alfabet Access Configuration</b> tab.
Import to ARIS from XML file	<p>In the <b>ARISAccess Configuration</b> tab, go to the <b>Import</b> section and perform the following settings:</p> <ul style="list-style-type: none"> <li>• Select the <b>Import from File</b> check box.</li> <li>• Enter the path to and name of the source XML file in the <b>Import File</b> field. You can click the <b>Browse</b>  button to select the file in the browser window.</li> </ul>	Click the <b>Export</b> button in the <b>Export</b> section of the <b>ARISAccess Configuration</b> tab.
Data transfer from ARIS to Alfabet	<p>In the <b>ARISAccess Configuration</b> tab, go to the <b>Export</b> section and perform the following settings:</p> <ul style="list-style-type: none"> <li>• Select the <b>Send Export to Alfabet</b> check box.</li> </ul> <p>In the <b>Alfabet Access Configuration</b> tab, go to the <b>Import</b> section and perform the following settings:</p> <ul style="list-style-type: none"> <li>• Select the <b>Import from ARIS</b> check box.</li> </ul>	Click the <b>Export</b> button in the <b>Export</b> section of the <b>ARISAccess Configuration</b> tab or the <b>Import</b> button in the <b>Import</b> section of the <b>Alfabet Access Configuration</b> tab.
Data transfer from Alfabet to ARIS	<p>In the <b>ARISAccess Configuration</b> tab, go to the <b>Import</b> section and perform the following settings:</p> <ul style="list-style-type: none"> <li>• Select the <b>Import from Alfabet</b> check box.</li> </ul> <p>In the <b>Alfabet Access Configuration</b> tab, go to the <b>Export</b> section and perform the following settings:</p> <ul style="list-style-type: none"> <li>• Select the <b>Send Export to ARIS</b> check box.</li> </ul>	Click the <b>Export</b> button in the <b>Export</b> section of the <b>Alfabet Access Configuration</b> tab or the <b>Import</b> button in the <b>Import</b> section of the <b>ARIS Access Configuration</b> tab.

## Running the Web Service Based ARIS - Alfabet Interoperability Interface

The Web Service based ARIS - Alfabet Interoperability Interface can be run via a command line or as part of a job scheduling engine.

Executable	A2AIntegrationBatch.exe located in the <b>programs</b> directory of the Alfabet components.
Access of applications via	Web Services on both ARIS and Alfabet side.
Preconditions	<ul style="list-style-type: none"> <li>All components of the Web Service based ARIS - Alfabet Interoperability Interface are successfully installed and running and a configuration file is available. For information about installing and configuring the interface, see <a href="#">Basic Initial Set-Up of the Web Service based ARIS - Alfabet Interoperability Interface</a>.</li> <li>The configuration file is located in the installation directory of the interface.</li> <li>Only for import of data from an XML file: The XML file containing the data in the XML structure required for import must be located in the installation directory of the interface.</li> </ul>
Logging	Standard Alfabet logging. For information about standard logging and the command line options, see <a href="#">Command Line Parameters for Logging</a> .
Command line help	Start executable with -h or -help.

The executable must be started with the following parameters:

```
A2AIntegrationBatch.exe -ConfigFile <ConfigurationFile.xml> -ACTION <Action>
```

The table below displays the command line options:

Command Line Option	Mandatory/Optional	Explanation
-ConfigFile <XMLConfigurationFile.xml>	Mandatory	Enter the name of the configuration file that stores the connection data required for the action that you want to define.
-ACTION <Action>	Mandatory	Enter the type of data transfer that you want to perform. Permissible values are: <ul style="list-style-type: none"> <li>ARISFILE for data export from ARIS to an XML file.</li> <li>FILEARIS for data import from an XML file to ARIS.</li> <li>ARISALFABET for data transfer from ARIS to Alfabet</li> </ul>

Command Line Option	Mandatory/ Optional	Explanation
		<ul style="list-style-type: none"> <li>• ALFABETFILE for data export from Alfabet to an XML file.</li> <li>• FILEALFABET for data import from an XML file to Alfabet.</li> <li>• ALFABETARIS for data transfer from Alfabet to ARIS.</li> </ul>



For example to transfer business process model data from ARIS to Alfabet, a configuration file `Transfer_ArisAlfabet.xml` was created defining the required connection to the ARIS and Alfabet application. To start the data transfer from a command line, open a command line window, for example by starting `cmd.exe`, navigate to the directory of the ARIS - Alfabet integration interface and start the executable with:

```
A2AIntegrationBatch.exe -ConfigFile Transfer_ArisAlfabet.xml -ACTION ARIS ALFABET
```

For the standard data transfer of the Web Service based ARIS - Alfabet Interoperability Interface, the following transfer mode is recommended:

- **Transfer of application information from Alfabet to ARIS :**

Application data is subject to constant changes. At the same time, transfer of application data to Alfabet does not require an immediate user interaction via the ARIS user interface. If no changes were performed in Alfabet and data is transferred to ARIS, no changes will be performed in ARIS.

It is recommended to transfer changes to the application landscape to ARIS in regular intervals. Data transfer can be automated for example by scheduling the execution of the batch utility by means of a Windows® batch job.

- **Transfer of business process models from ARIS to Alfabet :**

Business processes are usually more stable than application utilization. At the same time, integration of changes to business process models from ARIS to Alfabet require the manual check in of the resulting solution business process model via the Alfabet user interface.

It is therefore recommended that you transfer changes to the business process model on request (only if changes have actually been made to the business process model in ARIS ).

After data transfer, the following action is required in the interface of the target application:

- Transfer of Application information from Alfabet to ARIS:

Open the group **delete** in the Main Group of your target database and check whether the applications that are in the group are used in any models other than the Application System Type model in the group **delete**. *Application system types* in the group **delete** do not exist any longer in the Alfabet database and shall be deleted in ARIS as well. If an *application system type* in the group **delete** is used in ARIS, substitute it with an *application system type* from the group **active** in the respective models. After correction of modeling, delete the *application system types* in the group **delete**.

- Transfer of business process models from ARIS to Alfabet:

Check the solution business process models for correct assignment of subordinate objects and check in the solution business process models.

## Command Line Parameters for Logging

A standard logging method exists for the Web Service based ARIS - Alfabet Interoperability Interface. Logging information is written to a log file at runtime in the following format:

```
<date and time> <message type> <message text>
```



For example:

```
2009-02-24T12:30:15.4397058Z INFO start data transfer
```

The timestamp is the UTC time (coordinated Universal Time) and may therefore differ from the time in your local time zone. The timestamp is written in ISO 8601 combined date and time format as year-month-dayHour:minutes:secondsZ.

The message type can be one of the following:

- **ERROR:** An error occurred. The message describes the type of error.
- **WARNING:** Problems were encountered during the execution of the utility that are not as severe as an error. The process was executed but the result should be checked. The message describes the problem.
- **INFO:** Information about the normal execution of the utility is provided.

Command line options that can be used when calling the Web Service based ARIS - Alfabet Interoperability Interface utility allow you to specify the number and location of log files as well as the amount of detail of the logging information:

Command Line Option	Default	Description
-logpath <log file path>	Log file is stored in the working directory of the executable.	Specification of a path to a directory for storage of the log file.
-logfile <log file name>	<Executable><timestamp>.log	<p>Specification of the log file name. Allowed file extensions are LOG and TXT.</p> <p><b>NOTE:</b> Setting of -logfile changes the way log information is stored. If -logfile is not defined, a new log file is created each time the utility is used and the log file name is extended with a timestamp specifying the current UTC time. The timestamp is the UTC time written as "_yyymmdd_hhmmss". For example: logfilename_20090328_150827.log.</p> <p>If -logfile is defined, logging information is appended to the already existing log file each time the utility is used. To restrict the file size, you can set the -logclear option to delete old log messages.</p>

Command Line Option	Default	Description
-nologappend		<p>This option is only evaluated if -logfile is defined.</p> <p>If -nologappend is set, the log file defined with -logfile and -logpath is overwritten with a new file containing the current log information each time the utility is used.</p> <p>If -nologappend is not set, logging information is appended to the already existing log file each time the utility is used.</p> <p><b>NOTE:</b> To restrict the file size, you can set the -logclear option to delete old log messages.</p>
-logverbose		<p>If -logverbose is set, additional information about the running process is logged. The content and amount of additional information messages depend on the utility used. If the setting of -logverbose does not change the log output, there are no additional information messages available for the utility.</p> <p><b>NOTE:</b> Verbose logging is in most cases not required and can lead to a decrease in performance.</p>
-logclear <number of days>	Infinite	<p>This option can only be used if -logfile is defined and -nologappend is not set.</p> <p>During logging, the log file is scanned for log messages with a timestamp older than the number of days specified with -logclear and these messages are deleted.</p> <p><b>NOTE:</b> The scanning process can lead to drawbacks in performance.</p>

## Advanced Set-Up: Customizing Data Transfer from Alfabet to ARIS

The standard data exchanged triggered by the Web Service based ARIS - Alfabet Integration Interface can be adapted by the user to meet company specific requests. For example:

- For both applications and business processes the range of attribute/property values that is transmitted from ARIS to Alfabet and vice versa can be changed to include customer defined or standard data stored about the application or business process.
- Other object types can be added to the transfer of data between the applications.



For example organizations are both part of ARIS project planning and Alfabet enterprise architecture management. The data for organizations may therefore be included into the

data exchange between ARIS and Alfabet.

- The target of the links between the applications can be changed to access the object in the other user interface via a different than the standard view.



For example a customized object profile has been designed for applications in Alfabet with the name APP\_ImageView\_Customized. This object profile includes links to customer configured reports allow to edit and manage some customer defined properties of the application that are not part of the standard Alfabet meta-model. When a user navigates from an *application system type* in ARIS to the corresponding application in Alfabet, the object profile APP\_ImageView\_Customized shall be displayed instead of the standard object profile APP\_ImageView.

The data exchange between ARIS and Alfabet is triggered via an XML that is written by one of the applications providing the data and processed by the application to that the data is imported. To change the content of data transmission, the content of the XML must be adapted to include the new content. This requires changes in both the interface writing the data and the interface processing the data:

- On ARIS side the reports that generate or process the XML data must be changed. The standard reports for data exchange are
  - **Alfabet 2 ARIS - Import application systems**
  - **ARIS 2 Alfabet - Export processes**

The reports are Java based and can be customized in a user interface in the ARIS Architect. To change the scope of data import from or export to ARIS, the standard reports can be altered or completely new reports can be defined. If a new report is defined, the Web Service based ARIS - Alfabet Interoperability Interface must be configured to trigger the customer configured report.



- For information about how to configure ARIS reports, see the ARIS documentation.
- For information about how to configure the ARIS - Alfabet Integration Interface to use any other than the standard report, see [Configuration](#).

- On Alfabet side the ADIF Import and Export schemes that generate and process the XML data must be changed. The standard ADIF schemes for data exchange are:
  - **ARIS\_Import**
  - **ARIS\_Export**

The ADIF schemes are XML based and can be customized in a user interface in the tool Alfabet Expand . To change the scope of data import from or export to Alfabet, the standard ADIF schemes can be altered or completely new ADIF schemes can be defined. If a new ADIF scheme is defined, the Web Service based ARIS - Alfabet Interoperability Interface must be configured to trigger the customer configured report.



- For information about how to configure ADIF schemes, see the Alfabet documentation (Reference manual *Alfabet Data Integration Framework* ).
- For information about how to configure the ARIS - Alfabet Integration Interface to use any other than the standard report, see [Configuration](#).

This chapter provides you with the special information about the XML structure of the XML for data exchange between Alfabet and ARIS required to perform the correct configuration of both Alfabet ADIF schemes and ARIS reports.

- [The XML for Data Transfer From Alfabet to ARIS](#)

- [Links from ARIS to Alfabet](#)
  - [Required Syntax for the URL Linking to the Alfabet User Interface](#)
  - [Defining Links for Display in Multiple Languages](#)
- [The XML for Data Transfer From ARIS to Alfabet](#)
  - [Links from Alfabet to ARIS](#)
    - [Required Syntax for the URL Linking to the ARIS User Interface](#)
    - [Defining Links for Display in Multiple Languages](#)

## The XML for Data Transfer From Alfabet to ARIS

The XML has the following structure:

XML level	XML Elements	XML Attributes	Allowed child elements	Stored information
Root	Objects	None	Object, Relation	
First Level	Object	alfa_type alfa_guid aris_type	Attribute, Object	Information about an object of a defined ARIS object type/ Alfabet object class.
Second Level	Attribute	Name type language aris_type	<text>	Stores the value of a defined ARIS object type attribute / Alfabet object class property for the object.
Second Level	Object	alfa_type aris_type	Attribute	Stores a link for navigation between the user interfaces of ARIS and Alfabet.
Third Level	Attribute	Name type language aris_type	<text>	Stores data relevant for the link between the user interfaces of ARIS and Alfabet.

Each element **Object** represents an object that is exported/imported via the interface. The element **Object** must have the following attributes to store the information about which object is exported and to map the object with existing objects in the target database:



XML Attribute	Value	Mandatory/Optional
alfa_type	The identifier for the object class in Alfabet. It is recommended to use the name of the object class in Alfabet to identify the object class. The ADIF export scheme must be configured to write the identifier for the object class into the attribute alfa_type for each object of the class.	Mandatory
alfa_guid	The GUID of the object in the Alfabet database. In the ARIS database, the alfa_guid values are stored by the attribute type <i>Alfabet - GUID</i> .	Mandatory
aris_type	The identifier for the object type in ARIS. It is recommended to use the API name of the object type to identify the object type. The mapping of the object with a defined identifier to the correct object type is performed via the ARIS import report.	Mandatory

Attributes of the object class properties in Alfabet are mapped to attribute types in ARIS. Mapping is done by adding one element **Attribute** as child element to the **Object** element for each ARIS attribute / Alfabet property mapping that shall be performed. The **Attribute** element has the following attributes:

Attribute	Value	Mandatory/Optional
Name	The identifier for the object class property in Alfabet. It is recommended to use the name of the object class property in Alfabet to identify the object class property. The ADIF export scheme must be configured to write the identifier for the object class property into the XML attribute Name for each XML element Attribute exporting data for the object class property.	Mandatory
aris_type	The identifier for the attribute type in ARIS. It is recommended to use the API name of the attribute type to identify the attribute. The mapping of the XML attribute with a defined identifier to the correct attribute type is performed via the ARIS import report.	Mandatory
type	The data type of the value. Permissible data types are: Boolean, Date, DateTime, Integer, Real, String, Text, Time, URL.	Mandatory
language	The language culture name of the language the value is defined for. The language culture name must be written in lower case letters with a hyphen between language and country information (for example en-us, de-de)	Mandatory

The information about the value of the attribute is stored as text within the XML element **Attribute**.

### Links from ARIS to Alfabet

Multiple links to the Alfabet user interface can be defined per object. For each link, an XML element **Object** must be added as child element to the XML element **Object** defining an imported object. The XML element **Object** must be defined with the following attribute settings that shall not be altered:

Attribute	Value	Mandatory/Optional
alfa_type	A2ALink	Mandatory
aris_type	AT_LINK_INFO	Mandatory

The XML element **Object** must have two child elements **Attribute** to define a single link. The XML elements **Attribute** define the URL to the Alfabet user interface and the title of the link displayed on the ARIS user interface. URL and title are defined as text within the **Attribute** elements while the setting of the attributes of the **Attribute** element specify whether this element stores a title or a URL.

The following table lists the required settings for the attributes of the XML elements **Attribute** defining a link:

Attribute	Value for Title Definition	Value for Link Definition	Remark
Name	Name	Link	
type	string	url	
language	locale id of culture	locale id (LCID as string) of culture	The ARIS user interface is available for multiple language cultures. The locale ID (LCID) defines for which language the link definition in the current element is used.
aris_type	AT_EXT_<number>	AT_EXT_<number>	Multiple links to the Alfabet user interface can be defined for the same object. Each link defined for an object must have a different number. Please note that the same number must be used for all XML elements <b>Attribute</b> within an XML element <b>Object</b> . This includes different language versions of a link.
<text> (content of element)	The title string to be displayed	URL definition	The required syntax for the URL is described in the section <a href="#">Required Syntax for the URL Linking to the Alfabet User Interface</a> .

## Required Syntax for the URL Linking to the Alfabet User Interface

The link for access to the Alfabet user interface must have the following syntax:

**Alfabet\_URL** /ExternalAccess.aspx?AccessType=ExternalAccess&UserType=Name&Object  
= **REFSTROfObject** &View=ObjectView: **ObjectViewName**

with the following variables depending on the Alfabet installation and the object to be accessed:

- **Alfabet\_URL**: The URL for access to the Alfabet user interface.
- **REFSTROfObject**: The value of the REFSTR property of the object in the Alfabet database. Within an Alfabet database, the REFSTR is a unique property that allows the object to be identified unambiguously.
- **ObjectViewName**: The name of the object profile that shall open when the user clicks the link. The object profile must be defined for the object class of the object for that the URL is defined.



An object profile in Alfabet corresponds to the fact sheet of an object in ARIS. Object profiles can be created and/or customized. For more information about object profiles in the Alfabet meta-model and the customization of the object profiles, see the reference manual *Configuring Alfabet with Alfabet Expand* of the Alfabet documentation.

## Defining Links for Display in Multiple Languages

When the Alfabet user interface is displayed in multiple different languages, the two **Attribute** elements defining the link must be repeated for each language. The settings are the same except for the XML attribute **language** that must define the locale id the definition is valid for and the text content of the title definition that must include the translation into the defined language.

## The XML for Data Transfer From ARIS to Alfabet

The XML has the following structure:

XML level	XML Elements	XML Attributes	Allowed child elements	Stored information
Root	Objects	None	Object, Relation	
First Level	Object	alfa_guid aris_type aris_guid	Attribute, Object	Information about an object of a defined ARIS object type/ Alfabet object class.
First Level	Relation	from_guid to_guid alfa_name	<text>	Information about a relationship with another object.

XML level	XML Elements	XML Attributes	Allowed child elements	Stored information
		aris_type		
Second Level	Attribute	Name type language aris_type	<text>	Stores the value of a defined ARIS object type attribute / Alfabet object class property for the object.
Second Level	Object	alfa_type aris_type	Attribute	Stores a link for navigation between the user interfaces of ARIS and Alfabet.
Third Level	Attribute	Name type language aris_type	<text>	Stores data relevant for the link between the user interfaces of ARIS and Alfabet.

Each XML element **Object** represents an object that is exported/imported via the interface. The XML element **Object** must have the following XML attributes to store the information about which object is exported and to map the object with existing objects in the target database:

Attribute	Value	Mandatory/Optional
alfa_type	The identifier for the object class in Alfabet. It is recommended to use the name of the object class in Alfabet to identify the object class. The ADIF import scheme must be configured to write the identifier into the XML attribute alfa_type for each object.	Mandatory
aris_guid	The GUID of the object in the ARIS database. The Alfabet attribute ARIS_Guid stores the information about the aris_guid.	Mandatory
aris_type	The identifier for the object type in ARIS. It is recommended to use the API name of the object type to identify the object type. The mapping of the object with a defined identifier to the correct object type is performed via the ARIS import report.	Mandatory

Attributes of the object type in ARIS are mapped to object class properties in Alfabet and vice versa. Mapping is done by adding one XML element **Attribute** as child element to the XML element **Object** for each ARIS attribute / Alfabet property mapping that shall be performed. The XML element **Attribute** has the following XML attributes:

Attribute	Value	Mandatory/Optional
Name	The identifier for the object class property in Alfabet. It is recommended to use the name of the object class property in Alfabet to identify the object class property. The mapping of the attribute with a defined identifier to the correct object class property is performed via the ADIF import scheme.	Mandatory
aris_type	The identifier for the attribute type in ARIS. It is recommended to use the API name of the attribute type to identify the attribute. The ARIS export report must be configured to add the correct identifier to the XML for defined attribute types.	Mandatory
type	The data type of the value. Permissible data types are: Boolean, Date, DateTime, Integer, Real, String, Text, Time, URL.	Mandatory
language	The language culture name of the language the value is defined for. The language culture name must be written in lower case letters with a hyphen between language and country information (for example en-us, de-de)	Mandatory

The information about the value of the attribute is stored as text within the XML element **Attribute**.

### Links from Alfabet to ARIS

Multiple links to the Alfabet user interface can be defined per object. For each link, an XML element **Object** must be added as child element to the XML element **Object** defining an imported object. The XML element **Object** must be defined with the following XML attribute settings that shall not be altered:

Attribute	Value	Mandatory/Optional
alfa_type	AT_LINK_INFO	Mandatory
aris_type	LINK_ARCHITECT or LINK_CONNECT	Mandatory

The XML element **Object** must have two child elements **Attribute** to define a single link. The XML elements **Attribute** define the URL to the Alfabet user interface and the title of the link displayed on the ARIS user interface. URL and title are defined as text within the XML elements **Attribute** while the setting of the XML attributes of the XML element **Attribute** specify whether this XML element stores a title or a URL.

The following table lists the required settings for the XML attributes of the XML element **Attribute** defining a link:

Attribute	Value for Title Definition	Value for Link Definition	Remark
Name	Name	Link	
type	string	url	
language	locale id of culture	locale id (LCID as string) of culture	The ARIS user interface is available for multiple language cultures. The locale ID (LCID) defines for which language the link definition in the current element is used.
aris_type	AT_EXT_<number>	AT_EXT_<number>	Multiple links to the ARIS user interface can be defined for the same object. Each link defined for an object must have a different number. Please note that the same number must be used for all XML elements <b>Attribute</b> within an XML element <b>Object</b> . This includes different language versions of a link.
<text> (content of element)	The title string to be displayed	URL definition	The required syntax for the URL is described in the section <a href="#">Required Syntax for the URL Linking to the Alfabet User Interface</a> .

### Required Syntax for the URL Linking to the ARIS User Interface

The link for access to the ARIS user interface must have the following syntax:

**ARIS\_URL** /#default/item/ **objectidentifier**

with the following variables depending on the Alfabet installation and the object to be accessed:

- **ARIS\_URL**: The URL for access to the ARIS user interface.
- **objectidentifier**: link target definition as defined in ARIS export report.

### Defining Links for Display in Multiple Languages

When the ARIS user interface is displayed in multiple different languages, the two XML elements **Attribute** defining the link must be repeated for each language. The settings are the same except for the XML attribute **language** that must define the locale id the definition is valid for and the text content of the title definition that must include the translation into the defined language.

## Index

A2AIntegrationBatch.exe			
command line parameters	59		
logging	61		
A2AIntegrationGui.exe	55		
aDIF import scheme			
ARISIntegration_ImportFromARIS	35		
Alfabet database			
temporary database tables	36		
Alfabet user interface			
configuring	29		
Alfabet Web Application			
configuring RESTful services	24		
AMM file	22		
ARIS Connect Server			
configuring authorization	23		
configuring RESTful services	22		
ARIS Diagrams page view	14		
ARIS function			
creating Alfabet business process	14		
ARIS model diagram			
integrating changes to Alfabet	16		
integrating in Alfabet business process model	14		
ARIS-ALFABET-Integration.xml	23		
ArisApiConfig			
connection to ARIS	24		
data mapping	26		
navigation	28		
ArisGenericDataConnectivity	32		
ARISIntegration_ImportFromARIS			
generated temporary tables	36		
attribute group			
ALFABET in ARIS	23		
authorization			
from Alfabet to ARIS	24		
from ARIS to Alfabet	22		
RESTful services key for Alfabet	22		
business process model			
ARIS Diagrams page view	14		
building from ARIS data	15		
configuring integration	26		
integrating current changes from ARIS	16		
integrating from ARIS	14		
standard mapping of ARIS data	14		
Business Process Model ARIS Source Report	16		
command line parameters			
for data transfer	59		
components			
for interface	51		
configuration			
ARIS database	53		
batch tool	55		
connections to servers	55		
customization			
data import from ARIS	31		
data exchange			
from Alfabet to ARIS	13, 45		
from ARIS to Alfabet	13, 46		
data mapping			
from ARIS to Alfabet	26		
data transfer			
customizing	62		
running batch job	59		
database table			
temporary for ARIS assignments	41		
temporary for ARIS attributes	42		
temporary for ARIS connections	40		
temporary for ARIS models	36		
temporary for ARIS objects	38		
documentation	9		
Enterprise Architecture Extension	22		
functionality			
RESTful service based	13		
implementation			
RESTful service based ARIS - Alfabet Interoperability	21		
installation	52		

interface		RESTful service	
components	51	customization	31
pre-installation requirements	51	RESTful services	
running batch job	59	Alfabet license key	22
starting data transmission	59	configuring connection from Alfabet to ARIS	24, 26
technical overview	50	configuring connection from ARIS to Alfabet	22
key		configuring navigation from Alfabet to ARIS	28
for Alfabet RESTful services	22	solution business process planning	16
for Alfabet user	22	temporary database table	36
licenses	22	ARIS assignments	41
link		ARIS attributes	42
configuring	28	ARIS connections	40
to the ARIS user interface	19	ARIS models	36
logging		ARIS objects	38
for data transfer	61	terminology	11
navigating		token	
from Alfabet to ARIS	19	generating for Alfabet user	22
navigation		web.config	22
configuring	28	XML structure	
navigation between user interfaces	48	from Alfabet to ARIS	64
preconditions		from ARIS to ALFABET	67
RESTful service based ARIS - Alfabet			
Interoperability	22		
pre-installation requirements	51		