



Informatica® PowerExchange for Microsoft
Azure Data Lake Storage Gen1
10.2.2

User Guide

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, PowerExchange, and Big Data Management are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Table of Contents

Preface	5
Informatica Resources.	5
Informatica Network.	5
Informatica Knowledge Base.	5
Informatica Documentation.	5
Informatica Product Availability Matrices.	6
Informatica Velocity.	6
Informatica Marketplace.	6
Informatica Global Customer Support.	6
 Chapter 1: Introduction to PowerExchange for Microsoft Azure Data Lake Store.....	 7
PowerExchange for Microsoft Azure Data Lake Store Overview.	7
Introduction to Microsoft Azure Data Lake Store.	8
Administration of Microsoft Azure Data Lake Store Connector.	8
 Chapter 2: PowerExchange for Microsoft Azure Data Lake Store Configuration.....	 9
PowerExchange for Microsoft Azure Data Lake Store Configuration Overview.	9
Prerequisites.	9
 Chapter 3: Microsoft Azure Data Lake Store Connections.....	 11
Microsoft Azure Data Lake Store Connections Overview.	11
Microsoft Azure Data Lake Store Connection Properties.	11
Creating a Microsoft Azure Data Lake Store Connection.	12
 Chapter 4: PowerExchange for Microsoft Azure Data Lake Store Data Objects.....	 14
Microsoft Azure Data Lake Store Data Object Overview.	14
Microsoft Azure Data Lake Store Data Object Properties.	15
Microsoft Azure Data Lake Store Data Object Read Operation.	15
Directory Source in Microsoft Azure Data Lake Store Sources.	15
Microsoft Azure Data Lake Store Object Read Operation Properties.	16
Source Properties of the Data Object Read Operation.	16
General Properties - Source.	16
Ports Properties - Source.	16
Advanced Properties - Source.	17
Column Projection Properties.	17
Microsoft Azure Data Lake Store Data Object Write Operation Properties.	18
Input Properties of the Data Object Write Operation.	18
Ports Properties - Input Write.	19

Run-time Properties.	19
Advanced Properties - Target.	20
Column Projection Properties.	20
Importing a Microsoft Azure Data Lake Store Data Object.	21
Creating a Microsoft Azure Data Lake Store Object Read or Write Operation.	22
Creating a Microsoft Azure Data Lake Store Target.	22
Rules and Guidelines for Microsoft Azure Data Lake Store Target Data Object.	23
Chapter 5: Microsoft Azure Data Lake Store Mappings.....	24
Microsoft Azure Data Lake Store Mapping Overview.	24
Mapping Validation and Run-time Environments.	25
Microsoft Azure Data Lake Store Mapping Example.	25
Microsoft Azure Data Lake Store Dynamic Mapping Overview.	26
Refresh Schema.	27
Mapping Flow.	27
Microsoft Azure Data Lake Store Dynamic Mapping Example.	27
Appendix A: Microsoft Azure Data Lake Store Datatype Reference.....	29
Datatype Reference Overview.	29
Microsoft Azure Data Lake Store and Transformation Data Types.	30
Avro File Data Types and Transformation Data Types.	30
Intelligent Structure Model Data Types and Transformation Types.	31
JSON File Data Types and Transformation Data Types.	32
Parquet File Data Types and Transformation Data Types.	33
Index.....	34

Preface

The *Informatica PowerExchange® for Microsoft Azure Data Lake Store User Guide* provides information about reading data from and writing data to Microsoft Azure Data Lake Store. It is written for administrators and developers who create mappings to read from or write data to a Microsoft Azure Data Lake Store resource.

This guide assumes that you have knowledge of Microsoft Azure Data Lake Store and Informatica.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit <https://docs.informatica.com>.

Informatica maintains documentation for many products on the Informatica Knowledge Base in addition to the Documentation Portal. If you cannot find documentation for your product or product version on the Documentation Portal, search the Knowledge Base at <https://search.informatica.com>.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at <http://velocity.informatica.com>. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <https://network.informatica.com> and select the eSupport option.

CHAPTER 1

Introduction to PowerExchange for Microsoft Azure Data Lake Store

This chapter includes the following topics:

- [PowerExchange for Microsoft Azure Data Lake Store Overview, 7](#)
- [Introduction to Microsoft Azure Data Lake Store, 8](#)
- [Administration of Microsoft Azure Data Lake Store Connector, 8](#)

PowerExchange for Microsoft Azure Data Lake Store Overview

All the occurrences of *PowerExchange for Microsoft Azure Data Lake Store* in this document denote *PowerExchange for Microsoft Azure Data Lake Storage Gen1*.

You can use PowerExchange for Microsoft Azure Data Lake Store to connect to Microsoft Azure Data Lake Store from Informatica.

Use PowerExchange for Microsoft Azure Data Lake Store to read data from and write data to Microsoft Azure Data Lake Store. You can collate and organize the details from multiple input sources and use PowerExchange for Microsoft Azure Data Lake Store to write data to Microsoft Azure Data Lake Store. You can use Microsoft Azure Data Lake Store objects as sources and targets in mappings and dynamic mappings. When you use Microsoft Azure Data Lake Store objects in mappings, you must configure properties specific to Microsoft Azure Data Lake Store. You can validate and run mappings in native or Hadoop environments.

Use PowerExchange for Microsoft Azure Data Lake Store to read flat files and the industry-standard file formats, such as Avro, Parquet, and JSON files in the Hadoop environments. You can read and write hierarchical data present in the Avro, Parquet, and JSON files. You can read data from and write data to only uncompressed Avro, Parquet, and JSON files. In addition to the industry-standard file formats, you can also read data from intelligent structure sources on the Spark engine. You can read and write Avro and Parquet files that contain only primitive data types in the native environment.

Introduction to Microsoft Azure Data Lake Store

You can use Microsoft Azure Data Lake Store to store data irrespective of size, structure, and format. Use Microsoft Azure Data Lake Store to process large volumes of data to achieve faster business outcomes.

The Microsoft Azure Data Lake Store is a combination of following primary services:

Microsoft Azure Data Lake Store: Microsoft Azure Data Lake is an Hadoop Distributed File System (HDFS) compatible data repository that stores structured, semi structured, and unstructured data of your organization in native format. Data scientists and data analysts can use data in the Data Lake to find out specific patterns before you move the analyzed data to a data warehouse.

Azure HDInsight: Azure HDInsight is an Apache Hadoop distributed system that is configured with cloud. HDInsight supports the Hadoop components such as Apache HBase, Apache Spark, and Apache Storm and other technologies under the Hadoop system.

Data Lake Analytics: Data Lake Analytics manages the distributed infrastructure. You can dynamically allocate and de-allocate resources with Microsoft Azure Data Lake Store Analytics so that you can pay for services that you use.

Administration of Microsoft Azure Data Lake Store Connector

Before you use PowerExchange for Microsoft Azure Data Lake Store, perform the following tasks:

- Create a Microsoft Azure Data Lake Store.
- Authorize users to access the Microsoft Azure Data Lake Store and the directory in the Data Lake Store.
- Create an Azure AD web application for service-to-service authentication with Azure Data Lake Store. For information on service-to-service authentication of your web application, see, <https://kb.informatica.com/howto/6/Pages/20/512374.aspx>.

Note: Ensure that you have superuser privileges to access the folders or files created in the application using PowerExchange for Microsoft Azure Data Lake Store.

CHAPTER 2

PowerExchange for Microsoft Azure Data Lake Store Configuration

This chapter includes the following topics:

- [PowerExchange for Microsoft Azure Data Lake Store Configuration Overview, 9](#)
- [Prerequisites, 9](#)

PowerExchange for Microsoft Azure Data Lake Store Configuration Overview

PowerExchange for Microsoft Azure Data Lake Store installs with the Informatica services and clients.

To configure PowerExchange for Microsoft Azure Data Lake Store, complete the prerequisites.

Prerequisites

Before you use PowerExchange for Microsoft Azure Data Lake Store, you must complete the following prerequisites:

- Install and configure the Informatica services.
- Install and configure the Developer tool. You can install the Developer tool when you install Informatica clients.
- Create a Data Integration Service and a Model Repository Service in the Informatica domain.
- Verify that a cluster configuration is created in the domain.

Configure Databricks Connection Advanced Properties

Verify that a Databricks connection is created in the domain. If you want to read NULL values from or write NULL values to an Azure source, configure the following advanced properties in the Databricks connection:

- `infaspark.flatfile.reader.nullValue=True`

- `infaspark.flatfile.writer.nullValue=True`

Configure Azure Data Lake Store Access in Azure Databricks Cluster

You need to set some Hadoop credential configuration options under Spark Config in your Databricks cluster configuration to access the Azure Data Lake Store. Add "spark.hadoop" as a prefix to the Hadoop configuration keys as shown in the following text:

```
spark.hadoop.dfs.adls.oauth2.access.token.provider.type ClientCredential
spark.hadoop.dfs.adls.oauth2.client.id <your-service-client-id>
spark.hadoop.dfs.adls.oauth2.credential <your-service-credentials>
spark.hadoop.dfs.adls.oauth2.refresh.url "https://login.microsoftonline.com/<your-
directory-id>/oauth2/token"
```

CHAPTER 3

Microsoft Azure Data Lake Store Connections

This chapter includes the following topics:

- [Microsoft Azure Data Lake Store Connections Overview, 11](#)
- [Microsoft Azure Data Lake Store Connection Properties, 11](#)
- [Creating a Microsoft Azure Data Lake Store Connection, 12](#)

Microsoft Azure Data Lake Store Connections Overview

Microsoft Azure Data Lake Store connection enables you to read data from or write data to Microsoft Azure Data Lake Store.

You can use Microsoft Azure Data Lake Store connections to create data objects and run mappings. The Developer tool uses the connection when you create a data object. The Data Integration Service uses the connection when you run mappings.

You can create a Microsoft Azure Data Lake Store connection from the Developer tool or the Administrator tool. Create and manage connections in the Preferences dialog box or the Connection Explorer view.

Microsoft Azure Data Lake Store Connection Properties

Use a Microsoft Azure Data Lake Store connection to access a Microsoft Azure Data Lake Store.

Note: The order of the connection properties might vary depending on the tool where you view them.

You can create and manage a Microsoft Azure SQL Data Warehouse connection in the Administrator tool or the Developer tool. The following table describes the Microsoft Azure Data Lake Store connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! \$ % ^ & * () - + = { [] \ ; ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select Microsoft Azure Data Lake Store.

The following table describes the properties for metadata access:

Property	Description
ADLS Account Name	The name of the Microsoft Azure Data Lake Store.
ClientID	The ID of your application to complete the OAuth Authentication in the Active Directory.
Client Secret	The client secret key to complete the OAuth Authentication in the Active Directory.
Directory	The Microsoft Azure Data Lake Store directory that you use to read data or write data. The default is root directory.
AuthEndpoint	The OAuth 2.0 token endpoint from where access code is generated based on based on the Client ID and Client secret is completed.

For more information about creating a client ID, client secret, and auth end point, contact the Azure administrator or see Microsoft Azure Data Lake Store documentation.

Creating a Microsoft Azure Data Lake Store Connection

Create a Microsoft Azure Data Lake Store connection before you create a Microsoft Azure Data Lake Store data object.

1. In the Developer tool, click **Window > Preferences**.
2. Select **Informatica > Connections**.
3. Expand the domain in the **Available Connections**.
4. Select the connection type **Enterprise Application > ADLS**, and click **Add**.

5. Enter a connection name and an optional description.
6. Select Microsoft Azure Data Lake Store as the connection type.
7. Click **Next**.
8. Configure the connection properties.
9. Click **Test Connection** to verify the connection to Microsoft Azure Data Lake Store.
10. Click **Finish**.

CHAPTER 4

PowerExchange for Microsoft Azure Data Lake Store Data Objects

This chapter includes the following topics:

- [Microsoft Azure Data Lake Store Data Object Overview, 14](#)
- [Microsoft Azure Data Lake Store Data Object Properties, 15](#)
- [Microsoft Azure Data Lake Store Data Object Read Operation, 15](#)
- [Microsoft Azure Data Lake Store Data Object Write Operation Properties, 18](#)
- [Importing a Microsoft Azure Data Lake Store Data Object, 21](#)
- [Creating a Microsoft Azure Data Lake Store Object Read or Write Operation, 22](#)
- [Creating a Microsoft Azure Data Lake Store Target, 22](#)

Microsoft Azure Data Lake Store Data Object Overview

A Microsoft Azure Data Lake Store data object is a physical data object that represents Microsoft Azure Data Lake Store table as a source or target. A Microsoft Azure Data Lake Store data object is the representation of data that is based on a Microsoft Azure Data Lake Store object. You can configure the data object read and write operation properties that determine how data can be read from Microsoft Azure Data Lake Store or loaded to Microsoft Azure Data Lake Store.

To read data from the Microsoft Azure Data Lake Store, create a data object read operation based on the Microsoft Azure Data Lake Store data object. Configure the read operation properties to determine how the Data Integration Service must read data from the Microsoft Azure Data Lake Store table. Add the read operation as a Read transformation in a mapping.

To write data to the Microsoft Azure Data Lake Store, create a data object write operation based on the Microsoft Azure Data Lake Store data object. Configure the write operation properties to determine how the Data Integration Service must write data to the Microsoft Azure Data Lake Store. Add the write operation as a Write transformation in a mapping.

Microsoft Azure Data Lake Store Data Object Properties

The Microsoft Azure Data Lake Store Overview view displays general information about the Microsoft Azure Data Lake Store data object and the object properties that apply to the Microsoft Azure Data Lake Store table you import.

General Properties

You can configure the following properties for a Microsoft Azure Data Lake Store data object:

- Name. Name of the Microsoft Azure Data Lake Store data object.
- Description. Description of the Microsoft Azure Data Lake Store data object.
- Connection. Name of the Microsoft Azure Data Lake Store connection.

Microsoft Azure Data Lake Store Data Object Read Operation

The Data Integration Service reads data from a Microsoft Azure Data Lake Store table based on the data object read operation properties that you specify.

When you create a data object read operation, the Developer tool creates a Source transformation and an Output transformation.

The Source transformation represents the data that the Data Integration Service reads from the Microsoft Azure Data Lake Store table.

The Output transformation represents the data that the Data Integration Service passes into the mapping pipeline.

Directory Source in Microsoft Azure Data Lake Store Sources

You can select the type of source from which you want to read data.

You can select the following type of sources from the **Source Type** option under the advanced properties for a Microsoft Azure Data Lake Store data object read operation:

- File
- Directory

Note: To read compressed `.csv` files, the content type of the Microsoft Azure Data Lake Store source must be `application/x-gzip` for both File and Directory source types.

Use the following rules and guidelines to select **Directory** as the source type:

- All the source files in the directory must contain the same metadata.
- The directory read is not applicable to the binary file type.
- All the files must have data in the same format. For example, delimiters, header fields, and escape characters must be same.
- All the files under a specified directory are parsed. The files under subdirectories are not parsed.

- The connector does not perform any validation if there are multiple file formats in the directory you select and might result into errors.

Microsoft Azure Data Lake Store Object Read Operation Properties

The Data Integration Service reads data from a Microsoft Azure Data Lake Store object based on the data object read operation. The Developer tool displays the data object read operation properties of the Microsoft Azure Data Lake Store data object in the Data Object Operation view.

You can view or configure the data object read operation from the source and output properties.

Source properties

Represents data that the Data Integration Service reads from the Microsoft Azure Data Lake Store object. Select the source properties to view data, such as the name and description of the Microsoft Azure Data Lake Store object, the ports, and advanced properties.

Output properties

Represents data that the Data Integration Service passes into the mapping pipeline. Select the output properties to edit the port properties of the data object read operation. You can also use the Tracing Level advanced property to set the amount of detail that the Data Integration Service writes in the log.

Source Properties of the Data Object Read Operation

When you create a data object, the source properties populate based on the Microsoft Azure Data Lake Store object that you add. The source properties of the data object read operation include general, column, and advanced properties that apply to the Microsoft Azure Data Lake Store object.

You can view the source properties of the data object read operation from the **General**, **Column**, and **Advanced** tabs.

General Properties - Source

The following table describes the source general properties of the data object read operation:

Property	Description
Name	Name of the Microsoft Azure Data Lake Store source object.
Description	Description of the data object read operation.

Ports Properties - Source

The column properties display the data types, precision, and scale of the source property in the data object read operation.

The following table describes the source column properties of the data object read operation:

Property	Description
Name	Name of the column.
Type	Native data type of the column.

Property	Description
Precision	Maximum number of significant digits for Numeric data types, or maximum number of characters for String data types. For numeric data types, precision includes scale.
Scale	Maximum number of digits after the decimal point for numeric values.
Description	Description of the column.

Advanced Properties - Source

You can use the advanced properties to specify data object read operation properties to read data from a Microsoft Azure Data Lake Store server.

The following table describes the Advanced source column properties of the data object read operation:

Property	Description
Directory Path Override	Overrides the default directory path.
File Name Override	Overrides the file name.
Source Type	Select the type of source from which you want to read data. You can select the following source types: <ul style="list-style-type: none"> - File - Directory Default is File. The directory read is not applicable to the binary file type. You can read compressed <code>.csv</code> files in the native environment.

Column Projection Properties

The Developer tool displays the column projection properties for intelligent structure model, Avro, JSON, and Parquet complex file sources in the Properties view of the **Read** operation.

The following table describes the column projection properties that you configure for the complex file sources:

Property	Description
Enable Column Projection	Displays the column details of the complex files sources.
Use Schema	Enables the schema selection for the complex file. You can select a different schema. Note: If you disable the column projection, the schema associated with the complex file is removed. If you want to associate schema again with the complex file, enable the column projection and click Add Schema .
Schema Format	Displays the schema format that you selected while creating the complex file data object. You can change the schema format and provide respective schema.

Property	Description
Use Intelligent Structure Model	Select this option to associate an intelligent structure model with the complex file. Warning: Do not associate an intelligent structure model with a Write data object operation. If you use a Write operation that is associated with an intelligent structure model in a mapping, the mapping will not be valid.
Model	Displays the intelligent structure model associated with the complex file. You can select a different model. Note: If you disable the column projection, the intelligent structure model associated with the data object is removed. If you want to associate an intelligent structure model again with the data object, enable the column projection and click Select Model .
Column Mapping	Displays the mapping between input and output ports. Note: If you disable the column projection, the mapping between input and output ports is removed. If you want to map the input and output ports, enable the column projection and click Select Schema to associate a schema to the complex file.
Project Column as Complex Data Type	Displays columns with hierarchical data as a complex data type, such as, array, map, or struct. Select this property when you want to process hierarchical data on the Spark engine. Note: If you disable the column projection, the data type of the column is displayed as binary type.

Microsoft Azure Data Lake Store Data Object Write Operation Properties

The Data Integration Service writes data to a Microsoft Azure Data Lake Store object based on the data object write operation. The Developer tool displays the data object write operation properties for the Microsoft Azure Data Lake Store data object in the Data Object Operation section.

You can view the data object write operation from the Input and Target properties.

Input properties

Represent data that the Data Integration Service reads from a Microsoft Azure Data Lake Store directory server. Select the input properties to edit the port properties and specify the advanced properties of the data object write operation.

Target properties

Represent data that the Data Integration Service writes to Microsoft Azure Data Lake Store. Select the target properties to view data, such as the name and description of the Microsoft Azure Data Lake Store object.

Input Properties of the Data Object Write Operation

Input properties represent data that the Data Integration Service writes to a Microsoft Azure Data Lake Store directory server. Select the input properties to edit the port properties of the data object write operation. You

can also specify advanced data object write operation properties to write data to Microsoft Azure Data Lake Store objects.

The input properties of the data object write operation include general properties that apply to the data object write operation. Input properties also include port, source, and advanced properties that apply to the data object write operation.

You can view and change the input properties of the data object write operation from the **General**, **Ports**, **Targets**, **run-time**, and **Advanced** tabs.

Ports Properties - Input Write

The input ports properties list the data types, precision, and scale of the data object write operation.

The following table describes the input ports properties that you must configure in the data object write operation:

Property	Description
Name	Name of the port.
Type	Data type of the port.
Precision	Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale.
Scale	Maximum number of digits after the decimal point for numeric values.
Description	Description of the port.

Run-time Properties

The run-time properties display the name of the connection used for write transformation.

The following table describes the run-time properties that you configure for a Microsoft Azure Data Lake Store write operation:

Property	Description
Connection	Name of the Microsoft Azure Data Lake Store connection.

Advanced Properties - Target

You can use the advanced properties to specify data object write operation properties to write data to a Microsoft Azure Data Lake Store server.

The following table describes the advanced properties that you configure for a Microsoft Azure Data Lake Store write operation:

Property	Description
Tracing Level	By default, the tracing level for every transformation is Normal. Change the tracing level to a Verbose setting when you need to troubleshoot a transformation that is not behaving as expected. Set the tracing level to Terse when you want the minimum amount of detail to appear in the log.
Maintain row order	Not applicable
Directory Path Override	Overrides the default directory path.
File Name Override	Overrides the file name.
Compress Newly Created File	Enables compression. Applicable to the .csv files in the native environment.
If file exists	If the target exists, overwrites or appends the data to the existing file. You can also configure the mapping to fail if the target already exists. Applicable to .csv files in the native environment.

Column Projection Properties

The Developer tool displays the column projection properties for Avro, JSON, and Parquet complex file targets in the Properties view of the **Write** operation.

The following table describes the advanced properties that you configure for Avro, JSON, and Parquet complex file targets:

Property	Description
Enable Column Projection	Displays the column details of the complex files sources.
Use Schema	Enables the schema selection for the complex file. You can select a different schema. Note: If you disable the column projection, the schema associated with the complex file is removed. If you want to associate schema again with the complex file, enable the column projection and click Add Schema .
Schema Format	Displays the schema format that you selected while creating the complex file data object. You can change the schema format and provide respective schema.
Use Intelligent Structure Model	Not applicable for the data object write operation.
Model	Not applicable for the data object write operation.

Property	Description
Column Mapping	Displays the mapping between input and output ports. Note: If you disable the column projection, the mapping between input and output ports is removed. If you want to map the input and output ports, enable the column projection and click Select Schema to associate a schema to the complex file.
Project Column as Complex Data Type	Displays columns with hierarchical data as a complex data type, such as, array, map, or struct. Select this property when you want to process hierarchical data on the Spark engine. Note: If you disable the column projection, the data type of the column is displayed as binary type.

Importing a Microsoft Azure Data Lake Store Data Object

Import a Microsoft Azure Data Lake Store data object to add to a mapping.

1. Select a project or folder in the **Object Explorer** view.
2. Click **File > New > Data Object**.
3. Select **ADLS Data Object** and click **Next**.

The **ADLS Data Object** dialog box appears.

4. Enter a name for the data object.
5. In the **Resource Format** list, select any of the following formats:
 - Intelligent Structure Model: to read any format that an intelligent structure parses.
 - Binary: to read and write any resource format.
 - Flat: to read and write delimited resources.
 - Avro: to read and write Avro resources.
 - Json: to read and write JSON resources.
 - Parquet: to read and write Parquet resources.

Note: Intelligent structure model and JSON formats are supported only on the Spark engine. For a data object with an intelligent structure model, create a read operation. You cannot use a write transformation for a data object with an intelligent structure model in a mapping. Avro and Parquet are supported in the native and Hadoop environments.

6. Click **Browse** next to the **Location** option and select the target project or folder.
7. Click **Browse** next to the **Connection** option and select the Microsoft Azure Data Lake Store connection from which you want to import the Microsoft Azure Data Lake Store resource metadata.
8. To add a resource, click **Add** next to the **Selected Resources** option.
The **Add Resource** dialog box appears.
9. From the Package Explorer, select a naming context from which you want to import the schema.
10. You can Perform one of the following tasks to import an Microsoft Azure Data Lake Store table, and then click **OK**:

- Navigate to the Microsoft Azure Data Lake Store table that you want to import.
 - Search for the Microsoft Azure Data Lake Store table, enter the name of the Microsoft Azure Data Lake Store table entity that you want to add and click **OK**.
11. Click **Finish**.
The data object appears under Data Objects in the project or folder in the **Object Explorer** view.

Creating a Microsoft Azure Data Lake Store Object Read or Write Operation

You can add a Microsoft Azure Data Lake Store data object read or write operation to a mapping or mapplet as a source. You can create the data object read or write operation for one or more Microsoft Azure Data Lake Store data objects.

Before you create a Microsoft Azure Data Lake Store data object read or write operation, you must create at least one Microsoft Azure Data Lake Store data object.

1. Select the data object in the Object Explorer view.
2. Right-click and select **New > Data Object Operation**.
The **Data Object Operation** dialog box appears.
3. Enter a name for the data object read or write operation.
4. Select **Read** or **Write** as the type of data object operation.
5. Click **Add**.
The **Select Resources** dialog box appears.
6. Select the Microsoft Azure Data Lake Store object for which you want to create the data object read or write operation and click **OK**.
7. Click **Finish**.

The Developer tool creates the data object read or write operation for the selected data object.

Creating a Microsoft Azure Data Lake Store Target

You can create a Microsoft Azure Data Lake Store target using the **Create Target** option.

1. Select a project or folder in the **Object Explorer** view.
2. Select a source or a transformation in the mapping.
3. Right-click the Source transformation and select **Create Target**.
The **Create Target** dialog box appears.
4. Select **Others** and then select **adlsv2** data object from the list in the **Data Object Type** section.
5. Click **OK**.
The **New adlsv2 Data Object** dialog box appears.
6. Enter a name for the data object.

7. In the **Resource Format** list, select any of the following formats to create the target type:

- Avro
- Flat
- JSON
- Parquet

8. Click **Finish**.

The new target appears under the **Physical Data Objects** category in the project or folder in the **Object Explorer** view.

Rules and Guidelines for Microsoft Azure Data Lake Store Target Data Object

Use the following rules and guidelines when you create a new Microsoft Azure Data Lake Store target:

- You must specify a connection for the newly created Microsoft Azure Data Lake Store target in the **Connection** field to run a mapping.
- When you write an Avro, JSON, or Parquet file using the **Create Target** option, you cannot provide a Null data type.
- When you select a flat resource format that contains different data types and select the **Create Target** option to create a Microsoft Azure Data Lake Store target, the Data Integration Service creates string ports for all the data types with a precision of 256 characters.
- When you select a flat resource format to create a Microsoft Azure Data Lake Store target, the Data Integration Service maps all the data types in the source file to the String data type in the target file. You must manually map the data types in the source and target files.
- For a newly created Microsoft Azure Data Lake Store target, the Data Integration Service considers the value of the folder path that you specify in the **Directory** connection property and file name from the **Native Name** property in the Microsoft Azure Data Lake Store data object details. Provide a directory path and file name in the Microsoft Azure Data Lake Store data object read and write advanced properties to overwrite the values.
- When you use a flat resource format to create a target, the Data Integration Service considers the following values for the formatting options:

Formatting Options	Values
Delimiters	Comma (,)
Text Qualifier	No quotes
Import Column Names From First Line	Generates header
Row Delimiter	Backslash with a character n (\n)
Escape Character	Empty

If you want to configure the formatting options, you must manually edit the projected columns.

CHAPTER 5

Microsoft Azure Data Lake Store Mappings

This chapter includes the following topics:

- [Microsoft Azure Data Lake Store Mapping Overview, 24](#)
- [Mapping Validation and Run-time Environments, 25](#)
- [Microsoft Azure Data Lake Store Mapping Example, 25](#)
- [Microsoft Azure Data Lake Store Dynamic Mapping Overview, 26](#)
- [Microsoft Azure Data Lake Store Dynamic Mapping Example, 27](#)

Microsoft Azure Data Lake Store Mapping Overview

After you create the Microsoft Azure Data Lake Store data object with a Microsoft Azure Data Lake Store connection, you can develop a mapping. You can define the following types of objects in the mapping:

- A Read transformation of the Microsoft Azure Data Lake Store data object to read data from Microsoft Azure Data Lake Store.
- A Write transformation of the Microsoft Azure Data Lake Store data object to write data to Microsoft Azure Data Lake Store.

Validate and run the mapping. You can deploy the mapping and run it or add the mapping to a Mapping task in a workflow. You cannot define a Lookup transformation of the Microsoft Azure Data Lake Store data object.

Note: If you use multiple connections in a mapping, verify that all connections point to the same Microsoft Azure Data Lake Store. If you run a mapping on the HDInsight cluster that uses Microsoft Azure Data Lake Store file system, the Microsoft Azure Data Lake Store connector must point to the same Microsoft Azure Data Lake Store.

When you read data from Microsoft Azure Data Lake Store and terminate the mapping, the Integration Service continues to download the entire data set and then, deletes the data set. This might take a long time based on the size of the data set.

Mapping Validation and Run-time Environments

You can validate and run mappings in the native environment or in a non-native environment, such as Hadoop or Databricks.

When you validate a mapping, you can validate it against one or all of the engines. The Developer tool returns validation messages for each engine.

When you run a mapping, you can choose to run the mapping in the native environment or in a non-native environment, such as Hadoop or Databricks. Configure the run-time environment in the Developer tool to optimize mapping performance and process data that is greater than 10 terabytes. When you run mappings in the native environment, the Data Integration Service processes and runs the mapping. When you run mappings in a non-native environment, the Data Integration Service pushes the processing to a compute cluster, such as Hadoop or Databricks.

You can run standalone mappings, mappings that are a part of a workflow in a non-native environment. When you select the Hadoop environment, the Data Integration Service pushes the mapping logic to the Spark engine.

When you select the Databricks environment, the Integration Service pushes the mapping logic to the Databricks Spark engine, the Apache Spark engine packaged for Databricks.

Microsoft Azure Data Lake Store Mapping Example

You work as a data analyst for a large financial enterprise. The enterprise performs risk management, fraud detection, and other analysis with Microsoft Azure Data Lake Analytics. You need to write the data to Microsoft Azure Data Lake Store to perform the analytics. Create a mapping to read data from a flat file source and write data to Microsoft Azure Data Lake Store. After the data is available in the Microsoft Azure Data Lake Store, you can perform the data analytics.

You can use the following objects in a Microsoft Azure Data Lake Store mapping:

Flat file input

The input file is a flat file that contains the customer names and other details about customers.

Create a flat file data object. Configure the flat file connection and specify the flat file that contains the customer data as a resource for the data object. Drag the data object into a mapping as a read data object.

Transformations

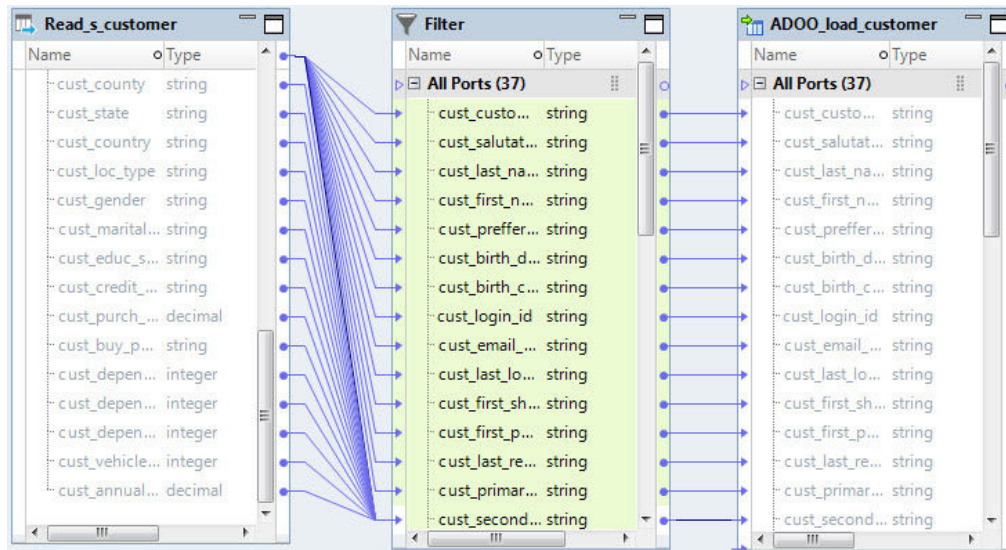
Add Filter transformation to get customer data in a particular region.

The Filter transformation filters the source data based on the value you specify for the region ID column. The Data Integration Service returns the rows that meet the filter condition.

Microsoft Azure Data Lake Store output

Create a Microsoft Azure Data Lake Store data object write operation. Configure the Microsoft Azure Data Lake Store connection and specify the Microsoft Azure Data Lake Store object as a target for the data object. Drag the data object into a mapping as a target data object.

The following image shows the Microsoft Azure Data Lake Store mapping example:



When you run the mapping, the customer records are read from the flat file and written to the Microsoft Azure Data Lake Store table.

Microsoft Azure Data Lake Store Dynamic Mapping Overview

You can use Microsoft Azure Data Lake Store data objects as dynamic sources and targets in a mapping.

Use the Microsoft Azure Data Lake Store dynamic mapping to accommodate changes to source, target, and transformation logics at run time. You can use a Microsoft Azure Data Lake Store dynamic mapping to manage frequent schema or metadata changes or to reuse the mapping logic for data sources with different schemas. Configure rules, parameters, and general transformation properties to create the dynamic mapping.

If the data source for a source or target changes, you can configure a mapping to dynamically get metadata changes at runtime. If a source changes, you can configure the Read transformation to accommodate changes. If a target changes, you can configure the Write transformation to accommodate target changes.

You do not need to manually synchronize the data object and update each transformation before you run the mapping again. The Data Integration Service dynamically determines transformation ports, transformation logic in the ports, and the port links within the mapping.

There are two options available to enable a mapping to run dynamically. You can select one of the following options to enable the dynamic mapping:

- In the **Data Object** tab of the data object read or write operation, select the **At runtime, get data object columns from data source** option when you create a mapping.
When you enable the dynamic mapping using this option, you can refresh the source and target schemas at the runtime.
- In the **Ports** tab of the data object write operation, select the value of the **Columns defined by** property as **Mapping Flow** when you configure the data object write operation properties.

Note: Dynamic mapping is applicable when you run the mapping in the native environment or on the Spark engine.

For information about dynamic mappings, see the *Informatica Developer Mapping Guide*.

Refresh Schema

You can refresh the source or target schema at the runtime when you enable a mapping to run dynamically. You can refresh the imported metadata before you run the dynamic mapping.

You can enable a mapping to run dynamically using the **At runtime, get data object columns from data source** option in the **Data Object** tab of the Read and Write transformations when you create a mapping.

When you add or override the metadata dynamically, you can include all the existing source and target objects in a single mapping and run the mapping. You do not have to change the source schema to update the data objects and mappings manually to incorporate all the new changes in the mapping.

You can use the mapping template rules to tune the behavior of the execution of such pipeline mapping.

When the Source or Target transformation contains updated ports such as changes in the port names, data types, precision, or scale, the Data Integration Service fetches the updated ports and runs the mapping dynamically. You must ensure that at least one of the column name in the source or target file is the same as before refreshing the schema to run the dynamic mapping successfully.

Even though the original order of the source or target ports in the table changes, the Data Integration Service displays the original order of the ports in the table when you refresh the schemas at runtime.

If there are more columns in the source file as compared to the target file, the Data Integration Service does not map the extra column to the target file and loads null data for all the unmapped columns in the target file.

If the Source transformation contains updated columns that do not match the Target transformation, the Data Integration Service does not link the new ports by default when you refresh the source or target schema. You must create a run-time link between the transformations to link ports at run time based on a parameter or link policy in the **Run-time Linking** tab and update the target schema manually. For information about run-time linking, see the *Informatica Developer Mapping Guide*.

Note: When you refresh a schema of a flat file, the Data Integration Service writes all data types as String data types.

Mapping Flow

You can add all the Source transformation or transformation ports to the target dynamically when enable a mapping to run dynamically using the **Mapping Flow** option. You can then use the dynamic ports in the Write transformation.

When you select the **Mapping Flow** option, the Data Integration Service allows the Target transformation to override ports of the Write transformation with all the updated incoming ports from the pipeline mapping and loads the target table with the ports at runtime.

The Data Integration Service creates the target tables dynamically based on the metadata of the incoming ports from the pipeline mapping.

To enable a dynamic mapping using the **Mapping Flow** option, select the value of the **Columns defined by** property as **Mapping Flow** in the **Ports** tab in the Write transformation.

Microsoft Azure Data Lake Store Dynamic Mapping Example

Your organization has a large amount of data that keeps changing. Your organization needs to incorporate all the updated data in a short span of time. Create a dynamic mapping, where you can refresh the source

schema dynamically to fetch the updated data. Add all the dynamic ports to the target to override the metadata of the existing ports.

1. Import the Microsoft Azure Data Lake Store read and write data objects.
2. Select a project or folder in the **Object Explorer** view.
3. Click **File > New > Mapping**.
The **Mapping** dialog box appears.
4. Enter the name of the mapping in the **Name** field.
5. Click **Finish**.
6. Drag the data object into a mapping.
The **Adlsv2 Data Object Access** dialog box appears.
7. Select the **Read** option and click **OK**.
8. In the **Data Object** tab, select the **At runtime, get data object columns from data source** check box.
9. Drag the data object into a mapping.
The **Adlsv2 Data Object Access** dialog box appears.
10. Select the **Write** option and click **OK**.
11. In the **Ports** tab, select the value of the **Columns defined by** as **Mapping Flow**.
12. Select all the source incoming ports and add the ports to the target.
13. Save and run the mapping.

APPENDIX A

Microsoft Azure Data Lake Store Datatype Reference

This appendix includes the following topics:

- [Datatype Reference Overview, 29](#)
- [Microsoft Azure Data Lake Store and Transformation Data Types, 30](#)
- [Avro File Data Types and Transformation Data Types, 30](#)
- [Intelligent Structure Model Data Types and Transformation Types, 31](#)
- [JSON File Data Types and Transformation Data Types, 32](#)
- [Parquet File Data Types and Transformation Data Types, 33](#)

Datatype Reference Overview

Informatica Developer uses the following data types in Microsoft Azure Data Lake Store mappings:

- Microsoft Azure Data Lake Store native data types. Microsoft Azure Data Lake Store data types appear in the physical data object column properties.
- Transformation data types. Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Data Integration Service uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

When PowerExchange for Microsoft Azure Data Lake Store reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When PowerExchange for Microsoft Azure Data Lake Store writes to a target, it converts the transformation data types to the comparable native data types.

Microsoft Azure Data Lake Store and Transformation Data Types

The following table lists the Microsoft Azure Data Lake Store data types that the Data Integration Service supports and the corresponding transformation data types:

Microsoft Azure Data Lake Store Native Data Type	Transformation Data Type	Range
String	string	1 to 104,857,600 characters
Number	Decimal	Precision 1 to 28 digits, scale 0 to 28
Binary	Binary	1 to 104,857,600 bytes The binary data type is applicable when you run a mapping in the native environment.

Avro File Data Types and Transformation Data Types

Avro file data types map to transformation data types that the Data Integration Service uses to move data across platforms.

The following table lists the Avro file data types that the Data Integration Service supports and the corresponding transformation data types:

Avro File Data Type	Transformation Data Type	Range and Description
Array	Array	Unlimited number of characters
Boolean	Integer	TRUE (1) or FALSE (0)
Bytes	Binary	Precision 4000
Double	Double	Precision 15
Float	Double	Precision 15
Int	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Long	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
Map	Map	Unlimited number of characters

Avro File Data Type	Transformation Data Type	Range and Description
Null	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Record	Struct	Unlimited number of characters
String	String	1 to 104,857,600 characters

Note: The Null data type is supported only in the native environment.

Avro Union Data Type

A union indicates that a field might have more than one data type. For example, a union might indicate that a field can be a string or a null. A union is represented as a JSON array containing the data types.

The Developer tool only interprets a union of ["primitive_type|complex_type", "null"] or ["null", "primitive_type|complex_type"]. The Avro data type converts to the corresponding transformation data type. The Developer tool ignores the null.

Unsupported Avro Data Types

The Developer tool does not support the following Avro data types:

- enum
- null

Intelligent Structure Model Data Types and Transformation Types

Intelligent structure model data types map to transformation data types that the Data Integration Service uses to move data across platforms.

The following table lists the intelligent structure model data types that the Data Integration Service supports and the corresponding transformation data types:

Intelligent Structure Model Complex File Data Type	Transformation Data Type	Range and Description
Array	Array	Unlimited number of characters.
Bigint	Bigint	Precision of 19 digits, scale of 0.
Datetime	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
Double	Double	Precision of 15 digits.

Intelligent Structure Model Complex File Data Type	Transformation Data Type	Range and Description
Int	Integer	-2,147,483,648 to 2,147,483,647.
Number	Decimal	Precision 1 to 28, scale 0 to 28
String	String	1 to 104,857,600 characters
Struct	Struct	Unlimited number of characters.

JSON File Data Types and Transformation Data Types

The following table lists the JSON data types that the Data Integration Service supports and the corresponding transformation data types:

JSON File Data Type	Transformation Data Type	Range and Description
Array	Array	Unlimited number of characters.
Bigint	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
Double	Double	Precision 15
Int	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Object	Struct	Unlimited number of characters.
String	String	1 to 104,857,600 characters

Unsupported JSON Data Types

The Developer tool does not support the following JSON data types:

- date/timestamp
- enum
- union

Parquet File Data Types and Transformation Data Types

Parquet file data types map to transformation data types that the Data Integration Service uses to move data across platforms.

The following table lists the Parquet file data types that the Data Integration Service supports and the corresponding transformation data types:

Parquet File Data Type	Transformation Data Type	Range and Description
Binary (UTF-8)	String	1 to 104,857,600 characters
Boolean	Integer	TRUE (1) or FALSE (0)
Byte_Array	Binary	Arbitrarily long byte array
Double	Double	Precision 15
Float	Double	Precision 15
Int32	Integer	-2,147,483,648 to +2,147,483,647 Precision of 10, scale of 0
Int64	Bigint	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807 Precision of 19, scale of 0
Map	Map	Unlimited number of characters.
Struct	Struct	Unlimited number of characters.
Union	Corresponding primitive data type in a union of ["primitive_type", "null"] or ["null", "primitive_type"].	Dependent on primitive data type.

The Parquet schema that you specify to read or write a Parquet file must be in smaller case. Parquet does not support case-sensitive schema.

Parquet Union Data Type

A union indicates that a field might have more than one data type. For example, a union might indicate that a field can be a string or a null. A union is represented as a JSON array containing the data types. The Developer tool only interprets a union of ["primitive_type", "null"] or ["null", "primitive_type"]. The Parquet data type converts to the corresponding transformation data type. The Developer tool ignores the null.

Unsupported Parquet Data Types

The Developer tool does not support the following Parquet data types:

- int96 (TIMESTAMP_MILLIS)
- date
- timestamp

INDEX

A

advanced properties
source [17](#)
target [20](#)

C

create
Microsoft Azure Data Lake Store connection [12](#)
create target
Microsoft Azure Data Lake Store [22](#)
creating
Microsoft Azure Data Lake Store data object read operation [22](#)

D

data object read operation
creating [22](#)
Data types
Avro file [30](#)
JSON file [32](#)
transformation [30](#)
directory source
Microsoft Azure Data Lake Store sources [15](#)

G

General properties [16](#)

I

importing
Microsoft Azure Data Lake Store data object [21](#)
input properties [19](#)
intelligent structure model complex file data types
transformation data types [31](#)

M

mapping example
Microsoft Azure Data Lake Store [25](#)
mapping flow
dynamic mapping [27](#)
Microsoft Azure Data Lake
introduction [8](#)
Microsoft Azure Data Lake Store
create [12](#)
dynamic mapping [26](#)
importing a data object [21](#)

Microsoft Azure Data Lake Store connection
overview [11](#)
properties [11](#)
Microsoft Azure Data Lake Store Connector
administration [8](#)
Microsoft Azure Data Lake Store data object
overview [14](#)
Microsoft Azure Data Lake Store data object read operation
creating [22](#)
Microsoft Azure Data Lake Store data types
comparing with transformation data types [30](#)
overview [29](#)
Microsoft Azure Data Lake Store dynamic mapping
example [28](#)
Microsoft Azure Data Lake Store mappings
overview [24](#)
Microsoft Azure Data Lake Store run-time environment
description [25](#)
Microsoft Azure Data Lake Store validation environment
description [25](#)
Microsoft Azure Data Lake Store write operation
properties [18](#)

O

Output properties [16](#)
overview
Microsoft Azure Data Lake Store connection [11](#)

P

Parquet file data types
transformation data types [33](#)
Ports properties [16](#)
PowerExchange for Microsoft Azure Data Lake Store configuration
overview [9](#)
prerequisites [9](#)

R

read operation properties
column projection properties [17](#)
refresh schema
dynamic mapping [27](#)
Rules and Guidelines
Microsoft Azure Data Lake Store target [23](#)

S

source properties [16](#)

W

write operation properties
column projection properties [20](#)