Informatica

Informatica® Vibe Data Stream for Machine Data

2.3.0

User Guide

Informatica Vibe Data Stream for Machine Data User Guide
2.3.0
December 2015

Publication Date: 2018-07-02

Table of Contents

Preface

The *Vibe Data Stream for Machine Data User Guide* is written for application administrators who want to use Vibe Data Stream for Machine Data to move data from one or more points of generation to multiple target systems for processing. The guide assumes that you have a basic understanding of messaging concepts.

Informatica Resources

Informatica My Support Portal

As an Informatica customer, the first step in reaching out to Informatica is through the Informatica My Support Portal at https://mysupport.informatica.com. The My Support Portal is the largest online data integration collaboration platform with over 100,000 Informatica customers and partners worldwide.

As a member, you can:

- Access all of your Informatica resources in one place.
- Review your support cases.
- Search the Knowledge Base, find product documentation, access how-to documents, and watch support videos.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Documentation

The Informatica Documentation team makes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from https://mysupport.informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. You can access the PAMs on the Informatica My Support Portal at https://mysupport.informatica.com.

Informatica Web Site

You can access the Informatica corporate web site at https://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica How-To Library

As an Informatica customer, you can access the Informatica How-To Library at https://mysupport.informatica.com. The How-To Library is a collection of resources to help you learn more about Informatica products and features. It includes articles and interactive demonstrations that provide solutions to common problems, compare features and behaviors, and guide you through performing specific real-world tasks.

Informatica Knowledge Base

As an Informatica customer, you can access the Informatica Knowledge Base at https://mysupport.informatica.com. Use the Knowledge Base to search for documented solutions to known technical issues about Informatica products. You can also find answers to frequently asked questions, technical white papers, and technical tips. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@informatica.com.

Informatica Support YouTube Channel

You can access the Informatica Support YouTube channel at http://www.youtube.com/user/INFASupport. The Informatica Support YouTube channel includes videos about solutions that guide you through performing specific tasks. If you have questions, comments, or ideas about the Informatica Support YouTube channel, contact the Support YouTube team through email at supportvideos@informatica.com or send a tweet to @INFASupport.

Informatica Marketplace

The Informatica Marketplace is a forum where developers and partners can share solutions that augment, extend, or enhance data integration implementations. By leveraging any of the hundreds of solutions available on the Marketplace, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at http://www.informaticamarketplace.com.

Informatica Velocity

You can access Informatica Velocity at https://mysupport.informatica.com. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the Online Support.

Online Support requires a user name and password. You can request a user name and password at http://mysupport.informatica.com.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at http://www.informatica.com/us/services-and-training/support-services/global-support-centers/.

CHAPTER 1

Introduction to Informatica Vibe Data Stream for Machine Data

This chapter includes the following topics:

Informatica Vibe Data Stream for Machine Data Overview

Informatica Vibe Data Stream for Machine Data (VDS) is a highly available, distributed, scalable, real-time application that collects and aggregates machine data. You can collect machine data from different types of sources, transform or process the data, and write it to different types of targets. VDS consists of source services that collect data from sources and target services that write data to targets.

You can use VDS to collect data from different types of sources, such as event logs, real-time logs, call detail records, TCP/UDP applications, Syslog sources, HTTP sources, WebSocket sources, and MQTT brokers.

You can stream data to different types of targets, such as a Hadoop Distributed File System (HDFS) cluster and Apache Cassandra. You can stream data to an Informatica PowerExchange for Ultra Messaging source to perform complex transformations and real-time data warehousing. You can also stream data to Informatica RulePoint source controller to process complex events in real time.

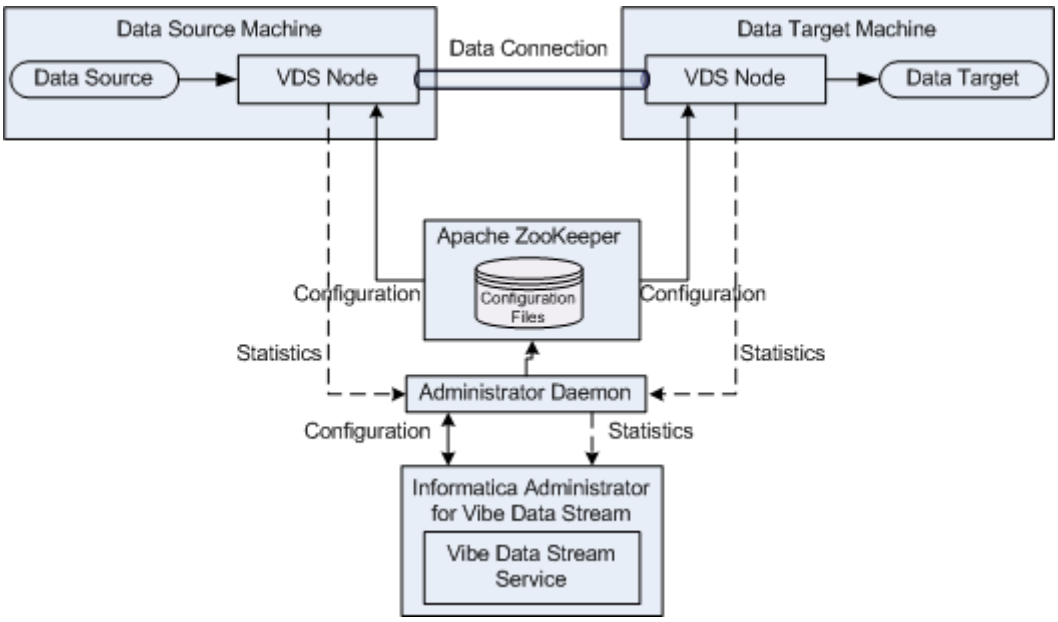
To gather operational intelligence from machine data or to perform real-time data warehousing, you need to collect and analyze the data before it becomes obsolete or corrupted. Use VDS to aggregate data from multiple sources in real time. If the data is in a form that is difficult to analyze, you can configure filters and transformations in VDS to prepare the data for analysis.

You can configure VDS for high availability so the processing fails over to a backup component when a primary component is unavailable. You can also use VDS to securely transfer data from sources to targets.

Vibe Data Stream Architecture

A Vibe Data Stream deployment consists of VDS Nodes, Apache Zookeeper, Administrator Daemon, and Informatica Administrator. These VDS components run on multiple host machines in a distributed application environment. You can install multiple components on a host machine.

The following image shows the components of a VDS deployment:



VDS consists of the following components:

VDS Node

A VDS Node is a process that contains one or more specialized threads. The threads work together to transfer data from a source to a target. You can create multiple VDS Nodes on a host machine. Install VDS Nodes on host machines on which you want to run a source service or target service. A VDS Node can contain multiple services and transformations. Services can include source services, target services, or both types of services. VDS uses a data connection to transport data from a source service to a target service.

Apache ZooKeeper

Apache ZooKeeper is a centralized service that maintains the data flow configuration information of the VDS Nodes in a deployment. You can deploy Apache ZooKeeper as a standalone instance or, for high availability and reliability, as a cluster called a ZooKeeper ensemble. When you start a VDS Node, it fetches data flow configuration information from ZooKeeper.

Administrator Daemon

The Administrator Daemon is a process that manages the deployment and undeployment of data flows and stores information about the data flows in a database. The daemon process also aggregates information about state and statistics from VDS Nodes.

Informatica Administrator for Vibe Data Stream (the Administrator tool)

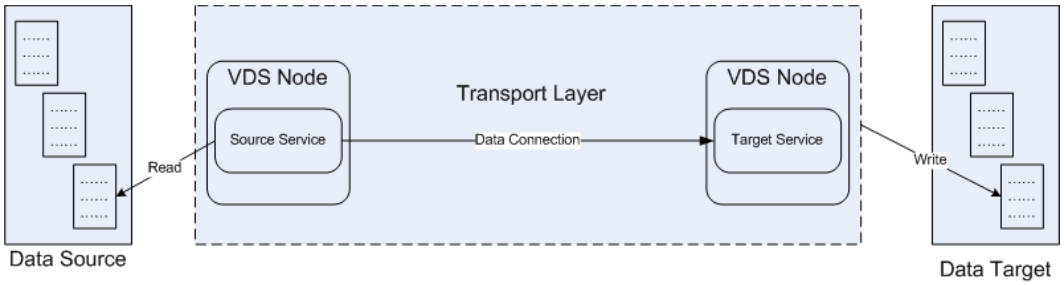
Informatica Administrator is a web application client that an administrator uses to manage the Vibe Data Stream Service and create users and user groups. The Vibe Data Stream Service is a service that you use to create, deploy, and undeploy data flows and to monitor data flow entities.

If you install the Administrator Daemon, the VDS Node, and Apache ZooKeeper in a PowerCenter setup, you can use the Administrator tool of PowerCenter and create and configure the Vibe Data Stream Service.

Data Flow Model

VDS uses a data connection to transport data from the source service to the target service. Use the Administrator tool to design the flow of data from the data source to the data target, deploy the data flow, and monitor the data flow.

The following image shows a simple VDS data flow:



The data flow consists of the following components:

Data source

You can read data from the following sources:

- Event logs
- Real-time logs
- Call detail records
- TCP/UDP applications
- Syslog sources
- MQTT brokers
- HTTP clients
- JMS providers
- Ultra Messaging applications
- WebSocket clients

VDS Node

The VDS Node runs on host machines on which you want to run a source service or a target service. A VDS Node can contain the following entities:

- Source service. A source service collects data from a data source and publishes the data. You can run a source service on a machine that either hosts the data source or can access it.
- Target service. A target service receives data from one or more source services and writes the data to a data target. You can run the target service on a machine that either hosts the data target or can access it.
- Aggregator. An aggregator collects data from source services or other aggregators and publishes the data to target services or other aggregators.

- Transformation. A process that transforms the data that a source service or aggregator publish or the data that a target service or aggregator receive, or both. A transformation runs on the VDS Node that contains the source service, the target service, or the aggregator.

Data target

You can write data to the following targets:

- Informatica PowerCenter
- Informatica RulePoint
- Hadoop Distributed File System (HDFS) cluster
- HTTP servers
- Amazon Kinesis streams
- Apache Kafka publish-subscribe messaging system broker
- Apache Cassandra
- JMS providers
- Ultra Messaging application
- WebSocket servers

Data connection

VDS Nodes use a data connection to transport data from a source service to a target service. When you design a data flow, you can choose one of the following data connections:

- Ultra Messaging. If you choose the Ultra Messaging data connection, VDS uses the Ultra Messaging or the publish/subscribe model to transport data.
- WebSocket. If you choose the WebSocket data connection, VDS uses the WebSocket protocol, which provides bi-directional communication over a single TCP connection.

Vibe Data Stream High-Level Process

Use the Administrator tool to create a Vibe Data Stream Service in the Administrator tool. Then, you can create a data flow, add source and target services to it, deploy the data flow and monitor it.

To create, deploy, and monitor a data flow, perform the following steps.

1. Log in to the Administrator tool with administrator privileges.
2. Create a Vibe Data Stream Service.
3. In the **Vibe Data Stream** view, create a data flow.
4. Add source services and target services.
5. Associate the source services and target services to nodes.
6. Connect the source services to the target services.
7. Optionally, add transformations to the connections.
8. Deploy and monitor the data flow.

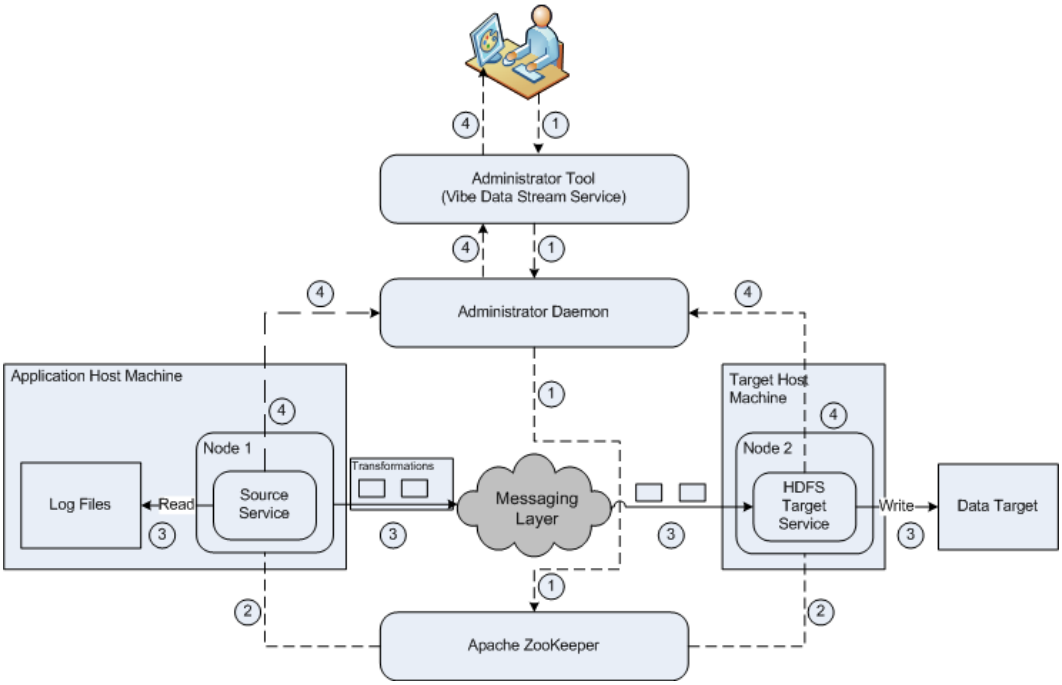
Vibe Data Stream Data Flow Process

You use the Administrator tool to design the flow of data from the data source to the data target and to deploy the data flow.

The Administrator Daemon pushes the data flow configuration information to Apache ZooKeeper. The VDS Nodes download the configuration information and start the source services and target services that the configuration specifies. Source services read data in blocks and publish messages through a data connection. Target services receive the data and write the data to a data target. The VDS Node monitors the entities in the data flow and sends information about state and statistics to the Administrator Daemon. The Administrator Daemon sends this information to the Administrator tool.

For example, an application writes log data to log files in the following directory: `/usr/app/logs/`. You want to transfer the data contained in the log files to an HDFS cluster. To transfer the data, install VDS Nodes on the application host machine and target host machine. As part of performing post-installation tasks, start a VDS Node Node1 on the application host and a VDS Node Node2 on the target host.

The following image shows how VDS works:



The image numbers the operations in the order of occurrence. The following steps describe the sequence of operations:

1. Use the Administrator tool to create and deploy a data flow. When you configure the data connection in the data flow, use the Ultra Messaging data connection. In the data flow, create a source service. Specify the source directory as `/usr/app/logs/`, and map the service to Node1. Create an HDFS target service and map the target service to Node2. Connect the source service to the target service, and add any transformations that you want to apply to the data. Finally, deploy the data flow. The Administrator Daemon sends the data flow configuration information to ZooKeeper.
2. The VDS Nodes download data flow configuration information from ZooKeeper. The VDS Node Node1 starts a source service. Similarly, Node2 starts a target service.

3. The source service reads data from the source files and publishes that data as messages on a topic. VDS applies the transformations that you added to the data flow. The target service subscribes to the topic, receives the data, and writes it to the HDFS cluster.
4. The VDS Node sends information about state and statistics to the Administrator Daemon. The Administrator Daemon publishes the information through the Vibe Data Stream Service. You can view the information on the **Monitoring** tab in the Administrator tool.

Related Topics:

- "Creating a Data Flow" on page 119

Vibe Data Stream User Interface

Use the Administrator tool to create the Vibe Data Stream Service, and to design, deploy, and monitor data flows.

You can use one of the following tabs in the user interface based on the task that you want to perform:

Domain

You can create the Vibe Data Stream Service.

Vibe Data Stream

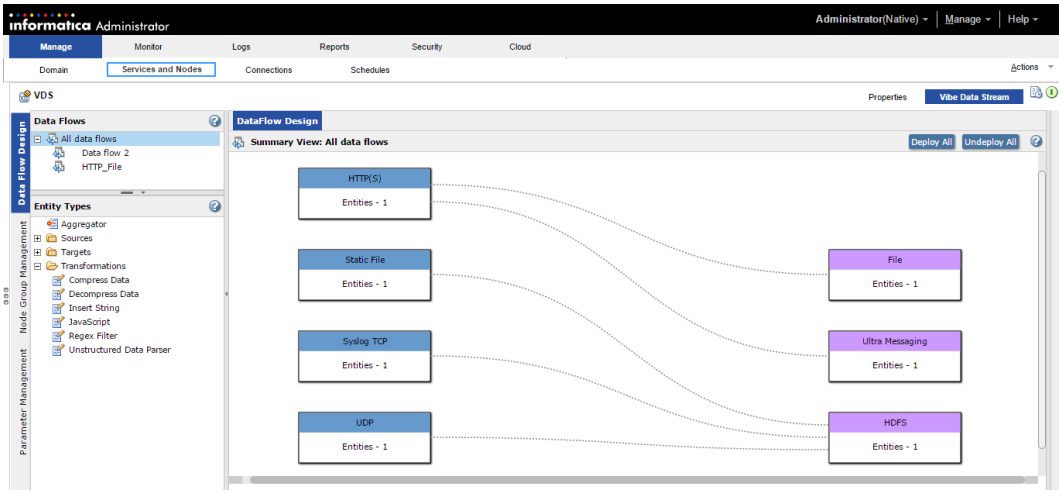
You can create, deploy, and undeploy data flows. You can also manage the properties of the source services, target services, and transformations in the data flows. The **Vibe Data Stream** tab includes the following tabs:

- Data Flow Design
- Node Group Management
- Parameter Management

Monitoring

You can monitor data flows, VDS Nodes, source services, target services, and aggregators. You can monitor the states of the entities and view statistics for source services, target services, and transformations.

The following image shows the **Vibe Data Stream** tab:



Example

You run the Information Technology (IT) department of a major firm that has thousands of employees. You want to collect real-time application monitoring data that includes application process logs from the computers of each of these employees.

The analysis of this data can be useful in many ways. Information about application usage helps you to manage IT resources optimally, reduce spending, and increase employee productivity by resolving issues. The application monitoring data is published by HTTP clients. You want to collect this data from the HTTP clients in real time and write it to HDFS for further processing.

You perform the following tasks:

1. In the Administrator tool, create the Vibe Data Stream Service.
2. Create a data flow with an HTTP source service and an HDFS target service.
3. Add an Insert String transformation that appends the IP address of the machine location.
4. Deploy the data flow. The HTTP source service receives data from an HTTP client that sends data through HTTP POST requests and sends the data through a data connection. The HDFS target service receives this data through the data connection.

CHAPTER 2

Licenses

This chapter includes the following topics:

Licenses Overview

VDS is available with an Enterprise license and a Free license.

The Enterprise license has the following features:

- High availability configuration for VDS entities
- Data usage limit each day based on your requirements
- Validity for an extended time period

The Free license has the following features:

- Unlimited data usage each day
- Validity for a limited time period

You can view the license details in the Administrator tool. VDS calculates the data usage every day and the Administrator tool displays warnings when the usage exceeds the limit. The Administrator tool also displays warnings 90, 60, 30, 15, and 7 days before the license expiration date.

License Expiry

When the license expires, the Vibe Data Stream Service becomes inaccessible. To access the Vibe Data Stream Service and redeploy the data flows, update the license.

Viewing the License Details

Use the Administrator tool to view license details. You might review license details to determine the options available for use.

To view license details, select the license in the **Domain Navigator**.

The Administrator tool displays the license properties in the following sections:

License Details

View license details on the **Properties** tab. The **Properties** tab shows license attributes, such as the license name, description, quota, and expiration date.

Service Options

View the service options on the **Options** tab. The **Options** tab shows the licensed services, such as the Vibe Data Stream Service.

Updating a License

Use a license key file to update the license in the Administrator tool. Place the license key file in a location that is accessible by the machine on which the Administrator tool is running. You can update a Free license to an Enterprise license. When you update the license, you must specify the location of the license key file.

Note: You must remove an existing license before you update it.

1. In the **Domain Navigator** of the Administrator tool, click **Actions** > **New** > **License**.

   The **Create License** dialog box appears.
2. Enter the following properties:

Property	Description
Name	Name of the license. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. Also, it cannot contain spaces or any of the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( )] [
Description	Description of the license. The description cannot exceed 765 characters.
Path	Path of the domain in which you create the license. Read-only field. Optionally, click **Choose Files** and select a domain in the **Select Folder** dialog box. Optionally, click **Create Folder** to create a folder for the domain.
License file	File containing the original key. Click **Browse** to find the file.

3. Click **OK**.
4. Select the license in the **Navigator**.
5. Click the **Assigned Services** tab.
6. In the **License** tab, click **Actions** > **Edit Assigned Services**.
7. Select the services under **Unassigned Services**, and click **Add**.
8. Click **OK**.
9. Select the **Vibe Data Stream** service in the **Domain Navigator** of the Administrator tool.
10. Click **Send license and database configuration to Admind** in the **Contents** panel to update the license.

    Note: Start the Administrator Daemon and Apache ZooKeeper before you apply the license.

Removing a License

Use the Administrator tool to remove a license.

1. Select the license in the **Domain Navigator** of the Administrator tool.
2. Click **Actions** > **Delete**.

CHAPTER 3

Using Informatica Administrator

This chapter includes the following topics:

Using Informatica Administrator Overview

Informatica Administrator (the Administrator tool) is a web application that you use to create the Vibe Data Stream Service, create and deploy data flows, and monitor VDS entities.

Use the Administrator tool to perform the following tasks:

- Domain administrative tasks. Manage logs, domain objects, and user permissions.
- Security administrative tasks. Manage users, groups, roles and privileges.

The Administrator tool has the following tabs:

- **Manage**. View and edit the properties of the domain and objects within the domain.
- **Monitor**. This tab is not applicable for VDS.
- **Logs**. View log events for the domain and services within the domain.
- **Reports**. Run a Web Services Report or License Management Report.

- **Security**. Manage users, groups, roles, and privileges.
- **Cloud**. View information about your Informatica Cloud organization.

The Administrator tool has the following header items:

Help

Access help for the current tab, determine the Informatica version, and configure the data usage policy.

Manage Tab - Domain View

The **Domain** view displays an overview of the domain and its contents. You can use the domain view to monitor the domain status, resource consumption, and events.

The **Domain** view has the following components:

- Domain Actions menu
- Contents panel
- Object Actions menus
- Service State Summary
- Resource usage indicators
- Manage tab Actions menu

Manage Tab - Services and Nodes View

The **Services and Nodes** view shows all application services and the node defined in the domain.

The **Services and Nodes** view has the following components:

Domain Navigator

Appears in the left pane of the **Domain** tab. The Navigator displays the following types of objects:

- Domain. You can view one domain, which is the highest object in the Navigator hierarchy.
- Application services. An application service represents server-based functionality. Select an application service to view information about the service and its processes.
- Node. A node represents a machine in the domain. You assign resources to a node and configure service processes to run on the node.
- License. View the license and the services assigned to the license.

Contents panel

Appears in the right pane of the **Domain** tab and displays information about the domain or domain object that you select in the Navigator.

Actions menu in the Navigator

When you select a domain object in the Navigator, you can create a folder, service, node, or license.

Actions menu on the Manage tab

When you select the domain in the Navigator, you can shut down or view logs for the domain.

When you select a node in the Navigator, you can remove a node association, recalculate the CPU profile benchmark, or shut down the node.

When you select a license in the Navigator, you can add an incremental key to the license.

Domain

You can view one domain in the **Services and Nodes** view on the **Domain** tab. It is the highest object in the Navigator hierarchy.

When you select the domain in the Navigator, the contents panel shows the following views and buttons:

- **Properties** view. View or edit domain resilience properties.
- **Resources** view. View available resources.
- **Permissions** view. View or edit group and user permissions on the domain.
- **Plug-ins** view. View plug-ins registered in the domain.
- **View Logs for Domain** button. View logs for the domain and services within the domain.

In the **Actions** menu on the **Manage** tab, you can shut down the domain, view logs, or access help on the current view.

Application Services

Application services are a group of services that represent Informatica server-based functionality.

In the **Services and Nodes** view on the **Domain** tab, you can create and manage the Vibe Data Stream Service.

Vibe Data Stream Service

Runs Informatica Vibe Data Stream for Machine Data in the Informatica domain. It manages the connections between service components and the users that have access to Informatica Vibe Data Stream for Machine Data.

When you select the Vibe Data Stream Service in the Navigator, the contents panel displays the following information:

- **Properties** view. View or edit the Vibe Data Stream Service properties.
- **Vibe Data Stream** view. Design and create data flows.
- **Actions** menu. Access help on the current view.

Logs Tab

You can view domain, application service, and user activity log events in the **Logs** tab of the Administrator tool. When you view log events in the **Logs** tab, the Log Manager displays the generated log event files in the log directory. When an error message appears in the Administrator tool, the error provides a link to the **Logs** tab.

On the **Logs** tab, you can view the following types of logs:

- Domain log. Domain log events are log events generated from the domain functions that the Service Manager performs.
- User activity log. User Activity log events monitor user activity in the domain.

- Service log. Events for the Vibe Data Service.

You can perform the following tasks on the **Logs** tab:

- View log events.
- Filter log event results.
- Save and purge log events.
- Copy log event rows.

Security Tab

You administer Informatica security on the **Security** tab of the Administrator tool.

The **Security** tab has the following components:

- **Search** section. Search for users, groups, or roles by name.
- **Navigator**. The Navigator appears in the left pane and display groups, users, and roles.
- **Contents** panel. The contents panel displays properties and options based on the object selected in the Navigator and the tab selected in the contents panel.
- **Security Actions** menu. Contains options to create or delete a group, user, or role.

Managing Your Account

Manage your account to change your password or edit user preferences.

If you have a native user account, you can change your password at any time with the Change Password application. If someone else created your user account, change your password the first time you log in to the Administrator tool.

User preferences control the options that appear in the Administrator tool when you log in. User preferences do not affect the options that appear when another user logs in to the Administrator tool.

Logging In

To log in to the Administrator tool, you must have a user account and the Access Informatica Administrator domain privilege.

1. Open Google Chrome.
2. In the Address field, enter the following URL for the Administrator tool login page:

   ```
   http://<host>:<port>/administrator/
   ```

   The Administrator tool login page appears.
3. Enter the user name and password.
4. Click **Log In**.

Password Management

You can change the password through the Change Password application.

You can open the Change Password application from the Administrator tool or with the following URL:
`http://<host>:<port>/passwordchange/`

Changing Your Password

Change the password for a native user account at any time. For a user account created by someone else, change the password the first time you log in to the Administrator tool.

1. In the Administrator tool header area, click **Manage** > **Change Password**.

   The Change Password application opens in a new browser window.
2. Enter the current password in the **Password** box, and the new password in the **New Password** and **Confirm Password** boxes.
3. Click **Update**.

Managing Users and Groups

To access the application services and objects in the Informatica domain and to use the application clients, you must have a user account.

During installation, a default administrator user account is created. Use the default administrator account to log in to the Informatica domain and manage application services, domain objects, and other user accounts. When you log in to the Informatica domain after installation, change the password to ensure security for the Informatica domain and applications.

User account management in Informatica involves the following key components:

Users

You can set up different types of user accounts in the Informatica domain. Users can perform tasks based on the roles, privileges, and permissions assigned to them.

Authentication

When a user logs in to an application client, the Service Manager authenticates the user account in the Informatica domain and verifies that the user can use the application client.

Groups

You can set up groups of users and assign different roles, privileges, and permissions to each group. The roles, privileges, and permissions assigned to the group determines the tasks that users in the group can perform within the Informatica domain.

Privileges and roles

Privileges determine the actions that users can perform in application clients. A role is a collection of privileges that you can assign to users and groups. You assign roles or privileges to users and groups for the domain and for application services in the domain.

Account lockout

You can configure account lockout to lock a user account when the user specifies an incorrect login in the Administrator tool. You can also unlock a user account.

Default Administrator

When you install Informatica services, the installer creates the default administrator with a user name and password you provide. You can use the default administrator account to initially log in to the Administrator tool.

The default administrator has administrator permissions and privileges on the domain and all application services.

The default administrator is a user account in the native security domain. You cannot create a default administrator. You cannot disable or modify the user name or privileges of the default administrator. You can change the default administrator password.

Managing Users

You can create, edit, and delete users depending on the type of license. You can assign roles, permissions, and privileges to a user account. The roles, permissions, and privileges assigned to the user determines the tasks the user can perform within the Informatica domain.

You can also unlock a user account.

Creating Users

Add, edit, or delete native users on the Security tab.

1. In the Administrator tool, click the Security tab.
2. On the Security Actions menu, click Create User.
3. Enter the following details for the user:

Property	Description
Login Name	Login name for the user account. The login name for a user account must be unique within the security domain to which it belongs. The name is not case sensitive and cannot exceed 128 characters. It cannot include a tab, newline character, or the following special characters: , + " \ < > ; / * % ? & The name can include an ASCII space character except for the first and last character. All other space characters are not allowed.
Password	Password for the user account. The password can be from 1 through 80 characters long.
Confirm Password	Enter the password again to confirm. You must retype the password. Do not copy and paste the password.
Full Name	Full name for the user account. The full name cannot include the following special characters: < > “

Property	Description
Description	Description of the user account. The description cannot exceed 765 characters or include the following special characters: < > “
Email	Email address for the user. The email address cannot include the following special characters: < > “ Enter the email address in the format UserName@Domain.
Phone	Telephone number for the user. The telephone number cannot include the following special characters: < > “

4. Click OK to save the user account.

   After you create a user account, the details panel displays the properties of the user account and the groups that the user is assigned to.

Unlocking a User Account

The domain administrator can unlock a user account that is locked out of the domain. If the user is a native user, the administrator can request that the user reset their password before logging back into the domain.

The user must have a valid email address configured in the domain to receive notifications when their account password has been reset.

1. In the Administrator tool, click the **Security** tab.
2. Click **Account Management**.
3. Select the users that you want to unlock.
4. Select **Unlock user and reset password** to generate a new password for the user after you unlock the account.

   The user receives the new password in an email.
5. Click the **Unlock selected users** button.

Managing Groups

You can create, edit, and delete groups in the native security domain.

You can assign roles, permissions, and privileges to a group. The roles, permissions, and privileges assigned to the group determines the tasks that users in the group can perform within the Informatica domain.

Adding a Native Group

Add, edit, or remove native groups on the Security tab.

A native group can contain user accounts or other native groups. You can create multiple levels of native groups. For example, the Finance group contains the AccountsPayable group which contains the OfficeSupplies group. The Finance group is the parent group of the AccountsPayable group and the

AccountsPayable group is the parent group of the OfficeSupplies group. Each group can contain other native groups.

1. In the Administrator tool, click the Security tab.
2. On the Security Actions menu, click Create Group.
3. Enter the following information for the group:

Property	Description
Name	Name of the group. The name is not case sensitive and cannot exceed 128 characters. It cannot include a tab, newline character, or the following special characters: , + " \ < > ; / * % ? The name can include an ASCII space character except for the first and last character. All other space characters are not allowed.
Parent Group	Group to which the new group belongs. If you select a native group before you click Create Group, the selected group is the parent group. Otherwise, Parent Group field displays Native indicating that the new group does not belong to a group.
Description	Description of the group. The group description cannot exceed 765 characters or include the following special characters: < > “

4. Click Browse to select a different parent group.

   You can create more than one level of groups and subgroups.
5. Click OK to save the group.

Managing Privileges

Privileges determine the actions that users can perform in application clients. Informatica provides domain privileges that determine actions that users can perform using the Administrator tool.

You assign privileges to users and groups for application services. You can assign different privileges to a user for each application service of the same service type.

You assign privileges to users and groups on the Security tab of the Administrator tool.

The Administrator tool organizes privileges into levels. A privilege is listed below the privilege that it includes. Some privileges include other privileges. When you assign a privilege to users and groups, the Administrator tool also assigns any included privileges.

Domain Privileges

Domain privileges determine the actions that users can perform in the Administrator tool.

The following table describes each domain privilege group:

Privilege Group	Description
Security Administration	Includes privileges to manage users, groups, roles, and privileges.
Domain Administration	Includes privileges to manage the domain, application services, and connections.
Monitoring	Includes privileges to monitor VDS deployments and view statistics.
Tools	Includes privileges to log in to the Administrator tool.

Roles

A role is a collection of privileges that you assign to a user or group. Each user within an organization has a specific role, whether the user is a developer, administrator, basic user, or advanced user.

You assign a role to users and groups for the domain and for application services in the domain.

Tip: If you organize users into groups and then assign roles and permissions to the groups, you can simplify user administration tasks. For example, if a user changes positions within the organization, move the user to another group. If a new user joins the organization, add the user to a group. The users inherit the roles and permissions assigned to the group. You do not need to reassign privileges, roles, and permissions.

Managing Roles

A role is a collection of privileges that you assign to a user or group. Each user within an organization has a specific role, whether the user is a developer, administrator, basic user, or advanced user.

You assign a role to users and groups for the domain and for application services in the domain.

Tip: If you organize users into groups and then assign roles and permissions to the groups, you can simplify user administration tasks. For example, if a user changes positions within the organization, move the user to another group. If a new user joins the organization, add the user to a group. The users inherit the roles and permissions assigned to the group. You do not need to reassign privileges, roles, and permissions.

You can assign the following types of roles:

- System-defined. Roles that you cannot edit or delete.
- Custom. Roles that you can create, edit, and delete.

A role includes privileges for the domain or an application service type. You assign roles to users or groups for the domain or for each application service in the domain.

VDS has the following types of roles:

- Administrator. This is a system-defined role that has privileges to administer the Administrator tool. With this role, you can create and manage user accounts, create the Vibe Data Stream Service and configure it.
- Operator. This is a custom role that has privileges to monitor VDS deployments.

When you select a role in the Roles section of the Navigator, you can view all users and groups that have been directly assigned the role for the domain and application services. You can view the role assignments by users and groups or by services. To navigate to a user or group listed in the Assignments section, right-click the user or group and select Navigate to Item.

You can search for system-defined and custom roles.

System-Defined Roles

A system-defined role is a role that you cannot edit or delete. The Administrator role is a system-defined role.

When you assign the Administrator role to a user or group for the domain or Vibe Data Stream Service, the user or group is granted all privileges for the service. The Administrator role bypasses permission checking. Users with the Administrator role can access all objects managed by the service.

Administrator Role

When you assign the Administrator role to a user or group for the domain or Vibe Data Stream Service, the user or group can complete some tasks that are determined by the Administrator role, not by privileges or permissions.

You can assign a user or group all privileges for the domain or Vibe Data Stream Service and then grant the user or group full permissions on all domain objects. However, this user or group cannot complete the tasks determined by the Administrator role.

For example, a user assigned the Administrator role for the domain can configure domain properties in the Administrator tool. A user assigned all domain privileges and permission on the domain cannot configure domain properties.

A user assigned the Administrator role for the domain or Vibe Data Stream Service can perform the following tasks:

- Configure domain properties.
- Grant permission on the domain
- Manage and purge log events.
- Receive domain alerts.
- View user activity log events.

Managing Custom Roles

You can create, edit, and delete custom roles.

Creating Custom Roles

When you create a custom role, you assign privileges to the role for the domain or for an application service type. A role can include privileges for one or more services.

1. In the Administrator tool, click the Security tab.
2. On the Security Actions menu, click Create Role.

   The Create Role dialog box appears.

3. Enter the following properties for the role:

Property	Description
Name	Name of the role. The role name is case insensitive and cannot exceed 128 characters. It cannot include a tab, newline character, or the following special characters: , + " \ < > ; / * % ? The name can include an ASCII space character except for the first and last character. All other space characters are not allowed.
Description	Description of the role. The description cannot exceed 765 characters or include a tab, newline character, or the following special characters: < > "

4. Click the Privileges tab.
5. Expand the domain or an application service type.
6. Select the privileges to assign to the role for the domain or application service type.
7. Click OK.

Assigning Privileges and Roles to Users and Groups

You determine the actions that users can perform by assigning the following items to users and groups:

- Privileges. A privilege determines the actions that users can perform in application clients.
- Roles. A role is a collection of privileges. When you assign a role to a user or group, you assign the collection of privileges belonging to the role.

Use the following rules and guidelines when you assign privileges and roles to users and groups:

- You assign privileges and roles to users and groups for the domain and for each application service that is running in the domain.
- You can assign different privileges and roles to a user or group for each application service of the same service type.
- A role can include privileges for the domain and multiple application service types. When you assign the role to a user or group for one application service, privileges for that application service type are assigned to the user or group.

If you change the privileges or roles assigned to a user, the changed privileges or roles take affect the next time the user logs in.

Note: You cannot edit the privileges or roles assigned to the default Administrator user account.

Steps to Assign Privileges and Roles to Users and Groups

You can assign privileges and roles to users and groups in the following ways:

- Navigate to a user or group and edit the privilege and role assignments.
- Drag roles to a user or group.

Assigning Privileges and Roles to a User or Group by Navigation

1. In the Administrator tool, click the Security tab.
2. In the Navigator, select a user or group.
3. Click the Privileges tab.

4. Click Edit.

   The Edit Roles and Privileges dialog box appears.
5. To assign roles, expand the domain or an application service on the Roles tab.
6. To grant roles, select the roles to assign to the user or group for the domain or application service.

   You can select any role that includes privileges for the selected domain or application service type.
7. To revoke roles, clear the roles assigned to the user or group.
8. Repeat steps 5 to 7 to assign roles for another service.
9. To assign privileges, click the Privileges tab.
10. Expand the domain or an application service.
11. To grant privileges, select the privileges to assign to the user or group for the domain or application service.
12. To revoke privileges, clear the privileges assigned to the user or group.

    You cannot revoke privileges inherited from a role or group.
13. Repeat steps 10 to 12 to assign privileges for another service.
14. Click OK.

Assigning Roles to a User or Group by Dragging

1. In the Administrator tool, click the Security tab.
2. In the Roles section of the Navigator, select the folder containing the roles you want to assign.
3. In the details panel, select the role you want to assign.

   You can use the Ctrl or Shift keys to select multiple roles.
4. Drag the selected roles to a user or group in the Users or Groups sections of the Navigator.

   The Assign Roles dialog box appears.
5. Select the domain or application services to which you want to assign the role.
6. Click OK.

Usage Collection Policy

VDS sends routine reports on data usage and system statistics to Informatica.

VDS reports the following data to Informatica:

- Maximum data usage across sources per day
- Average data usage

VDS uploads data to Informatica every day. Data collection and upload is enabled by default. You can choose to opt out of sharing data.

Disabling Informatica Data Usage

You can disable the upload of usage data from VDS in the Administrator tool.

1. In the Administrator tool, click **Help** > **About**.

2. Click **Usage Collection Policy**.
3. Clear **Enable Usage Collection**.
4. Click **OK**.

CHAPTER 4

Creating and Managing the Vibe Data Stream Service

This chapter includes the following topics:

Creating and Managing the Vibe Data Stream Service Overview

The Vibe Data Stream Service is an application service that you can use to create, deploy, and undeploy data flows and to monitor data flow entities.

Use the Administrator tool to create a Vibe Data Stream Service. Before you create the service, you must have a valid VDS license. The service stores information about data flows, license information, and data usage.

The type of VDS installation determines the database in which the service stores information. If you use the express installation to install VDS, the service stores information in the H2 database. If you use the custom installation to install VDS, you can configure one of the following databases for information storage:

- IBM DB2
- Microsoft SQL Server
- Oracle
- Sybase

After you create the Vibe Data Stream Service, you can create data flows, sources, targets, and transformations. You can monitor the VDS deployments on the **Monitoring** tab.

Creating the Vibe Data Stream Service

Create the Vibe Data Stream Service to create, manage, deploy and undeploy, and monitor data flows.

To create the Vibe Data Stream Service, you can use the Administrator tool or the Informatica command line program.

Creating the Vibe Data Stream Service in the Administrator Tool

Create the Vibe Data Stream Service in the Manage tab in Administrator tool and create, manage, deploy and undeploy, and monitor data flows.

Start the Administrator Daemon and Apache ZooKeeper before you create the service.

1. In the Administrator tool, click the **Services and Nodes** tab.
2. Click **Actions** > **New** > **Vibe Data Stream Service**.

   The **New Vibe Data Stream Service** window appears.
3. Enter the following general properties for the service:

   Name

   Name of the Vibe Data Stream Service. The name is not case sensitive and must be unique within the Informatica domain. The characters must be compatible with the code page of the associated repository. The name cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters:

   ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( )] [

   Description

   Description of the service. The description cannot exceed 765 characters.

   Location

   Name of the Informatica domain and folder where you want to create the service. Optionally, click **Browse** to select another folder in the Informatica domain.

   License

   License assigned to the service. Select from the list of VDS licenses available.

   Assign

   Node on which the service runs. Select **Single Node** to assign the service to a node.

   Node

   The host name or machine name where the service runs.
4. Click **Next**.
5. Enter the following database properties for the service:

   Database Type

   The type of database that contains the VDS repository.

   You can select one of the following databases:

   - DB2
   - Oracle
   - SQLServer
   - Sybase

   Kerberos Authentication

   Indicate that the database uses Kerberos authentication. This option is supported only for SQLServer.

   Username

   The database user.

Password

Database password corresponding to the database user.

JDBC Connect String

The connection string used to connect to the database.

- IBM DB2. `jdbc:informatica:db2://<host name>:<port>;DatabaseName=<database name>`
- Microsoft SQL Server. `Server: jdbc:informatica:sqlserver://<host name>:<port>;DatabaseName=<database name>`
- Kerberos-enabled Microsoft SQL Server. `jdbc:informatica:sqlserver://<host name>:<port>;DatabaseName=qadb;integratedSecurity=true;authenticationScheme=JavaKerberos;encrypt=false`
- Oracle. `jdbc:informatica:oracle://<host name>:<port>;ServceName=<service name>`
- Sybase. `jdbc:informatica:sybase://<host name>:<port>;DatabaseName=<database name>`

Secure JDBC Parameters

Secure JDBC parameters that you want to append to the database connection URL.

Use this property to specify secure connection parameters such as passwords. The Administrator tool does not display secure parameters or parameter values in the Vibe Data Stream Service properties. Enter the parameters as name=value pairs separated by semicolon characters (;). For example:

param1=value1;param2=value2

If secure communication is enabled for the database, enter the secure JDBC parameters in this property.

Database Schema

The schema name for the Microsoft SQL Server database.

Database Tablespace

Tablespace name for IBM DB2 database. When you specify the tablespace name, the Vibe Data Stream Service creates all repository tables in the same tablespace. You cannot use spaces in the tablespace name.

6. Click **Test Connection** to verify if the connection to the database is successful.
7. Click **Finish**.

Creating the Vibe Data Stream Service using Informatica Command Line Program

Use the infacmd command line program to create the Vibe Data Stream Service. You can access the command from the `<Installation Directory>/isp/bin` directory.

On UNIX, run the following command:

./infacmd.sh vds createService

On Windows, run the following command:

infacmd.bat vds createService

The command uses the following syntax:

```
<-DomainName|-dn> domain_name
        <-NodeName|-nn> node_name
        <-UserName|-un> user_name
```

```
<-Password|-pd> password
<-ServiceName|-sn> service_name
<-DbType|-dt> db_type (ORACLE, DB2, or SQLSERVER)
<-DbUser|-du> db_user
<-DbPassword|-dp> db_password
<-DbUrl|-dl> db_url
[<-DbDriver|-dr> db_driver]
[<-DbSchema|-ds> db_schema (used for SQL Server only)]
[<-DbTablespace|-db> db_tablespace (used for DB2 only)]
[<-SecureJDBCParameters|-sjdbcp> secure_jdbc_parameters]
<-licenseName|-lsn> license_name
```

Related Topics:

- "createService" on page 180

Editing the Vibe Data Stream Service

To edit a Vibe Data Stream Service, use the Administrator tool. Before you edit the service, verify that Administrator Daemon and Apache ZooKeeper are running.

1. In the Administrator tool, select the **Services and Nodes** tab.
2. Select the Vibe Data Stream Service in the **Services and Nodes** and click **Edit**.
3. Change values for the Vibe Data Stream Service general properties.
4. Change values for the database properties.

   Note: Before you make changes, back up the tables in the database. The Vibe Data Stream Service does not automatically back up the tables.
5. Click **Send license and database configuration to Admind** in the **Contents** panel to update changes.

CHAPTER 5

Vibe Data Stream Entity Types

This chapter includes the following topics:

Vibe Data Stream Entity Types Overview

The VDS entity types include aggregators, source service types, target service types, and transformation types. You can use built-in entity types or create and use custom entity types in data flows.

You can add the following built-in entity types to a data flow:

Source Services

Source service types read data from data streams and common data file formats.

Target Services

Target service types receive data from one or more source services, or aggregators, and write data to a data store.

Transformations

Transformation types transform data that a source service, another transformation, or aggregator publishes, or the data that a target service, another transformation, or aggregator receives, or both.

Aggregators

Aggregators collect data from source services or other aggregators and publish the data to target services or other aggregators.

When you add entities to data flows, specify the entity properties. You can also use parameters to set the properties for entities. After you add the entities to the data flows, map them to node groups.

Use built-in source service types, target service types, and transformation types for common use cases. If the built-in entity types do not meet your requirements, you can create and use custom entity types.

For information about creating custom entity types, see the *Vibe Data Stream for Machine Data Developer Guide*.

Aggregators

Aggregators collect data from source services or other aggregators and publish the data to target services or other aggregators. Aggregators read data in events. An event consists of one or more complete records. A record consists of data and can contain a delimiter. Specify the number of events that an aggregator can read when you configure it.

You can use aggregators in the following scenarios:

- To handle many-to-one and one-to many data flows effectively.
- To aggregate data from multiple source services and write the data to target services.
- To aggregate data from one or more aggregators and write the data to one or more aggregators through a firewall.
- To aggregate data from source services spread across geographies.

You can add one or more aggregators to a data flow. You can add transformations to aggregators. You can also view related statistics.

Aggregator Properties

Configure the properties of an aggregator when you add it to the data flow.

You can configure the following properties for an aggregator:

Entity Name

Name of the aggregator. Maximum length is 32 characters.

Description

Description of the aggregator. Maximum length is 256 characters.

Maximum Events to be Queued

Maximum number of events that the aggregator stores in an in-memory queue until the events are consumed by the target. When the queue is full, the aggregator blocks data and does not read from the source until the queue is free. You can specify a minimum value of 1000 and a maximum value of 1000000. Default is 100000.

Statistics

You can choose to view the following statistics for an aggregator:

- Bytes Received. The number of bytes of data received by the aggregator.
- Events Received. The number of events received by the aggregator.
- Receive Rate (Per Sec). The number of events received per second.
- Bytes Sent. The number of bytes of data sent by the aggregator.
- Events Sent. The number of events sent by the aggregator.
- Send Rate (Per Sec). The number of events sent per second.

- Events Reassigned. The number of events that the source service resends to the aggregator with a reassigned flag. The source service sets the reassigned flag if it does not receive acknowledgment from the aggregator. The statistic displays meaningful values when you select the load balancing messaging mode in the Ultra Messaging data connection or the acknowledgment mode in the WebSocket data connection while configuring data flows.
- Events to be Processed. Maximum number of messages that can be stored in the internal queue of the aggregator.

Built-in Source Service Types

You can use the built-in source service types to consume data from data streams and common data file formats. VDS includes the client libraries that a built-in source service type requires to consume data from a data source.

Source services read data in events. An event consists of one or more complete records. Specify the event size for a source service when you configure the source service. Event size defines the maximum length of data that a source service can read or receive at a time, and it determines the performance of a source service. The greater the event size, the more the messages that a source service can include in an event. The default event size is 8192 bytes. The length of the message includes the delimiter.

If a read operation that a source service performs returns an event that is equal to or less than the specified event size, the source publishes the event only if the records in the event contain a delimiter.

Source services optionally use an internal store to store events before sending them to the target services. This store is located on the machine on which the VDS Node that is associated with the source service is running. If there are network connectivity issues or the target services that the source service publish data to go down, the source services store the events in the internal store. When the network connectivity is restored or the target services come up, the source services retrieve data from the store and publish them again.

When you add a source service to data flow, you can configure the internal store options.

Note: If you select the **Streaming** messaging mode, the internal store of the source service is not supported.

VDS includes the following built-in source services:

- File
- HTTP(S)
- JMS
- MQTT
- Static File
- Syslog TCP
- Syslog UDP
- Syslog UDS
- TCP
- UDP
- Ultra Messaging
- WebSocket(S)

File Source Service Types

VDS includes two built-in source services to read from file sources.

When you read from file sources, you can use one of the following built-in source service types:

File Source Service Type

Use a File source service to read data from a directory that contains a source file or files that might change.

You can specify the file name or the Java regular expression pattern that applies to the convention used to name the source file. The File source reads the source file as it is being written and monitors the directory for new files matching the regular expression pattern.

Static File Source Service Type

Use a Static File source service to read data from a directory that contains a source file or files that do not change. When multiple files are processed in parallel, data from different files is interleaved.

When you configure the Static File source service properties, you can specify the file name or the Java regular expression pattern that applies to the convention used to name the source file. You can also configure the number of threads that are required to process the files and the waiting time. The waiting time is the amount of time to the Static File source service waits before reading data from the active file.

VDS creates a position file that tracks where the data has been read until. By default, VDS saves the position file in the directory that contains the source file or files. You can also specify a different directory when you configure the properties for the source service.

For more information about Java regular expressions, see the Java regular expression documentation.

File Rollover

In an application, the file rollover process closes the current file and creates a new file on the basis of file size or time. An active file is a file to which the source application writes data. Archived files are the rolled over files.

A file source service tracks the active file and consumes data in real time. If archive files exist when you start the source service, the source service starts reading from the oldest archive file that matches the specified regular expression. The source service then progresses toward the active file in chronological order.

If the chronological order of all files are the same, the file source service processes the files in lexical order. For example, if you name the files log.1, log.2, and log.10, the file source service processes the files in the order log.1, log.10, and log.2.

If you want the file source service to process the files in rollover order and you use the regular expression `log \..*`, you can name the files based on the rollover precision. For example, if the maximum backup index is 1000, you can name the files log.0001, log.0002, and log.0010. Or, if the maximum backup index is 10, you can name the files log.01, log.02, and log.10. In this case, though the source service processes the files in lexical order, it maintains the rollover order.

Note: If a rollover event occurs when the source service that processes the files is down, you might lose data. To avoid data loss, when you configure the source application for log file rollover, verify that the time between two rollover events exceeds maintenance periods.

You can use the File source service in the following rollover scenarios:

Name of the active file does not change after rollover

In this rollover scenario, an application writes data to only one file, for example, `alog.log`. When the rollover occurs, based on the criteria, the application renames the file as `alog.log.1` and continues to

write to `alog.log`. In this scenario, configure the File source service to read from `alog.log` file. Specify the following properties for the File source service:

- Name. Name for the file source service.
- Directory. Path to the directory where the source files are located.
- File Name. Name of the active file.
- Delimiter. Delimiter that separates entries in the source file.

In this rollover scenario, when you undeploy the data flow or stop the VDS Node mapped to the file source, the application continues to write to the active file `alog.log`. When rollover occurs, the application creates new files. When you redeploy the data flow or start the VDS Node, the source service continues to read from the `alog.log` file and the data in the rolled over files is not read.

Note: When the data flow is deployed and the source service is reading from the `alog.log` file, if the application performs a rapid rollover, then the data in the rolled over files is not read.

Name of the active file changes

In this rollover scenario, an application writes data to a new file and the File source service reads from that file. For example, if the application writes data to a file named `alog.log-<timestamp1>`, after a time based rollover the application writes to a new file named `alog.log-<timestamp2>`. In this scenario, configure the File source service to read from a file that changes according to a predefined pattern. Specify the following properties for the File source service:

- Name. Name for the file source service.
- Directory. Path to the directory where the source files are located.
- File Name. Name of the active file.
- File Name is Regular Expression. Java Regular expression that applies to the convention used for naming the active source file.
- Delimiter. Delimiter that separates entries in the source file.

File Source Service Properties

To add a file source service, use the File source service type.

You can configure the following properties for the File source service type:

Entity Name

Name of the source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Directory

Full path to the directory that contains the source files.

Position File Directory

Full path to the directory where VDS writes the position file. Specify this directory if the source file directory does not have write permissions.

File Name

Name of the source file or the Java regular expression that applies to the convention used to name the active source file.

File Name is Regular Expression

Indicates that the file name is a Java regular expression.

Processed File Directory

Optional. Location where the files should be moved after they are processed. Specify the location if the file name is a Java regular expression.

If you do not specify a location, the files will not be moved.

Delimiter

A character or sequence of characters that separates the data.

You can choose one of the following delimiters:

- LF. Line feed character sequence.
- CRLF. Carriage return line feed character sequence.
- Custom. Custom character sequence.

Default is LF.

Custom Delimiter

Custom character sequence that you want to specify as delimiter for the file source services.

Enter a custom character sequence if you select **Custom** as the delimiter.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

You can monitor the following statistics:

- Bytes Sent. The number of bytes sent by the source service.
- Events Sent. The number of events sent by the source service.
- Events to be Sent. The number of events that the source service is yet to send.
- Events not Delivered. Number of events that the source service did not deliver.
- Send Rate (Per Sec.). The number of bytes sent every second.
- Events Dropped. The number of events dropped while processing the source data.
- Files Written to Target. The number of files sent to the target service by the source service.
- Files to be Processed. The number of files to be processed. This number includes the file that is being processed and the pending files.
- Error while Moving Files. The number of errors that occur while moving the files.

Static File Source Properties

To add a static file source service, use the Static File source service type.

You can configure the following properties for the Static File source service type:

Entity Name

Name of the source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Directory

Full path to the directory that contains the source files.

Position File Directory

Full path to the directory where VDS writes the position file. Specify this directory if the source file directory does not have write permissions.

File Name

Name of the source file or the Java regular expression that applies to the convention used to name the active source file.

File Name is Regular Expression

Indicates that the file name is a Java regular expression.

Number of Threads

The number of threads used to process the files in parallel. Default is 5.

Specify at a minimum value of 1.

Waiting Time (seconds)

The time in seconds from file modification after which the file should be processed. Specify a waiting time that is greater than the time taken by the application to write to the active file.

Default is 5 seconds.

Processed File Directory

Optional. Location where the files should be moved after they are processed.

If you do not specify a location, the files will not be moved.

Note: The source service moves all files except the file that it is reading from, to the processed file directory, irrespective of whether it has sent the files to the target service or not.

Delimiter

End-of-line character sequence that marks the end of a line in the source file.

You can choose one of the following delimiters:

- LF. Line feed character sequence.
- CRLF. Carriage return line feed character sequence.
- EOF. End-of-file (EOF) marker.
- Custom. Custom character sequence.

Default is LF.

Custom Delimiter

Custom character sequence that you want to specify as delimiter for the file source services.

Enter a custom character sequence if you select **Custom** as the delimiter.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default value is 8192. Minimum value is 1.

Note: If you select **EOF** as the delimiter, specify an event size that is greater than the file size. If the source file naming convention uses a regular expression, specify an event size that is greater than the size of all files that match the regular expression.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.

- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

You can collect and monitor the following statistics for the Static File source service:

- Bytes Sent. The number of bytes sent by the source service.
- Events Sent. The number of events sent by the source service.
- Events to be Sent. The number of events that the source service is yet to send.
- Send Rate (Per Sec.). The number of bytes sent every second.
- Events Dropped. The number of events dropped while processing the source data.
- Events not Delivered. Number of events that the source service did not deliver.
- Files Written to Target. The number of files sent to the target service by the source service.
- Files to be Processed. The number of files to be processed. This number includes the file that is being processed and the pending files.
- Error while Moving Files. The number of errors that occured while moving the files.
- File Ignored. The number of files ignored by the source service. For example, if the delimiter is EOF and the value of the Maximum Event Size property is less than the size of the file, this statistic is updated.

HTTP Source Service Type

The HTTP source service type receives data from an HTTP client that sends data through HTTP POST requests. To create an HTTP source service, use the HTTP source service type.

Note: The HTTP source service type does not support HTTP GET requests.

When you configure the source service properties, you can configure either a secure or a nonsecure connection type. If you configure a secure connection, you must specify the location and password to the keystore file.

HTTP Source Service Properties

The HTTP source service accepts requests that use the HTTP protocol. The source service stores the paths in the VDS configuration file.

You can configure the following properties for the HTTP or HTTPS source service type:

Entity Name

Name of the HTTP source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Connection Type

You can select one of the following connection types:

- HTTP
- HTTPS

HTTP POST Path or HTTPS POST Path

Path of the HTTP or HTTPS POST requests that you want the HTTP service to receive.

Enter the path as the path appears in the POST requests. Enter the path in the following format:

```
myapp/path
```

KeyStore Path

The directory that contains the keystore file. Specify the absolute path to file.

You must specify this property if you select the **HTTPS** connection type.

Keystore Password

Password for the keystore file.

You must specify this property if you select the **HTTPS** connection type.

Select a secure parameter if you have already created it. To specify a key and value, click **Click to add secure parameters**.

Allow Prefix Path

Indicate that the POST path is a prefix.

Status Check Path

Path to which an HTTP GET request is sent to verify if the server is running.

Port

Port on which to listen for incoming connections.

Idle State Timeout (seconds)

Time after which the connection is closed when there is no incoming data. Specify a value of 0 to disable timeout.

Synchronous Response

Select one of the following types of responses:

- -. Disable synchronous response.
- Target ack based. The HTTP client gets a response after the target service sends an acknowledgment (UM Consumption report).
- HTTP response based. The HTTP client gets an HTTP response with the same session ID that the source service includes in the VDS event header.

Note: If the target service is not running when you deploy the data flow, it receives duplicate messages when it comes up later.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

You can monitor the following statistics for the HTTP source service:

- Bytes Sent
- Events Sent
- Events to be Sent
- Send Rate (Per Sec)
- Events Dropped
- Events to be Processed
- Events not Delivered

JMS Source Service Type

Use a JMS source service type to read data from a JMS provider.

You can configure the following properties for a JMS source service type:

Entity Name

Name of the source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

JNDI Context Factory

The JMS provider specific initial JNDI context factory implementation for connecting to the JNDI service. This value is a fully qualified class name of the Initial Context Factory.

For example, the class name of the Initial Context Factory for ActiveMQ is

```
org.apache.activemq.jndi.ActiveMQInitialContextFactory
```

For more information, see the documentation of the JMS provider.

JMS Connection Factory

The name of the object in the JNDI server that enables the JMS Client to create JMS connections.

For example, `jms/QCF` or `jmsSalesSystem`.

Connection URL

The location and port of the JMS provider on which to connect. For example:

`tcp://jndiserverA:61616`

User Name

User name to the connection factory.

Password

The password of the user account that you use to connect to the connection factory.

Select a secure parameter if you have already created it. To specify a parameter name and value, click **Click to add secure parameters**.

Destination Type

The type of destination on which the source service receives JMS messages. You can select one of the following destinations:

- Topic. The source service receives JMS messages on a topic.
- Queue. The source service receives JMS messages on a queue.

Durable Subscription Name

Durable subscriptions can receive messages sent while the subscribers are not active. Durable subscriptions provide the flexibility and reliability of queues, but still allow clients to send messages to many recipients. Specify the subscription name if the destination type is topic.

Destination

Name of the queue or topic on the JMS Provider as defined in the JMS Connection Factory created by the JMS Administrator. The source service receives JMS messages from this queue or topic.

Message Type

Type of message that the source service receives. Select one of the following message types:

- Byte
- Text
- Map

Acknowledgment Mode

Specifies the acknowledgement mode for non-transacted sessions. You can select one of the following acknowledgment modes:

- Auto Acknowledge. The session automatically acknowledges a message when a client receives it.
- Client Acknowledge. A client acknowledges a message by calling the message's acknowledge method.

Client ID

Client identifier to identify the connection.

Selector

Optional. Criteria for filtering message header or message properties, to limit which JMS messages the source service receives.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

You can collect and monitor the following statistics for the JMS source service:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events to be Sent. Number of events that the source service is yet to send.
- Send Rate (Per Sec). Number of bytes sent every second.
- Events not Delivered. The number of events not delivered by the source service.

MQTT Source Service Type

Use an MQTT source service to read data from an MQ Telemetry Transport (MQTT) broker. To create an MQTT source service, use the MQTT source service type. The MQTT source service type does not include a

user name property and a password property. An MQTT source service does not get authenticated by the MQTT broker when it retrieves data from the broker.

If multiple MQTT source services connect to an MQTT broker, each connection must have a unique identifier. The MQTT source service type includes a **Client ID** property that you can configure for the connections to the MQTT broker. An MQTT source service needs a persistence mechanism to store messages that it reads from an MQTT broker when they are being processed. The MQTT source service stores the messages in a persistence store, and uses the Client ID as the identifier for the persistence store.

Note: If two MQTT source services have the same client ID, one MQTT source service gets disconnected when the other source connects to the MQTT broker.

In a data flow in which an MQTT source service writes data to a target service, the source service first writes the data to an internal queue.

You can configure the following properties for the MQTT source service type:

Entity Name

Name of the MQTT source service. Maximum length

is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Server URL

URL of the MQTT broker or server to which to connect to receive messages. The URL is of the form `tcp://<IP_address>:<port>`.

Topics

Names of the topics to which to subscribe. The MQTT source service supports the topic wildcards that the MQTT specification describes. You can enter a comma-separated list of topic names.

For information about supported wildcards, see Wildcard documentation

Client ID

Optional. Unique identifier that identifies the connection between the MQTT source service and MQTT broker, and the file-based persistence store that the MQTT source service uses to store messages when they are being processed.

Enter a string of any length.

Maximum Enqueued Messages

Maximum number of messages that can be stored in the persistence store.

Default is 100000 messages.

Maximum Messages in a Batch

Maximum number of messages that are sent by the persistence store in a batch, to a target service.

Default is 100 messages.

No. of Retries to Add Messages to Persistent Store

Number of times that the MQTT source service tries to add messages to the internal queue if the internal queue is full.

Default is 3.

Retry Interval

The time in milliseconds between retries.

Default is 10 milliseconds.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

You can collect and monitor the following statistics for the MQTT source service:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events to be Sent. Number of events that the source service is yet to send.
- Events not Delivered. Number of events that the source service did not deliver.
- Send Rate (Per Sec). Number of bytes sent every second.
- Events Dropped. Events dropped by the source service while processing the source data. This statistic increases when the length of the message that is read is greater than the event size.
- Events to be Processed. Maximum number of messages that can be stored in the internal queue of the source.

Syslog Source Service Type

Use a Syslog source service to consume Syslog messages. To create a Syslog source service, use the Syslog source service type.

When you read from syslog sources, you can use one of the following built-in source service types:

Syslog TCP

Use a Syslog TCP source service to read from sources that use TCP protocol to send messages.

The Syslog TCP source service type expects the data source to use the `\n` delimiter. If the data source uses any of the other delimiters described in RFC 3164, "The BSD syslog Protocol" or RFC 6587, "Transmission of Syslog Messages over TCP," use the TCP source service type, instead. For more information about the delimiters described in RFC 3164 and RFC 6587, see The BSD syslog Protocol and Transmission of Syslog Messages over TCP.

Syslog UDP

Use a Syslog UDP source service to read from sources that use UDP protocol to send messages.

Syslog UDS

Use a Syslog UDS source service to read from sources that publish to a UNIX domain socket (UDS).

Syslog TCP Source Service Properties

To add a syslog TCP source service, use the Syslog TCP source service type.

You can configure the following properties for the Syslog TCP source service type:

Entity Name

Name of the Syslog TCP source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Port

TCP port that the source application uses to connect to the Syslog TCP source service.

Delimiter

Line feed character sequence that marks the end of a message in the TCP data stream.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

The statistics that you can choose to monitor for the Syslog TCP source. You can select the following statistics:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events to be Sent. Number of events that the source service is yet to send.
- Events not Delivered. Number of events that the source service did not deliver.
- Send Rate (Per Sec). Number of bytes sent every second.
- Events Dropped. The number of events dropped by the source service while processing the source data.
- Concurrent Connections. The number of Syslog TCP clients currently connected to the source service.
- Maximum Concurrent Clients. The maximum number of Syslog TCP clients connected to the source service since the time the source service is up.

Syslog UDP Source Service Properties

To add a syslog UDP source service, use the Syslog UDP source service type.

You can configure the following properties for the Syslog UDP source service type:

Entity Name

Name of the Syslog UDP source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Port

UDP port that the source application uses to connect to the Syslog UDP source service.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

The statistics that you can choose to monitor for the Syslog UDP source. You can select the following statistics:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events not Delivered. Number of events that the source service did not deliver.
- Events to be Sent. Number of events that the source service is yet to send.
- Send Rate (Per Sec). Number of bytes sent every second.

Syslog UDS Source Service Properties

To add a Syslog UDS source service, use the Syslog UDS source service type.

You can configure the following properties for the Syslog UDS source service type:

Entity Name

Name of the Syslog UDS source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Socket Address

The socket address that the source writes to. The Syslog UDS source service should have write access to the folder that contains the socket file. The socket file is used for communication between the Syslog client and server.

For example, if you specify the socket address as `/dev/log`, source service should have write access to the `/dev` directory so that it can create the `log` socket file.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

The statistics that you can choose to monitor for the Syslog UDS source. You can select the following statistics:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events to be Sent. Number of events that the source service is yet to send.
- Events not Delivered. Number of events that the source service did not deliver.
- Send Rate (Per Sec). Number of bytes sent every second.
- Events Dropped. The number of events dropped by the source service while processing the source data.

TCP Source Service Type

Use a TCP source service to consume data from an application that streams messages over the TCP protocol. To create a TCP source service, use the TCP source service type.

You can configure the following properties for the TCP source service type:

Entity Name

Name of the TCP source. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Port

TCP port that the source application uses to connect to the source service.

Delimiter

Character sequence that marks the end of a message in the TCP data stream. Choose a delimiter, or enter a different one.

You can choose one of the following built-in delimiters:

- LF. Line feed character sequence.
- CRLF. Carriage return line feed character sequence.
- STX-ETX. Start-of-text and end-of-text character sequences used in ISO/IEC 2022 systems.
- NUL. NULL character.
- Raw. The data stream consists of raw data without a delimiter.
- L1. One-byte integer that indicates the length of the message.
- L2. Two-byte integer that indicates the length of the message.
- L4. Four-byte integer that indicates the length of the message.
- Fixed Length. The data stream consists of records of the length that you specify in the **Length** box. In a message, the TCP source service adds as many complete records as the message can accommodate. The source service does not add incomplete records.
- Custom. String of your choice.

Default is LF.

Length

Length of each record that is read, in bytes. Specify the length when the source data stream consists of records of a fixed length. The maximum value is 51200.

Custom Delimiter

Custom character sequence that you want to specify as delimiter for the file source services.

Enter a custom character sequence if you select Custom as the delimiter.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

You can collect and monitor the following statistics for the TCP source service:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events to be Sent. Number of events that the source service is yet to send.
- Events not Delivered. Number of events that the source service did not deliver.
- Send Rate (Per Sec). Number of bytes sent every second.
- Events Dropped. The number of events dropped by the source service while processing the source data.
- Concurrent Connections. The number of TCP clients currently connected to the source service.
- Maximum Concurrent Clients. The maximum number of TCP clients connected to the source service since the time the source service is up.

UDP Source Service Type

Use a UDP source service to consume data from an application that sends messages over the UDP protocol. To create a UDP source service, use the UDP source service type.

You can configure the following properties for the UDP source service type:

Entity Name

Name of the UDP source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Port

UDP port that the source application uses to connect to the VDS source service.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

The statistics that you can choose to monitor for the UDP source service. You can select the following statistics:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events to be Sent. Number of events that the source service is yet to send.
- Events not Delivered. Number of events that the source service did not deliver.
- Send Rate (Per Sec). Number of bytes sent every second.

Ultra Messaging Source Service Type

Use an Ultra Messaging source service to read data from a UM source application. To create an Ultra Messaging source service, use the Ultra Messaging source service type.

You can configure only one UM source on a node. If you try to configure a second UM source, there is a conflict in configuration and the second source service will not start.

If you have a data flow that contains a UM source service type, when you change the configuration of the source service, you must perform the following tasks for the changes to take effect:

1. Undeploy the data flow.
2. Stop the node.
3. Start the node.
4. Deploy the data flow.

Note: When you use an Ultra Messaging source service in a data flow, you can only configure an Ultra Messaging data connection for the data flow.

Ultra Messaging Source Type Properties

You can configure the following properties for the Ultra Messaging source service type:

Entity Name

Name for the Ultra Messaging target service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

UM Topic Name

Topic on which the UM sending application publishes messages.

UM XML Configuration

UM configurations in XML format that the service uses.

Maximum length is 4000 characters.

You can use the following sample configuration file to configure how the source service reads data:

```
<um-configuration version="1.0">
  <applications>
   <application name="AppName">
<contexts>
    <context name = "test">
      <options type="context">
         <option default-value="10.65.43.186:15380"  name="resolver_unicast_daemon"/>
         <option default-value="18999"  name="request_tcp_port"/>
      </options>
      <options type="receiver">
         <option default-value="27997" name="transport_lbtru_port_low"/>
         <option default-value="27999" name="transport_lbtru_port_high"/>
      </options>
 </context>
</contexts>
    </application>
    </applications>
</um-configuration>
```

For more information about UM configuration, see the *Ultra Messaging Configuration Guide*.

UM Application Name

UM sending application name that the source service uses to get the configuration.

Context Name

Context name that the source service uses to get the configuration. If you do not specify a name, the source service gets the configuration from a context with no name.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

The statistics that you can choose to monitor for the Ultra Messaging source service. You can select the following statistics:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events to be Sent. Number of events that the source service is yet to send.
- Events not Delivered. Number of events that the source service did not deliver.
- Send Rate (Per Sec). Number of bytes sent every second.

WebSocket Source Service Type

The WebSocket source service type receives data from a WebSocket client that sends data through WebSocket. To create a WebSocket source service, use the WebSocket source service type.

When you configure the source service properties, you can configure either a secure or a nonsecure connection type. If you configure a secure connection, you must specify the location and password to the keystore file.

WebSocket Source Service Properties

If you configure a WebSocket source service with a WebSocket path, the source service accepts requests that uses the WebSocket protocol. The source service stores the paths in the VDS configuration.

You can configure the following properties for the WebSocket source service type:

Entity Name

Name of the WebSocket source service. Maximum length is 32 characters.

Description

Description of the source service. Maximum length is 256 characters.

Connection Type

You can select one of the following connection types:

- WebSocket
- WebSocket Secure

WebSocket Path or WebSocket Secure Path

Path of the WebSocket requests that you want the source service to receive.

Enter the path as the path appears in the POST requests. Enter the path in the following format:

```
myapp/path
```

KeyStore Path

The directory that contains the keystore file. Specify the absolute path to the file.

You must specify this property if you select the **WebSocket Secure** connection type.

KeyStore Password

Password for the keystore file.

You must specify this property if you select the **WebSocket Secure** connection type.

Select a secure parameter if you have already created it. To specify a key and value, click **Click to add secure parameters**.

Allow Prefix Path

Specify that the path is a prefix.

Status Check Path

Path to which an HTTP GET request is sent to verify if the server is running.

Port

Port on which the source service listens for incoming connections.

Idle State Timeout (seconds)

Time after which the connection is closed when there is no incoming data. Specify a value of 0 to disable timeout.

Maximum Event Size

Maximum length of data that the source service can read at a time, in bytes. Default is 8192. Minimum value is 1. Maximum value is 51200.

Retry on Failure

Indicates that the source service should try to open or read from the source if the operation fails the first time.

Number of Retries

The number of times the source service should retry to open or read from the source.

Delay between Retries

Time in seconds between retries.

Persist Data

Indicates that the source service should store data in the persistent store.

Persistence Options

You can configure the following persistence options:

- `batchSizeInBytes`. The batch size of the events in bytes after which the data is written to the persistent store. Specify a value of 0 if you want to write the data to the persistent store immediately. Default is 262144 bytes.
- `flushInterval`. The interval in milliseconds after which the source services writes data to the persistent store. Default is 5000 milliseconds.
- `maxDataFiles`. The maximum number of data files that the source service can keep in the persistent store. After the number of files exceed this number, a rollover occurs. Default is 10.
- `maxDataFileSize`. The maximum size of each data file in bytes. When the file reaches the specified size, a file rollover occurs. Default is 1073741824.
- `maxQueueSizeInBytes`. The maximum size of data that is unsent or has not received acknowledgment in bytes that can be stored in the persistent store. If the specified size is exceeded, the source service does not read data from the source. If you specify a value of 0, the unsent data can be as much as the value of `maxDataFiles` times `maxDataFileSize`. Default is unlimited size.

Statistics

You can collect and monitor the following statistics for the WebSocket source service:

- Bytes Sent. Number of bytes sent by the source service.
- Events Sent. Number of events sent by the source service.
- Events to be Sent. Number of events that the source service is yet to send.
- Events not Delivered. Number of events that the source service did not deliver.
- Send Rate (Per Sec). Number of bytes sent every second.
- Events Dropped. The number of events dropped by the source service while processing the source data.
- Events to be Processed. Maximum number of messages that can be stored in the internal queue of the source.

Built-in Target Service Types

Use built-in target service types to write data to a data store or to stream data to Informatica RulePoint and Informatica PowerCenter.

VDS includes the client libraries that a target service requires to write data to the targets, with the exception of the Hadoop Distributed File System (HDFS) client libraries. You install the HDFS distribution that you want and set an environment variable that VDS can use to find the client libraries for the HDFS distribution.

VDS includes the following target service types:

- Cassandra
- File
- HDFS
- HTTP(S)
- JMS
- Kafka
- Kinesis
- PowerCenter
- RulePoint
- Ultra Messaging
- WebSocket(S)

Cassandra Target Service Type

Use a Cassandra target service to write data to an Apache Cassandra database. To create a Cassandra target service, use the Cassandra target service type. You can use a JSON object as input with a Cassandra target service. The JSON object must be a valid and complete JSON object.

Each event received by Cassandra target should contain only one JSON record. If there is more than one record, only the first record is read and the other records are discarded.

The Cassandra target service does not support the incremental update of fields of type list, map and set.

Note: The **Retry on Failure** and **Number of Retries** properties are not applicable for the Cassandra target service.

Cassandra Target Service Properties

To add a Cassandra target service, use the Cassandra target service type.

You can configure the following properties for the Cassandra target service type:

Entity Name

Name of the Cassandra target service. Maximum length is 32 characters.

Description

Description of the target service. Maximum length is 256 characters.

Contact Point(s)

URI of the Cassandra database. Use the following URI format:

```
<hostname(s)>:<port>
```

Where:

- `hostnames` is a comma-separated list of the names or IP addresses of the hosts in the Cassandra cluster. In non-production environments, you might have only one host name or IP address.
- `port` is the port number on which the Cassandra database listens for connections.
  Optional. You can omit the port number, if the Cassandra database listens for connections on the default port number. Default is 9042.

Keyspace

Keyspace to use when writing to the database.

Table Name

Table name to use when writing to the database.

User Name

The user name of the Cassandra database.

Password

Optional. The password of the user account that you use to connect to the Cassandra keyspace.

Select a secure parameter if you have already created it. To specify a key and value, click **Click to add secure parameters**.

Note: If you want to specify an empty password, do not enter any value for **Password**.

Statistics

The statistics that you collect and monitor for the Cassandra target. You can select the following statistics:

- Bytes Received. The number of bytes the target service receives.
- Events Received. The number of events the target service receives.
- Receive Rate (Per Sec). The number of bytes the target service receives every second.
- Events Reassigned. The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.
- Events with Invalid JSON Objects. The number of events that contain JSON fields that do not match the data type of the columns in the Cassandra database.
- Events with Excess JSON Fields. The number of events that contain JSON fields that do not have a corresponding column in the Cassandra database.
- Events Written. The number of events written to the Cassandra database.
- Events Not Written. The number of events that were not written to the Cassandra database.

File Target Service Type

Use a file target service to receive data from source services and write the data to a flat file. To add a file target service, use the File Target service type. You can configure the target service for target file rollover. You can also perform advanced configurations to avoid data loss in high availability and load balancing deployments.

Related Topics:

- "Advanced Configuration for Entities" on page 95

File Rollover

In an application, the file rollover process closes the current file and creates a new file on the basis of file size or time.

When you write to a file target, you can perform the following types of rollover:

Size-based rollover

You can configure a target service to perform target file rollover when it reaches a certain size. To configure size-based rollover, specify the **Rollover Size** and **Date and Time Format** properties of the target service.

For example, set the rollover size as 1 MB, name of the file target as `target.log`, and a date and time format of your choice. If the source service sends 5 MB to the file target, the file target first creates the `target.log.<timestamp>` file. When the size of `target.log.<timestamp>` reaches 1 MB, the target service rolls the file over.

Time-based rollover

You can configure a target service to perform target file rollover when a certain period of time has elapsed. To configure time-based rollover, specify the **Rollover Time** property of the target service.

You can implement both size-based rollover and time-based rollover for a target file, in which case the event that occurs first triggers a rollover. For example, if you set rollover time as 1 hour and rollover size as 1 GB, the target service rolls the file over when the file reaches a size of 1 GB even if the 1-hour period has not elapsed.

The target file to which the target service is writing data is the active file. Files that the target service closes after performing a rollover are archived files.

File Target Service Properties

The following table describes the properties that you need to configure for the file target service type:

Entity Name

Name of the target service. Maximum length is 32 characters.

Description

Description of the target service. Maximum length is 256 characters.

Output File Directory

Full path to the directory that contains the target files.

File Name

Name of the target file. The following special characters are not allowed:

* : ? " < > | / \

Maximum length is 200 characters.

Append timestamp to filename?

Indicate whether the timestamp must be appended to the file name.

Date and Time Format

Date and time format to append to the rollover file name. The following special characters are not allowed:

* : ? " < > | / \

For example, the following date string represents 3:00 pm on December 07, 2014 if you specify the format as `<yyyy>-<MM>-<dd>-<HH>-<mm>-<ss>`:

2014-12-07-15-00-00

For more information about the date and time format, see the Java SimpleDateFormat documentation.

Rollover Size (MB)

Optional. Target file size, in megabytes (MB), at which to trigger rollover. A value of zero (0) means that the target service does not roll the file over based on size. Default is 10.

Rollover Time (Hrs)

Optional. Length of time, in hours, to keep a target file active. After the time period has elapsed, the target service rolls the file over. A value of zero (0) means that the target service does not roll the file over based on time. Default is 0.

Buffer Size (KB)

The size of the data that the target service flushes to the file system.

Receive Idle Events

Indicate whether the target service should flush the data to the file system if the size of the data is less than the buffer size.

Time Out (Seconds)

Time in seconds after which the target service flushes the data to the file system.

Retry on Failure

Indicates that the target service should try to write to the target if the operation fails the first time.

Number of Retries

The number of times the target service should retry to write to the source.

Delay between Retries

Time in seconds between retries.

Statistics

You can choose to monitor the following statistics for the File target service:

- Bytes Received. The number of bytes the target service receives.
- Events Received. The number of events the target service receives.
- Receive Rate (Per Sec). The number of bytes the target service receives every second.
- Events Reassigned. The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.
- Files Rolled Over. Number of files rolled over.

HDFS Target Service

Use a Hadoop Distributed File System (HDFS) target service to write data to HDFS. To create an HDFS target service, use the HDFS target service type. You can configure the target service for target file rollover. You can also perform advanced configurations to avoid data loss in high availability and load balancing deployments.

If HDFS is Kerberos enabled, create the `hdfs` super user principal. Ensure that the Hadoop users have a Kerberos principal or keytab to get the Kerberos credentials that are required to access the cluster and use the Hadoop services.

Before you deploy a data flow that uses HDFS target services, perform the following tasks:

- Install the HDFS distribution that you want and set an environment variable that VDS can use to find the client libraries for the HDFS distribution. Verify that you have the client libraries installed in the path where the VDS Node is running.
  For example, if you have a Cloudera Distribution, download the libraries from the Cloudera Downloads page.
- Set an environment variable on each host on which an HDFS target service runs. The environment variable must point to the Hadoop base directory. The environment variable is of the form `HADOOPBASEDIR=<Hadoop_Home_Directory>`. For example, `HADOOPBASEDIR=/usr/hadoop-2.0.2-alpha`.

Note: The **Retry on Failure** and **Number of Retries** properties are not applicable for the HDFS target service.

For more information about product requirements and supported platforms, see the Product Availability Matrix on the Informatica My Support Portal:
https://mysupport.informatica.com/community/my-support/product-availability-matrices

Related Topics:

- "Advanced Configuration for Entities" on page 95

Target File Rollover

When you write to an HDFS target, you can perform the following types of rollover:

Size-based rollover

You can configure an HDFS target service to perform target file rollover when the target file reaches a certain size. To configure size-based rollover, specify the **Rollover Size** property of the target service.

Time-based rollover

You can configure an HDFS target service to perform target file rollover when a certain period of time has elapsed after the target service creates the target file. To configure time-based rollover, specify the **Rollover Time** property of the target service.

You can implement both rollover schemes for a target file, in which case, the event that occurs first triggers a rollover. For example, if you set rollover time to 1 hour and rollover size to 1 GB, the target service rolls the file over when the file reaches a size of 1 GB even if the 1-hour period has not elapsed.

The target file to which the target service is writing data is the active file. Files that the target service closes after performing a rollover are archived files. When the target service archives a file, it appends the timestamp to the file name. For example, set the rollover size as 1 MB and name of the file target as `target.log`. If the source service sends 5 MB to the file target, the file target first creates the `target.log.<timestamp>` file. When the size of `target.log.<timestamp>` reaches 1 MB, the target service rolls the file over.

HDFS Target Service Properties

You can configure the following properties for the HDFS target service type:

Entity Name

Name of the HDFS target service. Maximum length is 32 characters.

Description

Description of the target service. Maximum length is 256 characters.

Destination

URI of the target file to which the target service writes data.

The HDFS target service type supports the following URI formats:

HDFS URI format

```
hdfs://<namenode-name>[:<port>]/<path>/<file-name>
```

MapR URI format

```
mapr://IP:port/path/filename
```

Where:

- `namenode-name` is the host name or IP address of the HDFS NameNode.
- `port` is the port number on which the HDFS NameNode listens for connections. You can omit the port number if you have configured HDFS to listen for connections on the default port number, 8020.
- `path` and `file-name` compose the location of the target file in the target file system.
  The URI format is suitable for a standalone HDFS target service. The URI is also suitable for an HDFS target service that runs on a node that is not part of a high availability setup. To use multiple target service instances for load balancing or high availability, use variables in the URI.

Security Mode

Indicates if HDFS has Kerberos authentication enabled. Select one of the following options:

- Secure
- Non Secure

User Principal

User principal to log in to the HDFS super user account. Specify a user principal in the following format:

```
user@DOMAIN.COM
```

Keytab Path

Location of the keytab files that HDFS uses.

Date and Time Format

The time stamp that is appended to the name of the file written to the target.

Rollover Size

Target file size, in gigabytes (GB), at which to trigger rollover. Default is 1.

Rollover Time

Length of time, in hours, to keep a target file active. After the time period has elapsed, the target service rolls the file over. A value of zero (0) means that the HDFS target service does not roll the file over based on time. Default is 0.

Force Synchronization

Flush the client's buffer to the disk device once every second. If you enable forceful synchronization, the data written by the target service is visible to other readers immediately. Forceful synchronization degrades the performance of VDS. For more information, see Hadoop documentation.

Default is to not synchronize forcefully.

Receive Idle Events

Indicate whether the target service should flush the data to the target file if the size of the data is less than the rollover size.

Time Out (Seconds)

Time in seconds after which the target service flushes the data to the target file.

Statistics

You can choose to monitor the following statistics for the HDFS target service:

- Bytes Received. The number of bytes the target service receives.
- Events Received. The number of events the target service receives.
- Receive Rate (Per Sec). The number of bytes the target service receives every second.
- Events Reassigned. The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.
- Files Rolled Over. Number of files rolled over.

Configuring the HDFS Target Service Type to Work With a High Availability HDFS

You can configure the HDFS target service to write to a highly available HDFS. Perform the following tasks before you add the HDFS target service to the data flow and deploy it:

1. Ensure that the HDFS client and HDFS server in the Hadoop distribution are of the same version.
2. Edit properties in the `/hadoop-<version>/etc/hadoop/hdfs-site.xml` file. The following code sample shows some of the properties that you can add:

```
<property>
  <name>dfs.nameservices</name>
  <value>NAMESERVICE1</value>
</property>
<property>
  <name>dfs.ha.namenodes.NAMESERVICE1</name>
  <value>MACHINE1,MACHINE2</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.NAMESERVICE1.MACHINE1/name>
  <value>MACHINE1:8020</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.NAMESERVICE1.MACHINE2</name>
  <value>MACHINE2:8020</value>
</property>
<property>
  <name>dfs.client.failover.proxy.provider.NAMESERVICE1</name>
  <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</
value>
</property>
```

3. When you configure the HDFS target service, specify the following URL in the **Destination** property:
`hdfs://NAMESERVICE1/user/http.txt`

HTTP Target Service Type

The HTTP target service type writes data to an HTTP server through HTTP POST requests. When you configure an HTTP target service, you must configure the URL for the HTTP server. When you configure the target service properties, you can configure a secure or a nonsecure connection type.

You can configure the following properties for the HTTP target service type:

Entity Name

Name for the HTTP target service. Maximum length is 32 characters.

Description

Description of the target service. Maximum length is 256 characters.

Connection Type

You can select one of the following connection types:

- HTTP
- HTTPS: Accept All Certificates
- HTTPS: Accept Certificates in TrustStore

Target URL

URL of the HTTP server.

Enter the path in the following format:

```
http://<server>:<port>/myapp/path
```

TrustStore Path

Path and file name of the Java truststore file. Specify the path if you select **HTTPS: Accept Certificates in TrustStore** connection type.

TrustStore Password

Password for the truststore file. Specify the password if you select **HTTPS: Accept Certificates in TrustStore** connection type.

Retry on Failure

Indicates that the target service should try to write to the target if the operation fails the first time.

Number of Retries

The number of times the target service should retry to write to the source.

Delay between Retries

Time in seconds between retries.

Statistics

You can monitor the following statistics for the HTTP target type:

- Bytes Received. The number of bytes the target service receives.
- Events Received. The number of events the target service receives.
- Receive Rate (Per Sec). The number of bytes the target service receives every second.
- Events Reassigned. The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.

JMS Target Service Type

Use a JMS target service type to receive data from source services and write data to a JMS provider.

You can configure the following properties for a JMS target service type:

Entity Name

Name of the source service. Maximum length is 32 characters.

Description

Description of the target service. Maximum length is 256 characters.

JNDI Context Factory

The JMS provider specific initial JNDI context factory implementation for connecting to the JNDI service. This value is a fully qualified class name of the Initial Context Factory.

For example, the class name of the Initial Context Factory for ActiveMQ is:

```
org.apache.activemq.jndi.ActiveMQInitialContextFactory
```

For more information, see the documentation of the JMS provider.

JMS Connection Factory

The name of the object in the JNDI server that enables the JMS Client to create JMS connections.

For example, `jms/QCF` or `jmsSalesSystem`.

Connection URL

The location and port of the JMS provider on which to connect. For example:

```
tcp://jndiserverA:61616
```

User Name

User name to the connection factory.

Password

The password of the user account that you use to connect to the connection factory.

Select a secure parameter if you have already created it. To specify a key and value, click **Click to add secure parameters**.

Statistics

You can collect and monitor the following statistics for the JMS target service:

- Bytes Received. The number of bytes the target service receives.
- Events Received. The number of events the target service receives.
- Receive Rate (Per Sec). The number of bytes the target service receives every second.
- Events Reassigned. The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.

Destination Type

The type of destination on which the target service sends JMS messages. You can select one of the following destinations:

- Topic. The target service sends JMS messages on a topic.
- Queue. The target service sends JMS messages on a queue.

Destination

Name of the queue or topic on the JMS Provider as defined in the JMS Connection Factory created by the JMS Administrator. The target service sends JMS messages to this queue or topic.

Priority Level

Priority of the message. The priority level ranges from 0 to 9.

Expiration Time(ms)

Time in seconds after which the message expires.

Message Type

Type of message that the target service sends. Select one of the following message types:

- Byte
- Text
- Map

Delivery Mode

Specifies the delivery mode for the messages. You can select one of the following delivery types:

- Persistent.
- Non-Persistent

Kafka Target Service Type

Use a Kafka target service to write data to an Apache Kafka publish-subscribe messaging system broker. Kafka runs as a cluster that comprises one or more servers, each of which is called a broker.

When you configure a Kafka target service, specify the topic on which it sends messages and the IP address and port on which the Kafka broker runs.

If you deploy a data flow that contains a Kafta target service, you might have duplication of messages when the VDS Node fails and restarts. The Kafka target service that is running on that VDS Node might receive duplicate messages from the source service when the node comes up later. This duplication of messages occurs because the source service does not receive an acknowledgment from the Kafka target service and resends the messages.

In a high availability setup, you might have duplication of messages when a VDS Node process fails over to one of the backup machines.

For more information about the Apache Kafka messaging system, see http://kafka.apache.org/documentation.html.

Kafka Target Service Properties

You can configure the following properties for the Kafka target service type:

Entity Name

Name for the Kafka target service. Maximum length is 32 characters.

Description

Description of the target service. Maximum length is 256 characters.

Kafka Destination

The IP address and port combination of the Kafka messaging system broker.

Topic

Topic on which the Kafka target service sends messages.

Statistics

You can monitor the following statistic for the Kafka target service type:

- Bytes Received
- Events Received
- Receive Rate (Per Sec)
- Events Reassigned. The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.

Amazon Kinesis Target Service Type

Use a Kinesis target service to receive data from source services and write the data to an Amazon Kinesis stream. To create a Kinesis target service, use the Kinesis target service type.

Before you use a Kinesis target service, perform the following tasks:

1. In the Amazon Kinesis console, create and configure an Amazon Kinesis stream.
2. In the Amazon Web Services (AWS) Identity and Access Management (IAM) service, create a user.
3. Download the access key and secret access key that are generated during the user creation process.
4. Associate the user with a group that has permissions to write to the Kinesis stream.

Supplying Partition Keys

The Kinesis target service requires partition keys to put data into the shards within a stream. The target service can generate random partition keys, but you can supply partition keys if you want to use specific strings. The target service scans the message headers for a key-value pair to use as a partition key.

You can supply partition keys in one of the following ways:

Insert a custom message header to use as a partition key.

Use the VDS API and create a custom transformation type to insert a key-value pair as a message header. When you configure the Kinesis target service, use the key of the custom header as the partition key. Add the transformation to the connection between the source and the Kinesis target. For more information about creating a custom transformation type, see *The Vibe Data Stream for Machine Data Developer Guide*.

Use the built-in IP address message header as a partition key.

VDS inserts a message header that stores the IP address of the host on which the source service runs. When you configure the Kinesis target service, specify the IP address as the partition key name.

Let the Kinesis target service generate partition keys.

If you use a dynamic partition key, the target service generates and uses a random partition key. The target service generates partition keys that are sufficiently random to ensure that the shards in the Kinesis stream receive an even distribution of data. When you configure the Kinesis target service, choose the random partition key name option.

Amazon Kinesis Target Service Properties

The Kinesis target service requires the stream name, a partition key, and the security credentials for the user you created in AWS IAM. The Kinesis target service implements the asynchronous client for accessing Amazon Kinesis.

A Kinesis target service uses a bounded queue and a fixed-size thread pool to write data to the Kinesis stream. The length of the queue and the size of the thread pool determine the throughput of the target service. You can configure the length of the queue and the size of the thread pool to control the throughput.

For more information about the thread pool size and queue length, see the Oracle Java ThreadPoolExecutor documentation.

You can configure the following properties for the Kinesis target service type:

Entity Name

Name of the Kinesis target service. Maximum length is 32 characters.

Description

Description of the target service. Maximum length is 256 characters.

Access Key ID

Access key ID that the AWS IAM service generates when you create a user.

Secret Access Key

Secret access key that the AWS IAM service generates when you create a user.

Stream Name

Name of the Kinesis stream to which to write data.

Thread Count

Optional. Number of threads in the thread pool. Default is 5. You can change the thread count if you want additional parallel requests.

This parameter initializes the `ExecutorService` in the Kinesis client.

Queue Length

Optional. The length of the queue that the target service uses. Default is 10000.

This parameter initializes the `ExecutorService` in the Kinesis client.

Partition Key Name

Optional. Name of the partition key in the message header. You can select the following partition keys:

- IP Address. To use the built-in IP address message header, choose **IP Address**.
- Custom Partition Key. To enter the name of a custom message header, choose **Custom Partition Key**.
- Dynamic Partition Key. If you want the target service to generate and use a random partition key, choose **Dynamic Partition Key**. If you select this option, all the shards in the stream are utilized.

Default is **Dynamic Partition Key**.

Custom Partition Key Name

Name of the custom message header to use as the partition key. Required if you choose to use a custom IP address message header as the partition key.

Retry on Failure

Indicates that the target service should try to write to the target if the operation fails the first time.

Number of Retries

The number of times the target service should retry to write to the source.

Delay between Retries

Time in seconds between retries.

Statistics

You can monitor the following statistics:

- Bytes Received. The number of bytes the target service receives.
- Events Received. The number of events the target service receives.
- Receive Rate (Per Sec). The number of bytes the target service receives every second.
- Events Reassigned. The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.

PowerCenter Target Service Type

Use a PowerCenter target service to stream data to Informatica PowerCenter. To create a PowerCenter target service, use the PowerCenter target service type. When you add a PowerCenter target service to a data flow, VDS generates a value for a PowerCenter target service property called **Receiver Type ID**. VDS also generates a value for a link property called **Transport Topic**. Add the PowerCenter target service to the data flow, get the values of those properties, and then configure PowerCenter.

You can configure the following property for the PowerCenter target service type:

Entity Name

Name of the PowerCenter target service. Maximum length is 32 characters.

Configuring PowerCenter

The PowerCenter target service is the data source for PowerCenter. The PowerCenter target service consumes the messages that a source service publishes and forwards the messages to PowerCenter. To consume the messages that a source service publishes, PowerCenter requires Informatica PowerExchange for Ultra Messaging. Install PowerExchange for Ultra Messaging and create the configuration files that it requires to read and write messages. For information about installing PowerExchange for Ultra Messaging, see *Informatica PowerExchange for Ultra Messaging User Guide for PowerCenter*.

After you create the configuration files, perform the following tasks:

1. Open the Ultra Messaging source and target configuration file.
2. In the receiver type options section of the configuration file, specify the receiver type ID that VDS assigned to the PowerCenter target service.
   The following lines from an Ultra Messaging source and target configuration file show how to specify the receiver type ID:

   ```
   <options type="receiver">
   <option name="umq_receiver_type_id" default-value="100"/>
   </options>
   ```

3. In PowerCenter, create an Ultra Messaging connection. When creating the connection, perform the following tasks:
   - In **Destination Name**, enter the topic name that VDS assigned to the connection.
   - In **Configuration File Full Path**, specify the Ultra Messaging source and target configuration file to which you added the receiver type ID of the PowerCenter target service.
   - From the **UM Service Level** list, select **Queueing**.

Related Topics:

- "Getting the Receiver Type ID of a Target Service" on page 127
- "Getting the Topic Name Assigned to a Connection" on page 127

RulePoint Target Service Type

Use a RulePoint target service to stream data to Informatica RulePoint. To create a RulePoint target service, use the RulePoint target service type. When you add a RulePoint target service to a data flow, VDS generates a value for a RulePoint target service property called **Receiver Type ID**. VDS also generates a value for a link property called **Transport Topic**. Add the RulePoint target service to the data flow, get the values of those properties, and then configure RulePoint.

You can configure the following property for the RulePoint target service type:

Entity Name

Name of the RulePoint target service. Maximum length is 32 characters.

Configuring RulePoint

After you create a RulePoint target service in VDS, to configure RulePoint to receive messages from VDS, perform the following tasks:

1. Log in to the RulePoint host, and then create a plain-text configuration file named `lbm.cfg` in the following location:
   `$RULEPOINT_HOME/service-conf/`
2. In the `lbm.cfg` file, specify the value of the receiver type ID property that VDS generated for the RulePoint target service, as follows:
   ```
   receiver umq_receiver_type_id <Receiver_Type_ID>
   ```
3. Log in to RulePoint Administrator, and then create an Ultra Messaging source. When creating the source, in the source topic box, specify the topic name that VDS generated for the link.

Related Topics:

- "Getting the Receiver Type ID of a Target Service" on page 127
- "Getting the Topic Name Assigned to a Connection" on page 127

Ultra Messaging Target Service Type

Use an Ultra Messaging target service to stream data to Informatica Ultra Messaging. To create an Ultra Messaging target service, use the Ultra Messaging target service type. When you add a Ultra Messaging target service to a data flow configured with the messaging mode as load balancing, VDS generates a value for the Ultra Messaging target service property called **Receiver Type ID**. VDS also generates a value for a link property called **Transport Topic**. Add the Ultra Messaging target service to the data flow, get the values of those properties, and then configure the UM receiver.

You can configure the following property for the Ultra Messaging target service type:

Entity Name

Name of the Ultra Messaging target service. Maximum length is 32 characters.

Configuring Ultra Messaging

After you create an Ultra Messaging target service in VDS, perform the following tasks to configure the UM receiver:

1. Log in to the UM host.

2. If you use load balancing messaging mode, in the `lbm.cfg` file, specify the value of the receiver type ID property that VDS generated for the UM target service, as follows:

   ```
   receiver umq_receiver_type_id <Receiver_Type_ID>
   ```

3. Start a UM receiver that listens on the topic name that VDS generated for the link.

WebSocket Target Service Type

The WebSocket target service type writes data to a WebSocket server through WebSocket messages. When you configure a WebSocket target service, you must configure the URL for the WebSocket server.

You can configure the following properties for the WebSocket target service type:

Entity Name

Name for the WebSocket target service. Maximum length is 32 characters.

Description

Description of the target service. Maximum length is 256 characters.

Connection Type

You can select one of the following connection types:

- WebSocket
- WebSocketSecure: Accept All Certificates
- WebSocketSecure: Accept Certificates in TrustStore

Target URL

URL of the WebSocket server.

Enter the path in the following format for a nonsecure connection:

```
ws://<server>:<port>/myapp/path
```

Enter the path in the following format for a secure connection:

```
wss://<server>:<port>/myapp/path
```

TrustStore Path

Path and file name of the Java truststore file.

TrustStore Password

Password for the truststore file.

Select a secure parameter if you have already created it. To specify a key and value, click **Click to add secure parameters**.

Statistics

You can collect and monitor the following statistics for the WebSocket target service:

- Bytes Received. The number of bytes the target service receives.
- Events Received. The number of events the target service receives.
- Receive Rate (Per Sec). The number of bytes the target service receives every second.
- Events Reassigned. The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.

Built-in Transformation Types

Add a transformation to a link to transform data. Use the built-in transformation types available in VDS for common transformation requirements.

The process of transformation in a data flow begins with the left-most transformation in the chain consuming the messages that the source service publishes. You can add a transformation to a link either on the source service or on the target service, or on an aggregator.

You can perform the following types of transformations in VDS:

- Transformation on records. Data can have multiple records and records are separated by delimiter. When you apply a transformation on the source service, every record is transformed.
- Transformations on events or chunks. An event or chunk has multiple records. When you apply transformation on the target service, the event or chunk is transformed.

Note: You cannot perform a transformation on an event or chunk if you add the transformations on the source service.

You can configure transformations to accept messages from one source service and forward transformed messages to multiple target services. You can configure a transformation to accept data from multiple sources and forward transformed messages to one or more target services.

VDS includes the following transformations:

- Compress Data
- Decompress Data
- Insert String
- JavaScript
- Regex Filter
- Unstructured Data Parser

Delimiters in Source Service Transformations

A delimiter separates records in the source file. If a message contains multiple records separated by a delimiter, VDS breaks the message down at each delimiter boundary to get individual records. VDS removes the delimiter from each record and passes the records to the transformation one record at a time. The transformation transforms a record and, based on whether or not a transformation follows it in the chain, forwards the record to either the next transformation or the target services. After forwarding a record, the transformation accepts the next record in the message.

If you want to replace the delimiter that VDS removes, specify a delimiter when you create a transformation. The number and choice of delimiters that you can use vary across the built-in transformations. In some transformations, you can insert custom delimiters while in other transformations, you cannot use custom delimiters and the number of built-in delimiters is restricted.

If you add multiple transformations on a connection, and you append a delimiter in more than one transformation, the data that the target data store receives contains all those delimiters in sequence. Therefore, either append a delimiter in only the final transformation or, if you append a delimiter in an intermediate transformation, do not append delimiters in subsequent transformations in the chain.

Guidelines for Adding Transformations

Consider the following guidelines when you add connections and transformations to data flows:

- You can add transformations to the source service, the target service, or the aggregator. To add a transformation, drag the transformation to the link and associate it with the entity name.
- You can add only one link between two entities, between two transformations, and between a transformation and an entity.
- You can add multiple transformations to a link.
- You cannot add a link between two transformations.
- If you have a transformation that is running on a target service, you cannot add another link between that transformation and a different target service.

Related Topics:

- "Adding Entities to a Data Flow" on page 120

Examples of Transformations in Data Flows

The examples show how you can add transformations in data flows.

One-to-one Data flow

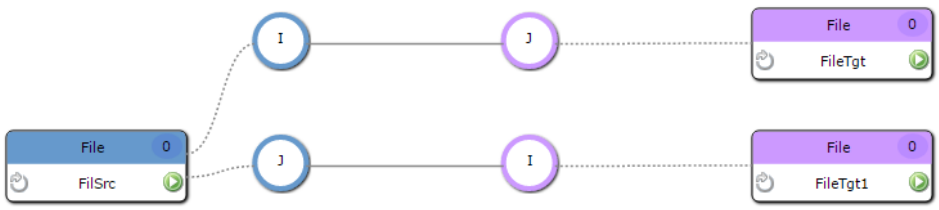
The following image shows a one-to-one data flow with one transformation each on the source service and on the target service:

The data flow publishes data in the following way:

1. The File source service publishes data in chunks to the Insert String transformation. The Insert String transformation breaks the chunks into records by using the delimiter on the File source service. The Insert String transformation then transforms each record and publishes the data on a topic.
2. The target service that listens on the topic, receives the data, and performs Javascript transform on each record.
3. The File target service then writes the transformed data to the file.

One-to-many Data Flow

The following image shows a one-to-many data flow:

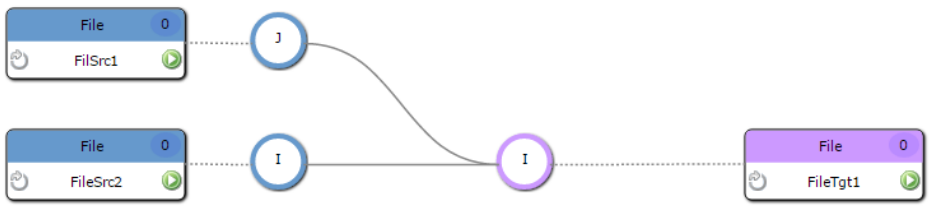


The data flow publishes data in the following way:

1. The File source service publishes data in chunks to the Insert String transformation. The Insert String transformation breaks the chunks into records by using the delimiter on the File source service. The Insert String transformation then transforms each record and publishes the data on a topic.
2. Each target service that listens on the topic, receives the data, and performs Javascript transformation and an Insert String transformation on each record.
3. The File target services then write the transformed data to the files.

Many-to-one Data Flow

The following image shows a many-to-one data flow:



The data flow publishes data in the following way:

1. The two File source services publish data in chunks to the Insert String transformation. The Insert String transformations breaks the chunks into records by using the delimiter on the File source services. The Insert String transformations then transform each record and publish the data on topics.
2. The File target service that listens on the topic, receives the data, and performs an Insert String transformation on each record.
3. The File target service then writes the transformed data to the file.

Compress Data Transformation Type

Use the Compress Data transformation type to compress data sent by source services.

You can apply a Compress Data transformation only on a source service or an aggregator. The Compress Data and the Decompress Data transformations work as a pair. Use these transformations to reduce bandwidth consumption between the source service and the target service. When you apply the transformations, the Compress Data transformation should be the last transformation on the source service and the Decompress Data transformation should be the first transformation on the target service.

The Compress Data transformation type supports the following standard compression algorithms to compress data:

- GZIP
- BZIP2
- XZ

Compress Data Transformation Type Properties

You can configure the following properties for the Compress Data transformation type:

Entity Name

Name of the transformation. Maximum length is 32 characters.

Description

Description of the transformation. Maximum length is 256 characters.

Compression Technique

Compression technique that the transformation type uses to compress data.

You can select one of the following compression algorithms:

- GZIP
- BZIP2
- XZ

Apply Transformation On

Select the source service, target service, or aggregator on which you want to apply the transformation.

Statistics

You can collect and monitor the following statistics:

- Events Received
- Bytes Received
- Events Generated
- Bytes Sent
- Time Taken for Transformation (Milliseconds)

Decompress Data Transformation Type

Use the Decompress Data transformation type to decompress data sent received by target services.

You can apply the Decompress Data transformation on a target service or an aggregator. The Compress Data and the Decompress Data work as a pair. When you apply these transformations, the Compress Data transformation is the last transformation on the source service and the Decompress Data is first transformation on the target service.

To decompress data, the Decompress Data transformation uses the same method that the Compress Data transformation uses.

Decompress Data Transformation Type Properties

You can configure the following property for the Decompress Data transformation type:

Entity Name

Name of the transformation. Maximum length is 32 characters.

Description

Description of the transformation. Maximum length is 256 characters.

Apply Transformation On

Select the source service, target service, or aggregator on which you want to apply the transformation.

Statistics

You can collect and monitor the following statistics:

- Events Received
- Bytes Received
- Events Generated
- Bytes Sent
- Time Taken for Transformation (Milliseconds)

Insert String Transformation Type

Use the Insert String transformation type to insert strings into a record. Insert strings by building an expression of tokens. An Insert String transformation does not perform conditional processing, so it transforms and forwards all the records that it receives.

Insert String Properties

You can configure the following properties for the Insert String transformation type:

Entity Name

Name of the transformation. Maximum length is 32 characters.

Description

Description of the transformation. Maximum length is 256 characters.

String Expression

Token expression that describes how to transform each record.

Apply Transformation On

Select the source service, target service, or aggregator on which you want to apply the transformation.

Statistics

You can collect and monitor the following statistics:

- Events Received
- Bytes Received
- Events Sent
- Bytes Sent
- Time Taken for Transformation (Milliseconds)

Tokens to Use to Insert a String

Tokens begin with the pound sign (#). Use a built-in token to add a time stamp, host name, or delimiter. Use the custom-string token to add a custom string to the record that you want to transform. VDS also provides a token for the input data, and you must include that token in your expression. Delimiters are optional.

Use the following tokens to construct an expression:

- `#HOSTNAME`. Name of the host on which the source service runs.
- `#TIMESTAMP`. System time, in milliseconds, recorded from the host that runs the source. The time stamp indicates the time at which the transformation transformed the record.
- `#DATA`. Data to transform.

- `#<custom_string>`. Custom string to insert.
- `#LF`. Line feed delimiter.
- `#CRLF`. Carriage return line feed delimiter.
- `#TIMEUUID`. Unique identifier to append or prepend to the event based on timestamp and IP address.
- `#RANDOMUUID`. Random unique identifier to append or prepend to the event.

You can insert a token multiple times in an expression. For example, you can insert the host name twice, as follows:

```
#HOSTNAME#DATA#HOSTNAME
```

VDS does not validate token expressions. Use the following rules to verify that the token expression is valid:

- The token must begin with the pound sign (`#`).
- Use only upper case characters in the token. For example, enter `#HOSTNAME`, and not `#hOSTNAME`. If you try to insert a host name with the token `#hOSTNAME`, VDS considers `hOSTNAME` as a custom string. Custom string tokens can contain both uppercase and lowercase characters. If all the characters in a string use uppercase, verify that the string does not match the strings in the built-in tokens, such as `HOSTNAME` or `TIMESTAMP`.
- Insert delimiter tokens at the end of your expression.
- Always include the `#DATA` token.
- Do not use an expression that includes only the `#DATA` token.

Note: If you specify an erroneous token expression for an Insert String transformation, it forwards the record to the targets without performing any transformation.

Sample Token Expressions

The following token expression prefixes the host name and the system time to the input data:

```
#HOSTNAME#TIMESTAMP#DATA
```

The following token expression appends the custom string `MyCustomString` to the input data:

```
#DATA#MyCustomString
```

The following example appends a line feed delimiter to the input data:

```
#DATA#LF
```

The following example appends a unique identifier based on timestamp and IP address to the input data:

```
#DATA#TIMEUUID
```

The following example appends a random unique identifier to the input data:

```
#DATA#RANDOMUUID
```

Note: If you are transforming raw data, you cannot predict where the strings that you specify in the token expression will appear in a line of data. If the source service breaks a line down into blocks, the transformation appends and prefixes strings to each block of raw data. After transformation, the line includes multiple instances of the prefixed and appended strings. For example, if you configure the Insert String transformation to prefix the time stamp and append the host name to data, and the source service breaks a line down into three data blocks, the transformed line includes multiple instances of the time stamp and host name strings, as follows:

```
<timestamp><first_line_fragment><hostname><timestamp><second_line_fragment><hostname><tim
estamp><third_line_fragment><hostname>
```

JavaScript Transformation Type

Use a JavaScript transformation to transform records and insert delimiters. To create a JavaScript transformation, add the JavaScript transformation type to a link.

In the JavaScript transformation, you specify a JavaScript function that applies the transformations that you want. You can include methods to check whether input records match a particular condition, and you can either drop or forward records on the basis of the results. If the JavaScript function is not valid, the transformation logs an error message and the data flow ceases.

The JavaScript function uses the Rhino 1.7R1 JavaScript engine. For more information about the Rhino 1.7R1 JavaScript engine, see https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino#Rhino_documentation.

JavaScript Transformation Properties

You can configure the following properties for the JavaScript transformation type:

Entity Name

Name of the JavaScript transformation. Maximum length is 32 characters.

Description

Description of the JavaScript transformation. Maximum length is 256 characters.

JavaScript Functions

JavaScript transformation program to transform records and insert delimiters. The program must include the filter function that applies the transformations. Maximum length is 4000 characters.

The JavaScript Transformation Function

The JavaScript transformation is associated with an input stream, from which it reads data, and an output stream, to which it writes data. Input data consists of records in byte array form. The output is a `ByteArrayOutputStream` object.

To transform the input data and write the transformed data to the output stream, in the Administrator tool, create a JavaScript program. You can use the default `transform` function. From the body of the `transform` function, call the methods that you require.

Use the `transform` function to perform the following tasks:

- Read operations from the input and write operations to the output without an intermediate byte array-to-string conversion. Use simple read and write operations if you want to only append or prepend strings to the input.
- Read the byte array from the input as a string and write the string to the output as a byte array. Use read and write operations that convert from and to a byte array form if you want to transform the data.
- Transform the record.

The `transform` function has the following signature:

```
function transform(input, output, utility){
     //TODO:Add your function code here
    }
```

The `transform` function accepts the following arguments:

input

Input data stream in byte array form.

output

`ByteArrayOutputStream` object to which you write transformed data as a byte array.

utility

Built-in utility class that includes methods for the read and write operations that you require to transform data and methods to append LF and CRLF delimiters.

The `utility` class includes the following Java methods:

- `getBytes(String str)`. Convert a text string to a byte array.
- `getString(byte[] bytes)`. Convert a byte array to a text string.
- `getBytesForLF()`. Get the raw byte sequence of the line feed delimiter.
- `getBytesForCRLF()`. Get the raw byte sequence of the carriage return line feed delimiter.

You can also use your own methods instead of the methods in the `utility` class. For example, you can add a method if you want to add delimiters other than the line feed and carriage return line feed delimiters. Define your own methods in the `transform` function, and then call the methods from the `transform` function, as shown in the following example:

```
function transform(input, output, utility)
{
//funtion body
// Call to custom function MyFunction1
   myFunction1();
// Call to custom function MyFunction2
   myFunction2();

   // Definition of custom function myFunction1
   function myFunction1(a,b){
   // myFunction1 body
   }
   // Definition of custom function myFunction2
function myFunction2(c,d){
   // myFunction2 body
   }
}
```

Apply Transformation On

Select the source service, target service, or aggregator on which you want to apply the transformation.

Statistics

You can collect and monitor the following statistics:

- Events Received
- Bytes Received
- Events Generated
- Bytes Sent
- Time Taken for Transformation (Milliseconds)

Note: If you want to validate the JavaScript program, you must use a third-party tool. VDS does not validate the JavaScript program. The JavaScript transformation does not support the `print`, `let` and, `of` functions.

Sample Function to Append Line Feed Delimiter

The following JavaScript function appends the line feed delimiter to each record:

```
function transform(input, output, utility)
{
            //Write the input to the output
         for(j = 0; j< input.length; j++){
       output.write(input[j]);
         }
         //Add a line feed delimiter
```

```
        output.write(utility.getBytesForLF());
}
```

Sample Function to Append String

The following JavaScript function appends a string `myString` to the input byte array:

```
function transform(input, output, utility)
{
var string = utility.getString(input);
var textToAdd = "myString ";
var newString = textToAdd.concat(string);
output.write(utility.getBytes(newString));
}
```

Regex Filter Transformation Type

Use a regular expression filter to publish entries that match a Java regular expression. To create a regular expression filter, use the Regex Filter transformation type. A regular expression filter publishes records that match the expression and drops records that do not match the expression.

You can configure the following properties for the Regex Filter transformation type:

Entity Name

Name of the regular expression filter. Maximum length is 32 characters.

Description

Description of the transformation. Maximum length is 256 characters.

Filter Expression

Java regular expression to apply to the line of text. For example, you can use the following regular expression:

```
^.*\[.*\].*$
```

For more information about regular expressions and examples, see http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html

Apply Transformation On

Select the source service, target service, or aggregator on which you want to apply the transformation.

Statistics

You can collect and monitor the following statistics:

- Events Received
- Bytes Received
- Events Generated
- Bytes Sent
- Time Taken for Transformation (Milliseconds)

Unstructured Data Parser Type

Use the Unstructured Data Parser transformation type to parse log files and other files. The Unstructured Data Parser uses a method similar to Java Grok to transform unstructured data input into structured JSON format. When you create the transformation you define how the data is transformed.

You can use the transformation to transform syslog log files, Apache logs, MySQL logs, and webserver logs. It works by parsing text patterns into output that matches the log data.

For more information about the Grok tool, see the Grok documentation.

Unstructured Data Parser Transformation Properties

You can configure the following properties for the Unstructured Data Parser transformation type:

Entity Name

Name for the transformation. Maximum length is 32 characters.

Description

Description of the transformation. Maximum length is 256 characters.

Pattern

Pattern that describes how to parse the data. The format must match the pattern defined in the Grok pattern file or the regular expression you specify in **Custom Regex**.

Custom Regex

Optional field. Regular expression that applies to any part of the input that is not defined in **Pattern**.

Apply Transformation On

Select the source service, target service, or aggregator on which you want to apply the transformation.

Statistics

You can collect and monitor the following statistics:

- Events Received
- Bytes Received
- Events Generated
- Bytes Sent
- Time Taken for Transformation (Milliseconds)

Pattern Examples

The following examples describe some patterns that you can configure:

Pattern Without Custom Regular Expression

If you want to transform the input `vds` into JSON format, configure the following properties:

- Entity Name. Enter a name for the transformation.
- Pattern. Enter `%{WORD:myword}`

When you deploy the data flow, you get the following output:

```
{"myword": [["vds"]]}
```

Pattern With Custom Regular Expression

If you want to transform the input `ABC` into JSON format, configure the following properties:

- Entity Name. Enter a name for the transformation.
- Pattern. Enter `%{ALLCAPS:capword}`
- Custom Regex. Enter `ALLCAPS=[A-Z]+`

When you deploy the data flow, you get the following output:

```
{"capword":"ABC"}
```

Pattern to Convert a Log Line

If you want to transform the log line `Jul 9 22:41:51 myserver sshd[4295]: Failed password for invalid user myuser from 220.113.135.154 port 55993 ssh2` into JSON format, configure the following properties:

- Entity Name. Enter a name for the transformation.
- Pattern. Enter `%{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} sshd\[%{BASE10NUM}\]: Failed password for (invalid user |)%{USERNAME:username} from %{IP:src_ip} port %{BASE10NUM:port} ssh2`

When you deploy the data flow, you get the following output:

```
{"BASE10NUM":4295,"HOUR":22,"MINUTE":41,"MONTH":"Jul","MONTHDAY":9,"SECOND":
51,"TIME":"22:41:51","host_target":"myserver","port":
55993,"src_ip":"220.113.135.154","timestamp":"Jul 9 22:41:51","username":"myuser"}
```

Pattern to Convert a Syslog Log Line

If you want to transform the log line `Oct 17 08:59:00 suod newsyslog[6215]: logfile turned over` into JSON format, configure the following properties:

- Entity Name. Enter a name for the transformation.
- Pattern. Enter `%{SYSLOGBASE} %{GREEDYDATA}`

When you deploy the data flow, you get the following output:

```
{"SYSLOGBASE":[["Oct 17 08:59:00 suod newsyslog[6215]:"]],"timestamp":[["Oct 17
08:59:00"]],"MONTH":[["Oct"]],"MONTHDAY":[["17"]],"TIME":[["08:59:00"]],"HOUR":
[["08"]],"MINUTE":[["59"]],"SECOND":[["00"]],"SYSLOGFACILITY":[[null]],"facility":
[[null]],"priority":[[null]],"logsource":[["suod"]],"IPORHOST":[["suod"]],"HOSTNAME":
[["suod"]],"IP":[[null]],"IPV6":[[null]],"IPV4":[[null]],"SYSLOGPROG":
[["newsyslog[6215]"]],"program":[["newsyslog"]],"pid":[["6215"]],"GREEDYDATA":[["logfile
turned over"]]}
```

Pattern to Convert an Apache Log Line

If you want to transform the log line `127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET / apache_pb.gif HTTP/1.0" 200 2326` into JSON format, configure the following properties:

- Entity Name. Enter a name for the transformation.
- Pattern. Enter `%{COMMONAPACHELOG}`

When you deploy the data flow, you get the following output:

```
{"COMMONAPACHELOG":[["127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET/
apache_pb.gifHTTP/1.0" 200 2326"]],"clientip":[["127.0.0.1"]],"HOSTNAME":
[["127.0.0.1"]],"IP":[[null]],"IPV6":[[null]],"IPV4":[[null]],"ident":[["-"]],"USERNAME":
[["-","frank"]],"auth":[["frank"]],"timestamp":[["10/Oct/2000:13:55:36 -0700"]],"MONTHDAY":
[["10"]],"MONTH":[["Oct"]],"YEAR":[["2000"]],"TIME":[["13:55:36"]],"HOUR":[["13"]],"MINUTE":
[["55"]],"SECOND":[["36"]],"INT":[["-0700"]],"verb":[["GET"]],"request":[["/
apache_pb.gif"]],"httpversion":[["1.0"]],"BASE10NUM":[["1.0","200","2326"]],"rawrequest":
[[null]],"response":[["200"]],"bytes":[["2326"]]}
```

Pattern to Convert a log4j Log Line

If you want to transform the log line `2015-03-03 11:35:53,759 [WARN ] [lbm:Thread-27] Core-5688-1883: timer returned error 5 [CoreApi-5688-3337: lbm_socket_sendb send/sendto:`

`(10049) The requested address is not valid in its context.` into JSON format, configure the following properties:

- Entity Name. Enter a name for the transformation.
- Pattern. Enter `%{TIMESTAMP_ISO8601} \[%{LOGLEVEL} %{GREEDYDATA}`

When you deploy the data flow, you get the following output:

```
{"TIMESTAMP_ISO8601":[["2015-03-03 11:35:53,759"]],"YEAR":[["2015"]],"MONTHNUM":
[["03"]],"MONTHDAY":[["03"]],"HOUR":[["11",null]],"MINUTE":[["35",null]],"SECOND":
[["53,759"]],"ISO8601_TIMEZONE":[[null]],"LOGLEVEL":[["WARN"]],"GREEDYDATA":[["]
[lbm:Thread-27] Core-5688-1883: timer returned error 5 [CoreApi-5688-3337: lbm_socket_sendb
send/sendto: (10049) The requested address is not valid in its context."]]}
```

Using Parameters in Entity Properties

A parameter represents a value that you can set for an entity property. You can create parameters for entity properties to provide flexibility each time you configure entities.

You can set the values for parameters in the **Parameter Management** tab. To use the parameters, specify the parameter name in the entity properties.

Use the following format to specify the parameter name in the entity properties:

```
${parametername}
```

Enclose the parameter name in the `{` and `}`parentheses and prefix with a `$`.

You can also configure secure parameters for passwords or other secure fields in the entity properties. You can configure secure parameters in the following ways:

- Set the values for secure parameters in the **Secure Parameters** tab.
- Set the value for a secure field in the entity properties, when you add the entity to the data flow.

Setting Values for Parameters

Set the values for parameters in the **Parameters** tab.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream service.
3. Select the **Vibe Data Stream** tab.
4. In the **Parameter Management** tab, select one of the following tabs:
   - **Parameters** tab. Use this tab to set nonsecure parameters and values.
   - **Secure Parameters** tab. Use this tab to set secure parameters and values. The values that you specify for the secure parameters appear masked.
5. Click **Add**.
6. In the **Key** field, enter a name for the parameter.

   The parameter name can contain uppercase letters A-Z, lowercase letters a-z, the numbers 0 though 9, and the special characters - _

   The parameter name cannot contain spaces.

7. In the **Value** field, enter a value for the parameter.
8. Click **Save**.

Related Topics:

- "Step 4: Set Parameters" on page 148

Examples

The following examples show how you can use parameters to configure entity properties:

Example 1

To use a parameter for a host name, configure the following parameter properties:

- Key. Specify `hostname` as the key.
- Value. Specify `informatica.com` as the value.

To use the parameter, specify `${hostname}` in the entity property. When you deploy the data flow, the value `informatica.com` is used.

Example 2

You can use parameters to define a URL. The following table describes keys and values that you can configure:

Key	Value
hostname	informatica.com
portnumber	8080
pathtofile	files

To use the parameters in a URL field in an entity property, specify `http://${hostname}:${portnumber}/${pathtofile}` in the entity property. When you deploy the data flow, the value `http://informatica.com:8080/files` is used.

Custom Entity Types

In addition to adding built-in entity types, you can develop your own source service types, target service types, and transformation types.

For more information about developing a custom entity type, see *The Vibe Data Stream for Machine Data Developer Guide*.

Advanced Configuration for Entities

Based on the features that you want to use, you might have to perform advanced configuration for the entities in the data flow. For example, if you configure load balancing or high availability in the data flow, you must use variables in the properties of the service.

Node Name Variable

Use the node name variable `#infa.nodename` to include the name of a node.

For example, an HDFS URI of the form `hdfs://<namenode-name>[:<port>]/<path>/<file-name>` is a static URI. If you deploy the HDFS target service on multiple VDS Nodes, to reduce the load on an HDFS target service, all the instances use the same URI. The instances try to write to the same file. Only one target service succeeds because, at any specified time, only one instance can acquire a lease to write to a file. This scenario results in data loss.

To enable all the instances to write to the HDFS file system, use the `#infa.nodename` variable in the `<file-name>` component of the HDFS URI. when configuring the HDFS target service,

The following example shows a URI that uses the `#infa.nodename` variable:

```
hdfs://MyNamenode/app_data/logs/#infa.nodename.log
```

`#infa.nodename` is the variable that VDS replaces with the VDS Node name.

The name of the file to which a particular instance writes changes with the name of the VDS Node on which the instance runs. Each target service writes to its own HDFS file. For example, the target service in a VDS Node called `node1` writes to the file `hdfs://MyNamenode/app_data/logs/node1.log` and the target service in a VDS Node called `node2` writes to the file `hdfs://MyNamenode/app_data/logs/node2.log`.

Time Stamp Variable

Use the time stamp variable `#infa.timestamp` when you want to configure high availability for an HDFS service.

The nodes in a high availability pair share the same node name. The HDFS target service instance that runs on one node becomes active and acquires a lease to write to the HDFS file. If an active service fails, a standby instance becomes active. The new active service has to wait several minutes for the HDFS NameNode node to close the file and recover the lease before it can write to the target file. This scenario results in data loss.

To enable the instance on the secondary node to write data to the HDFS file system, use the `#infa.timestamp` variable in the `<file-name>` component of the HDFS URI.

The following example shows a URI that uses the `#infa.timestamp` variable:

```
hdfs://MyNamenode/app_data/logs/#infa.timestamp.log
```

`#infa.timestamp` is the variable that VDS replaces with the system time in milliseconds.

The name of the file varies with system time. The primary service creates a file whose name is the current system time in milliseconds, and starts writing to the file. If an active service fails, another service does not have to wait for the HDFS NameNode node to revoke the lease on the file. The newly active service creates its own file with the current system time as its name and starts writing to the file.

Related Topics:

- "Configuring High Availability for Entities" on page 96

Node Name and Time Stamp Variables

Use the node name variable and time stamp variable to configure load balancing for a service and high availability for the load balanced service instances.

For example, use the following URI:

```
hdfs://MyNamenode/app_data/logs/#infa.nodename#infa.timestamp.log
```

`#infa.nodename` is the variable that VDS replaces with the VDS Node name.

`#infa.timestamp` is the variable that VDS replaces with the system time in milliseconds.

The name of the file varies with node name and system time.

Configuring High Availability for Entities

Configure high availability for an entity to ensure service continuity when the active entity goes down or you stop the node for administrative purposes. To configure high availability for an entity, start a VDS Node with the same name on two or more hosts. When you start a node with the same name on multiple hosts, the entity on one of the nodes becomes active. If you stop the node on which the active entity is running or if the primary node fails, the entity on one of the standby nodes becomes active.

Additional Configuration Tasks for Source Services

The following table describes additional tasks for configuring high availability for a particular type of source service:

Source Service Type	Additional Configuration Tasks
MQTT	No additional tasks.
Flat File, JSON	Configure the source application instances to log data to a file on a shared network drive and specify the shared location for the source service.
TCP	Configure the source application to connect to the new active node when fail over occurs.
Syslog	For information about how to set up high availability for a Syslog target service, contact Informatica Global Customer Support.
UDP	High availability is not supported.

Additional Configuration Tasks for Target Services

For some types of services, you might need to perform additional tasks to set up high availability.

The following table describes additional tasks for configuring high availability for a particular type of target service:

Target Service Type	Additional Configuration Tasks
Cassandra, PowerCenter, RulePoint	No additional tasks.
HDFS	Perform advanced configuration tasks for supporting high availability.

CHAPTER 6

Vibe Data Stream Nodes

This chapter includes the following topics:

Vibe Data Stream Nodes Overview

A VDS Node is a process that runs different entity types to transfer and process data.

A VDS Node can contain multiple services and transformations. Services can include source services, target services, or both types of services. Install VDS Nodes on host machines on which you want to run a source service or target service.

Node Groups

You can group multiple VDS Nodes in groups to simplify working with them. When you work with a large number of VDS Nodes, you can group them to manage and monitor them effectively.

You can create the following group types:

- Normal Node Group. You can manually add VDS Nodes to a Normal Node group or import from a CSV file.
- Dynamic Node Group. You can define a regular expression that describes the names of VDS Nodes to add to the group.

After you create node groups, associate them with entities when you create data flows.

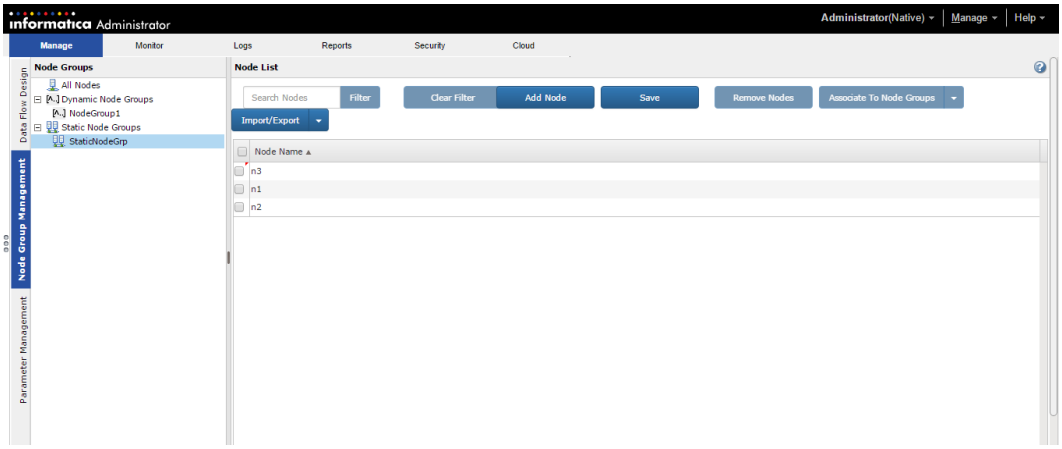
Node Group Management Tab

Use the **Node Group Management** tab to work with node groups.

The **Node Management** tab consists of the following panels:

- Node Groups
- Node List

The following image shows the **Node Management** tab:



Working with Node Groups

When you work with node groups, you can perform the following tasks:

1. Create a node group.
2. Associate VDS Nodes to node groups.
3. Import node groups.
4. Export node groups.
5. Import nodes to a node group.
6. Export nodes in a node group.

Creating Node Groups

You can create node groups in the **Data Flow Design** tab or the **Node Group Management** tab of the Administrator tool.

1. In the Administrator tool, on the **Domain** tab, click **Services and Nodes**.
2. In the Domain Navigator, select the Vibe Data Stream Service.
3. In the upper-right corner of the **Vibe Data Stream Service** panel, select the **Vibe Data Stream** tab.
4. Create node groups in the **Data Flow Design** tab or the **Node Group Management** tab.

To create node groups in the **Node Group Management** tab, perform the following steps:

a. In the **Node Groups** panel, right-click on one of the following node groups and specify the properties:
   - Dynamic Node Group. Specify the group name and the regular expression pattern for the node names.
   - Normal Node Group. Specify the group name.

b. Click **OK**.

To create node groups in the **Data Flow Design** tab, perform the following steps:

a. Select the **Data Flow Design** tab.

b. Select a data flow.

c. Select an entity in the data flow.

d. Click **Add Node Groups** in the **Node Groups** panel and click one of the following node groups:
   - Dynamic Node Group. Specify the group name and the regular expression pattern for the node names.
   - Normal Node Group. Specify the group name.

Adding Nodes to Node Groups

You can add VDS Nodes to node groups in the **Node Group Management** tab of the Administrator tool. You must first add a VDS Node and then associate it with a node group.

1. Select the **Node Group Management** tab.
2. Click **Add Node** in the **Node List** panel.
3. Specify the name of the VDS Node and click **Save Node**.
4. In the **Node Groups** panel, select **All Nodes**.

   The list of nodes appears in the **Node List** panel.
5. Select the VDS Node that you want to associate with a node group.
6. Click **Associate to Node Group**.

   The list of normal node groups appears.
7. Select one or more node group that you want to associate the node to and click **Save**.

   Note: To delete a VDS Node from a node group, select it and click **Delete Nodes**.

Export and Import Node Groups

To export and import node groups, use the Informatica command line program.

Before you run the command line program, make sure that the Administrator Daemon and the Administrator tool are running, and that you have created a Vibe Data Stream Service.

Related Topics:

- "Running Commands" on page 180
- "Step 3: Back Up The Node Groups " on page 148

Exporting Node Groups

Use the infacmd command line program to export node groups as a JSON file.

On UNIX, run the following command:

./infacmd.sh vds exportNodeGroup

On Windows, run the following command:

infacmd.bat vds exportNodeGroup

The command uses the following syntax:

```
<-DomainName|-dn> Domain name
  [<-UserName|-un> User Name]
  [<-Password|-pd> Password]
  <-FilePath|-fp> File path where you want to export the node groups to.
  [<-Nodes|-nd> Export nodes associated with node group. Default is false.]
```

Related Topics:

- "exportNodeGroup" on page 184

Importing Node Groups

Use the infacmd command line program to import node groups. When you import node groups, you have to import a file that is the JSON format. If you try to import a file that you have changed, a checksum error occurs and you cannot import the node groups.

On UNIX, run the following command:

./infacmd.sh vds importNodeGroup

On Windows, run the following command:

infacmd.bat vds importNodeGroup

The command uses the following syntax:

```
<-DomainName|-dn> Domain name
  <-UserName|-un> User name]
  [<-Password|-pd> Password]
  <-FilePath|-fp> File path where you want to import node groups from.
  [<-Overwrite|-ow> Overwrite existing node groups. Default is false.]
```

Related Topics:

- "importNodeGroup" on page 185

Importing Multiple Vibe Data Stream Nodes

You can map an entity to multiple nodes. Instead of mapping multiple nodes one at a time, you can import a comma-separated values (CSV) file that contains the node names. The CSV file should not exceed 2 MB. When you import the file, the node names that you import are appended to the list of nodes already mapped to the entity.

1. In the Administrator tool, on the **Domain** tab, click **Services and Nodes**.
2. In the **Domain Navigator** panel, select the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. Select the **Node Group Management** tab.
5. In the **Node Groups** panel, select the node group to which you want to import nodes.

6. Click **Import/Export** > **Import Nodes**.

   The **Open File** dialog box appears.
7. Browse to the location of the CSV file and select the file.
8. Click **OK**.
9. Click **Save**.

Exporting Multiple Vibe Data Stream Nodes

You can export multiple VDS Nodes as a CSV file.

1. In the Administrator tool, on the **Domain** tab, click **Services and Nodes**.
2. In the **Domain Navigator** panel, select the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. Select the **Node Group Management** tab.
5. In the **Node Groups** panel, select the node group that you want to export.
6. Click **Import/Export** > **Export Nodes**.

   A list of VDS Nodes associated with the service is downloaded as a CSV file.

CHAPTER 7

Data Connections

This chapter includes the following topics:

Data Connections Overview

VDS uses a data connection to send data from the source service to the target service.

When you design a data flow, you can choose one of the following data connections:

Ultra Messaging

You can choose the Ultra Messaging data connection when you want to send data to one or more target services in a LAN.

WebSocket(S)

You can choose the WebSocket data connection in a data flow in the following scenarios:

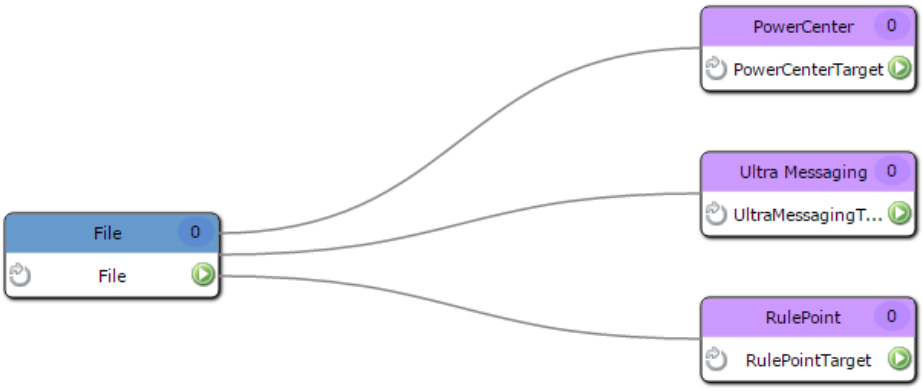
- To transport data over a firewall.
- To aggregate data before writing to the data target.
- For secure data transfer.

Ultra Messaging Data Connection

To transport data, the UM data connection uses Ultra Messaging. VDS generates a topic name for every data connection in the data flow. Source services that use the data connection read data in blocks and publish messages on the topic. Target services subscribe to that topic, receive the data in the form of events, and send the data to the data target.

You can choose the Ultra Messaging data connection when you want to send data to one or more target services in a LAN environment, that is, use a publish/subscribe model.

The following image shows a sample data flow that uses an Ultra Messaging data connection:



When you configure the Ultra Messaging data connection, you can configure source services to use the streaming, load balancing, or persistence messaging mode to publish data.

You can also configure security for the data connection. The Ultra Messaging data connection uses TSL/SSL protocols to secure events. The TSL/SSL protocols use cipher suites or symmetric and asymmetric encryption algorithms to establish a secure communication. When you configure security for the Ultra Messaging data connection, specify the certificate details and optionally specify the cipher suites. The data connection uses the same set of certificates for both the source services and the target services.

Transport Topic Name

When you choose the Ultra Messaging data connection, you can specify the topic name for the transport on which the source service publishes data. This topic is known as well-known topic. If you do not specify a well-known topic name, the data connection uses an autogenerated topic name.

If you create custom source services or transformations, you can partition or shard events to separate the processing of large event streams. When you configure the Ultra Messaging data connection, you can specify comma-separated transport topic names.

You can specify comma-separated transport topic names when you configure data connections between the following entities:

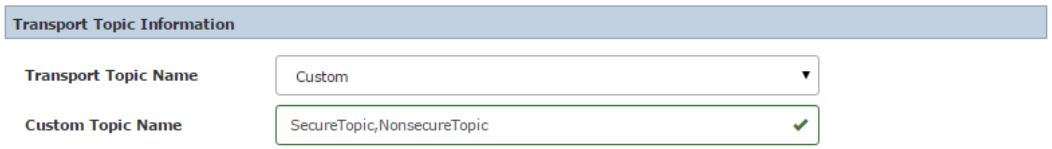
- Source services and target services
- Transformations running on a source service or aggregator and targets

Note: You can partition or shard events only when you publish data to Ultra Messaging, PowerCenter and RulePoint target services.

For example, if you have a source service that publishes secure and nonsecure events, you can publish secure events on one topic and nonsecure events on another topic. To achieve this, select **Custom** in the Transport Topic Name property. Specify the two topic names separated by a comma in the **Custom Topic Name** property as follows:

```
SecureTopic,NonsecureTopic
```

The following image shows a sample **Custom Topic Name** property:



For more information about custom source services, see the *Vibe Data Stream for Machine Data Developer Guide*.

Streaming Mode

You can configure the VDS data flows to use the streaming messaging mode, called Ultra Messaging Streaming, to send data to targets across a network.

When you configure the streaming mode, VDS delivers all the messages published by the source services to all the target services. When you use the streaming mode, associate only one VDS Node with each target service.

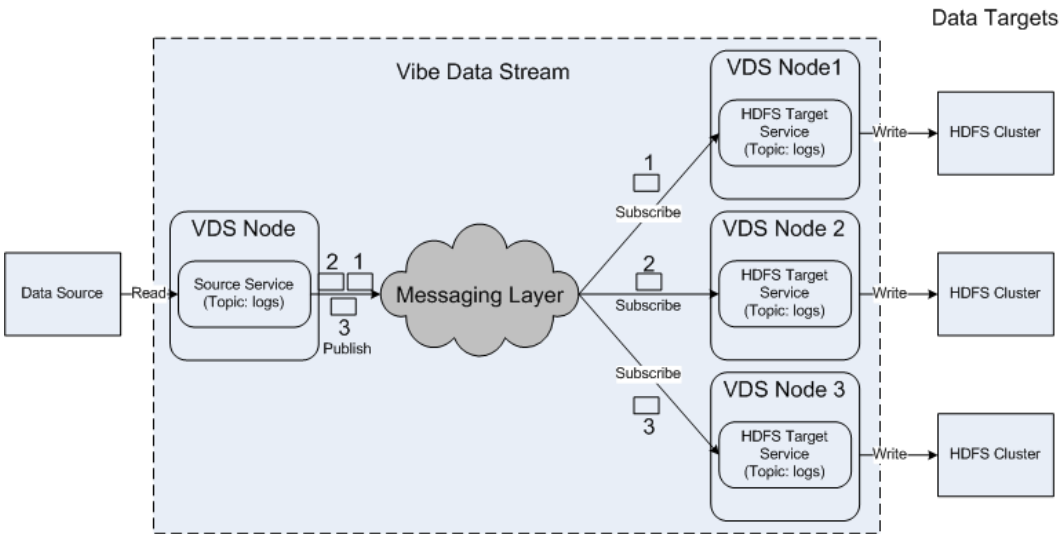
You can configure the messaging mode when you create or edit a connection. Before you change the messaging mode, you must undeploy a data flow. You can also specify advanced Ultra Messaging configuration for the data connection in the data flow.

Note: If you associate more than one VDS Node with a single target service, all the VDS Nodes get duplicate data.

Load Balancing

When you deploy a target service on multiple VDS Nodes, VDS uses the round-robin method to distribute the messages across the instances. VDS avoids data duplication within the load balanced group of instances by delivering each message to one instance. Deploying a target service on multiple nodes ensures that a large volume of data from multiple sources does not overwhelm a single target service instance.

The following image shows how VDS balances the load across HDFS target services that you deployed on three nodes:



VDS balances the load as follows:

1. A source service reads data from the data source and publishes it as three messages over a topic called logs.
2. Three instances of a target service receive the messages.
3. The target service is deployed on three nodes for purposes of load balancing. VDS balances the load across the three instances of the target service in round-robin fashion.

Note: Although VDS uses the round-robin method, some instances might receive more messages than other instances.

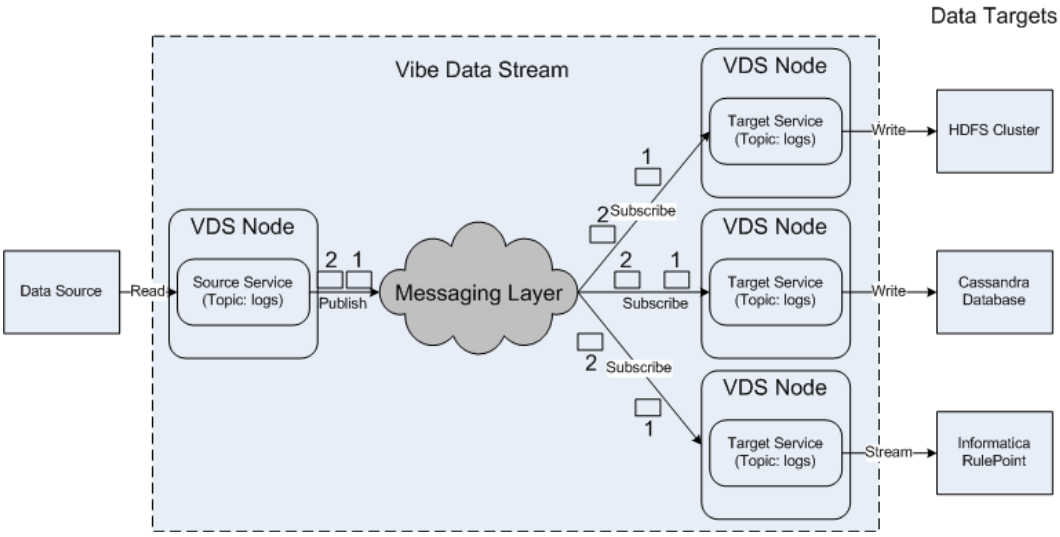
If one or more instances become inactive, VDS removes the inactive instances from load balancing decisions and redistributes the load across the remaining instances.

Data Duplication

In a deployment in which each target system receives data from a standalone instance of a target service, load balancing is not a requirement. VDS duplicates the data across the data targets. VDS delivers all the messages published by the source service to all the data target services so that they have the complete data set for analysis.

For example, you create separate target services to send data to an HDFS cluster, a Cassandra database, and a RulePoint instance.

The following image shows VDS data duplication across the data targets:



VDS duplicates data as follows:

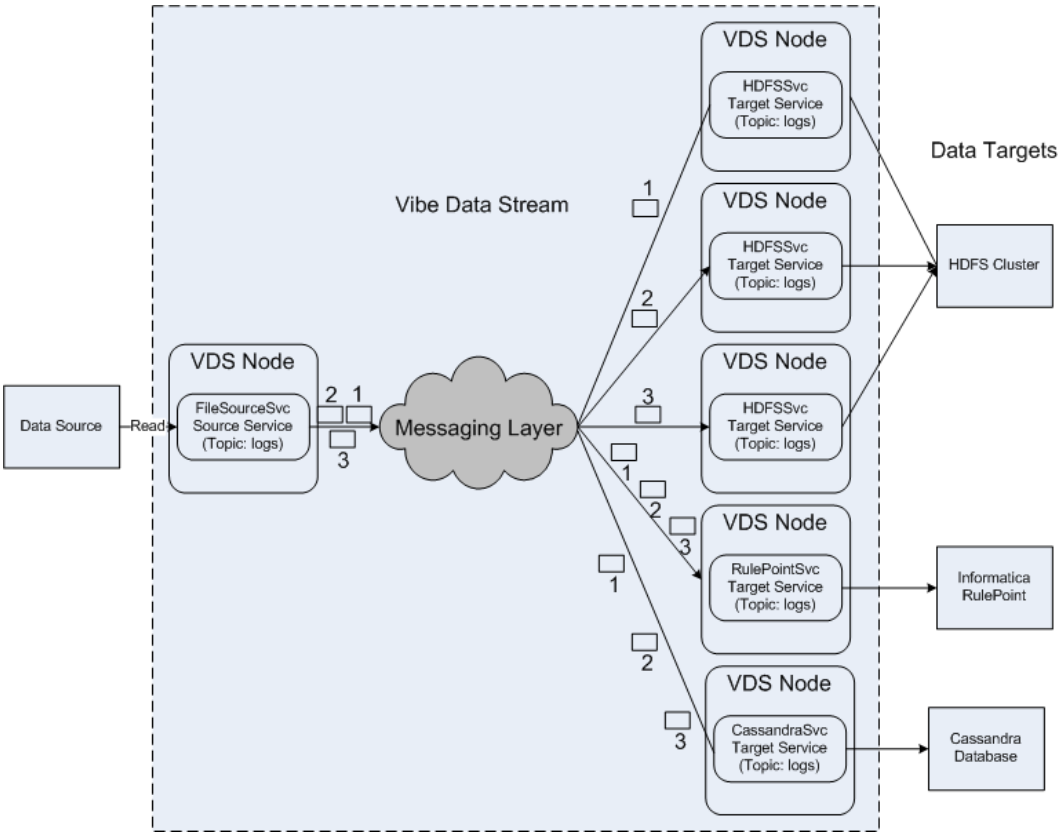
1. A source service reads data from the data source and publishes it as two messages over a topic called logs.
2. VDS delivers both the messages to all the target services.
3. The target services receive both messages and send them to the targets.

Complex Data Flow with Data Duplication and Load Balancing

A data flow can contain target services running on a single VDS Node or on multiple VDS Nodes. In such a deployment, VDS sends all the messages that the source publishes to each standalone target service. For a

target service that you deploy on multiple nodes, VDS uses the round-robin method to distribute a subset of the messages to each target service instance.

The following image shows how VDS performs load balancing across multiple instances of a target service while duplicating data flows across different targets:



The following process describes how VDS balances the load:

1. A file source service FileSourceSvc publish messages on a topic called logs.
2. A RulePoint target service named RulePointSvc, a standalone Cassandra target service named CassandraSvc, and three instances of an HDFS target service named HDFSSvc receive those messages.
3. You deploy HDFSSvc on three VDS Nodes for purposes of load balancing.
4. VDS distributes messages so that CassandraSvc and RulePointSvc receive all the messages that the source service publishes.
5. Simultaneously, VDS performs load balancing across the three instances of HDFSClient in round-robin fashion. VDS delivers a message to one instance of HDFSClient.

Persistence

You can configure the VDS data flows to use the persistence messaging mode, called Ultra Messaging Persistence, to send data to targets across a network.

When you configure the persistence mode, VDS delivers all the messages published by the source services to the target services and to a default persistent store that provides storage to message streams.

You can configure the persistence messaging mode when you create or edit a data flow. If you want to use an external persistence store, you can configure the properties of the store in the **UM XML Configurations**

property. You can also specify advanced Ultra Messaging configuration for the data connection in the data flow.

For more information about Ultra Messaging Persistence, see the *Ultra Messaging Guide for Persistence*.

Ultra Messaging Data Connection Properties

To add an Ultra Messaging data connection, use the Ultra Messaging data connection type when you connect a source service to a target service in a data flow.

You can configure the following properties for the Ultra Messaging data connection type:

Entity Name

Name of the data connection. Maximum length is 32 characters.

Description

Description of the data connection. Maximum length is 256 characters.

Topic resolution type

Type of topic resolution that you want to use.

You can choose one of the following topic resolution types:

- Multicast. Data connection uses multicast topic resolution.
- Unicast. Data connection uses the unicast resolver daemon (LBMRD) for topic resolution.

Resolver daemon

Address of the topic resolution daemon if you choose unicast topic resolution type.

Resolver address

Multicast address that you want to use.

Messaging mode

Mode in which the source service distributes data to the target service.

You can choose one of the following messaging modes:

- Load Balancing. The source service uses load balancing to send data.
- Streaming. The source service uses streaming to distribute data.
- Persistence. The source service uses persistence to distribute data. Select this messaging mode to use the default persistence store. If you want to use an external persistence store, specify the configuration in the **UM XML Configurations** property.

Connection Type

Type of connection.

You can select one of the following connection types:

- UM. The data connection uses a nonsecure connection to publish events.
- UM Secure. The data connection uses a secure connection to publish events.

Cipher suites

The cipher suite that the Ultra Messaging data connection uses to secure data.

Certificate

The certificate file name that the Ultra Messaging data connection uses to secure data. Specify the certificate if you select the **UM Secure** connection type.

Certificate key

The certificate key file name that the data connection uses. Ultra Messaging supports certificates that are in PEM or ASN1 format.

Specify the certificate key file name if you select the **UM Secure** connection type.

Note: This file is located in the `<VDS installation directory>/node/certificates` directory.

Certificate key password

Password to the key file. Specify the certificate if you select the **UM Secure** connection type. Select a secure parameter if you have already created it. To specify a key and value, click **Click to add secure parameters**.

Note: This file is located in the `<VDS installation directory>/node/certificates` directory.

Trusted certificates

The trusted certificates file that the Ultra Messaging data connection uses. Specify the file name if you select the **UM Secure** connection type.

Note: This file is located in the `<VDS installation directory>/node/certificates` directory.

Resolver port

Optional. The unicast or multicast UDP port used by the UM transport for topic resolution.

Specify the port if you want to open fixed ports in a firewall.

Note: If you specify the port, the default port ranges used by VDS are overwritten. You must map each entity in the data flow to a unique VDS node instance.

TCP request port

Optional. The port on which to listen for responses from requests. Each UM context binds to a TCP port for requests when it is initialized.

Specify the port if you want to open fixed ports in a firewall.

TCP Transport Port

Optional. The preferred TCP port for the topic that the UM application publishes on. All receivers interested in the topic establish a connection to this port.

Specify the port if you want to open fixed ports in a firewall.

UM XML Configuration

The UM configuration that the data connection uses. Specify the configuration in the following format:

```
<template name="entity-user-template">
<options type="source">
<option name="umq_ulb_flight_size" default-value="1000"/>
</options>
<options type="context">
</options>
<options type="receiver">
</options>
</template>
```

Maximum length is 4000 characters.

To configure intragroup or intergroup stability, you can use the following sample configuration:
Intragroup stability

```
<template name="entity-user-template">
<options type="source">
```

```
    <option default-value="all-stores"
    name="ume_retention_intragroup_stability_behavior"/>
    </options>
    <options type="context"></options>
    <options type="receiver"></options>
    </template>
```

Intergroup stability

```
    <template name="entity-user-template">
    <options type="source">
    <option default-value="all"  name="ume_retention_intergroup_stability_behavior"/>
    <option default-value="0:3"  name="ume_store_group"/>
    <option default-value="1:3"  name="ume_store_group"/>
    </options>
    <options type="context"></options>
    <options type="receiver"></options>
    </template>
```

You can use the following sample configuration to use an external persistence store if you select the Persistence messaging mode:

```
<template name="entity-user-template">
                    <options type="source">
                        <option name="ume_store_name" default-value="VDS230Store1"/>
                           <option name="ume_store_name" default-
value="VDS230Store2"/>
                           <option name="ume_store_name" default-
value="VDS230Store3"/>
                    </options>
                    <options type="context">
                    </options>
                    <options type="receiver">
                    </options>
            </template>
```

You can use the following sample configuration for the external persistence store if you select the Persistence messaging mode:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  - A simple configuration file for a UME store daemon that listens on port
  - 14567 for registration. Logs are appended to the file ume-example-stored.log
-->
<!DOCTYPE umestore SYSTEM "umestore.dtd">
<ume-store version="1.2">
    <daemon>
        <lbm-config>C:\Informatica\UM\UMP_6.8\config\ump.cfg</lbm-config>
        <log>ump.out</log>
        <pidfile>umestored.pid</pidfile>
        <web-monitor>*:15304</web-monitor>
        <lbm-license-file><Path to license>\UMQLicense.txt</lbm-license-file>
    </daemon>
    <stores>
        <store name="VDS Store A" port="14567">
            <ume-attributes>
                <option type="store" name="disk-cache-directory" value="C:
\Informatica\UM\UMP_6.8\cache1"/>
                <option type="store" name="disk-state-directory" value="C:
\Informatica\UM\UMP_6.8\state1"/>
                <option type="store" name="allow-proxy-source" value="1"/>
                <option type="store" name="context-name" value="VDSStoreA"/>
            </ume-attributes>
            <topics>
                <topic pattern="." type="PCRE">
                    <ume-attributes>
                  <option type="store" name="repository-type" value="disk"/>
                        <option type="store" name="repository-size-threshold"
value="104857600"/>
                        <option type="store" name="repository-size-limit"
value="209715200"/>
```

```
                        <option type="store" name="repository-disk-file-size-limit"
value="1073741824"/>
                        <option type="store" name="source-activity-timeout"
value="30000"/>
                        <option type="store" name="receiver-activity-timeout"
value="604800000"/>
                        <option type="store" name="retransmission-request-
forwarding" value="0"/>
                    </ume-attributes>
                </topic>
            </topics>
        </store>
        <store name="VDS Store B" port="14568">
            <ume-attributes>
                <option type="store" name="disk-cache-directory" value="C:
\Informatica\UM\UMP_6.8\cache2"/>
                <option type="store" name="disk-state-directory" value="C:
\Informatica\UM\UMP_6.8\state2"/>
                <option type="store" name="allow-proxy-source" value="1"/>
                <option type="store" name="context-name" value="VDSStoreB"/>
            </ume-attributes>
            <topics>
                <topic pattern="." type="PCRE">
                    <ume-attributes>
                  <option type="store" name="repository-type" value="disk"/>
                        <option type="store" name="repository-size-threshold"
value="104857600"/>
                        <option type="store" name="repository-size-limit"
value="209715200"/>
                        <option type="store" name="repository-disk-file-size-limit"
value="1073741824"/>
                        <option type="store" name="source-activity-timeout"
value="30000"/>
                        <option type="store" name="receiver-activity-timeout"
value="604800000"/>
                        <option type="store" name="retransmission-request-
forwarding" value="0"/>
                    </ume-attributes>
                </topic>
            </topics>
        </store>
        <store name="VDS Store C" port="14569">
            <ume-attributes>
                <option type="store" name="disk-cache-directory" value="C:
\Informatica\UM\UMP_6.8\cache3"/>
                <option type="store" name="disk-state-directory" value="C:
\Informatica\UM\UMP_6.8\state3"/>
                <option type="store" name="allow-proxy-source" value="1"/>
                <option type="store" name="context-name" value="VDSStoreC"/>
            </ume-attributes>
            <topics>
                <topic pattern="." type="PCRE">
                    <ume-attributes>
                  <option type="store" name="repository-type" value="disk"/>
                        <option type="store" name="repository-size-threshold"
value="104857600"/>
                        <option type="store" name="repository-size-limit"
value="209715200"/>
                        <option type="store" name="repository-disk-file-size-limit"
value="1073741824"/>
                        <option type="store" name="source-activity-timeout"
value="30000"/>
                        <option type="store" name="receiver-activity-timeout"
value="604800000"/>
                        <option type="store" name="retransmission-request-
forwarding" value="0"/>
                    </ume-attributes>
                </topic>
            </topics>
        </store>
```

```
    </stores>
</ume-store>
```

Configuring a Secure Persistence Store

You can configure security for the persistence store when you choose the Persistence messaging mode and UM Secure connection type.

1. Stop the persistence store (UM Store).
2. If you use the default persistence store, navigate to the `<VDS installation directory/admind/config` directory. Include the following properties in the `umestored.cfg` file:

```
context tls_certificate <path of certificate>
context tls_certificate_key <path of certificate key>
context tls_certificate_key_password <password key of the certificate>
context tls_trusted_certificates <path to trusted certificate>
context use_tls 1
context tls_cipher_suites
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE
_ECDSA_WITH_AES_256_GCM_SHA384
```

   If you use an external persistence store, add the properties in the configuration file of the persistence store.
3. Start the persistence store.
4. Optionally, if you use an external persistence store, add the following configuration for the UM XML Configuration property of the Ultra Messaging data connection:

```
<template name="entity-user-template">
<options type="source">
<option name="ume_store_name" default-value="VDS230Store1"/>
<option name="ume_store_name" default-value="VDS230Store2"/>
<option name="ume_store_name" default-value="VDS230Store3"/>
</options>
<options type="context">
</options>
<options type="receiver">
</options>
</template>
```

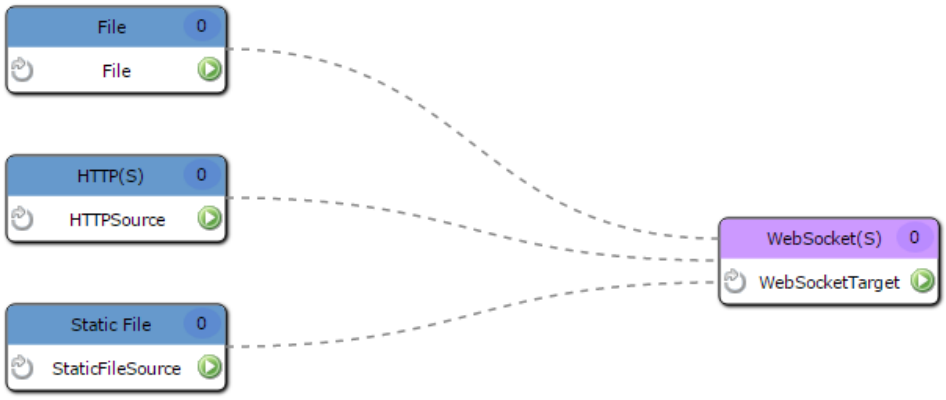
WebSocket Data Connection

To transport data, the WebSocket data connection uses the WebSocket protocol, which provides full-duplex communication over a single TCP connection. A full-duplex communication system allows simultaneous communication in both directions.

You can choose the WebSocket data connection to configure a data flow in the following scenarios:

- To transport data over a WAN.
- To transport data over a firewall.
- To transport data from multiple source services to a single target service.
- To transfer data securely. You can use the WebSocket secure mode for secure data transfer.
- To aggregate data before writing to the data target.

The following image shows a sample data flow that uses a WebSocket data connection:



When you configure the WebSocket data connection, you can configure the following modes to transport data:

Streaming mode

The Streaming mode does not guarantee the delivery of streamed data.

Acknowledgment mode

In the Acknowledgment mode the target service sends an acknowledgment to source service after it writes data to the target.

You can also configure security for the data connection. When you configure security, specify the keystore and truststore file details.

When you use a WebSocket data connection in a data flow with one source service sending data to a target service associated with multiple VDS Nodes, the source service sends data to only one of the target service instances.

You can configure the WebSocket data connection to connect to an external load balancer if you are sending data from multiple source services to multiple target service instances. If you use a load balancer in a data flow that has one source service and multiple target service instances, the load balancer does not balance the data across the target service instances.

Configuring an External Load Balancer

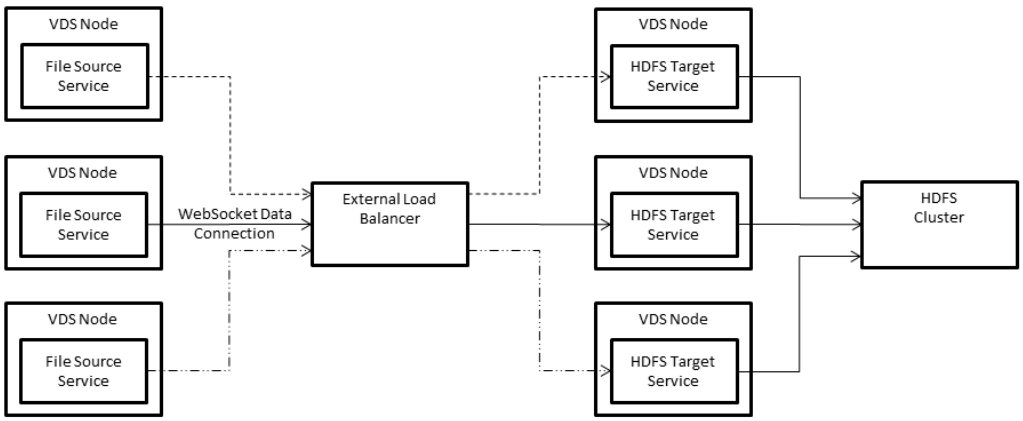
Configure the WebSocket data connection to connect to an external load balancer when you have multiple source services sending data to multiple target services. An external load balancer is a hardware or software load balancer that balances load across target services in an external network.

To connect to an external load balancer, use the **Load Balancer IP** property when you configure the data connection.

Example

You create a data flow that uses a WebSocket data connection to write data to an HDFS cluster. In the data flow, you create three File source services to send data to three HDFS target services mapped to multiple VDS Nodes. If the HDFS cluster uses a load balancer to distribute the load across the VDS Nodes, you must configure the IP address of the load balancer when you configure the WebSocket data connection. Ensure that the load balancer configuration contains the IP addresses of all the VDS Nodes to which the HDFS target services are mapped, so that it can perform load balancing to distribute the load.

The following image shows a data flow in which a File source service sends data to an external load balancer:



WebSocket Data Connection Properties

To add a WebSocket data connection, use the WebSocket data connection type when you connect a source service to a target service in a data flow.

The following table describes the properties that you need to configure for the WebSocket data connection type:

Entity Name

Name of the data connection. Maximum length is 32 characters.

Description

Description of the data connection. Maximum length is 256 characters.

Messaging mode

Mode in which the source service distributes data to the target service.

You can choose one of the following messaging modes:

- Acknowledgment. The target service sends an acknowledgment to source service after it writes data to the target.
- Streaming. The data connection uses streaming to distribute data.

Default is Acknowledgment mode.

Connection Type

The type of connection that the data connection uses.

You can select one of the following connection types:

- WebSocket
- WebSocket Secure

KeyStore Path

The directory that contains the keystore file. Specify the absolute path to the file.

Specify this path if you select the **WebSocket Secure** connection type.

KeyStore Password

Password for the keystore file.

You must specify this property if you select the **WebSocket Secure** connection type.

Select a secure parameter if you have already created it. To specify a key and value, click **Click to add secure parameters**.

TrustStore Path

Path and file name of the truststore file.

Specify this path if you select the **WebSocket Secure** connection type.

TrustStore Password

Password for the truststore file.

You must specify this property if you select the **WebSocket Secure** connection type.

Select a secure parameter if you have already created it. To specify a key and value, click **Click to add secure parameters**.

Port

Port on which the target service listens for incoming connections.

Status Check Path

Path to which an HTTP GET request is sent to verify if the server is running.

Load Balancer IP

IP address of the external load balancer.

WS Advanced Configuration

The advanced configuration that the data connection uses. You can configure the following options:

- `event_flush_frequency`. The amount of time in milliseconds after which events are flushed from the source service to the target service. Default is 100.
- `maximum_events_batched`. The maximum number of events that are sent in a batch from the source service to the target service. Default is 1000.

  Note: The data is sent to the source service when either the event flush frequency or the maximum events batched event occurs.
- `maximum_client_buffer`. The buffer size in bytes allocated to the source service. if this buffer is full, source service will not accept more data from the source, to send to the target service. Default is 33554432.
- `acknowledgment_flush_frequency`. The frequency in milliseconds at which the target service sends acknowledgments to the source service after receiving data. Default is 500.
- `maximum_send_attempts`. The maximum number of times, including the first time, that the source service tries to send data to the target service. Default is 5.
- `acknowledgment_idle_time`. The amount of time in milliseconds that the source service waits for an acknowledgment from the target service before it tries to resend data. Default is 10000.
- `connection_idle_time`. The amount of time in milliseconds that the source service waits before it tries to reconnect to the target service. Default is 10000.
- `maximum_connection_attempts`. The maximum number of times the source service tries to connect to the target service. Default is 3.

Specify positive numeric values for the options.

Note: You cannot configure `acknowledgment_idle_time`, `maximum_send_attempts`, and `acknowledgment_flush_frequency` if you use the Streaming messaging mode.

CHAPTER 8

Working With Data Flows

This chapter includes the following topics:

Working With Data Flows Overview

Use the Administrator tool to design the flow of data from the data source to the data target and to deploy the data flow. In a data flow, you can manage entities, data connections, and mappings between entities and nodes. You can deploy and undeploy data flows.

Before you create a data flow, perform the following tasks:

1. Perform the postinstallation tasks.
2. Determine the transport layer that you want to use in the data connection of the data flow.
3. Get information about the data sources and data targets.
4. Get information about the file rollover scheme that the application administrator has implemented for the data source. Determine a rollover scheme for the files that the target services create.
5. Determine what transformations you want to apply to the data.

To create and deploy data flows, perform the following tasks:

1. Design a data flow that reflects how you want VDS to move the data from the source to the target.
2. Add source services and target services to the data flow.
3. Connect source services to target services and add any data transformations that you want.
4. Map the source services and target services to the nodes on which you want them to run. The source services use a data connection to send data to the target services.
5. Deploy the data flow. You can deploy a data flow and also deploy the individual entities in the data flow. At deployment time, VDS starts the source services and target services. The source services collect data, the transformations transform the data, and the target services write the data to the targets. The process of deployment of a data flow does not interrupt other data flows.

To change the design after you deploy a data flow, or to edit the properties of a running entity, undeploy the data flow. Make the design changes that you want and then redeploy the dataflow. In the following scenarios, you might lose data that the source generates during the period that the data flow is undeployed:

- The source application transmits the data as a stream.
- The source application performs an active file rollover when the data flow is in the undeployed state.

Types of Data Flows

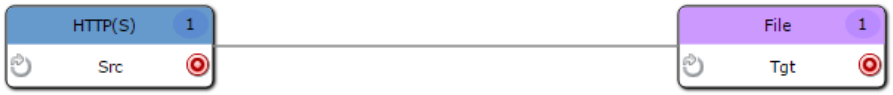
Based on which target services subscribe to a particular source, you get a one-to-one, a one-to-many, or a many-to-many data flow.

The following data flows show how you can connect sources and targets:

One-to-One Data Flow

Connects one source service to one target service.

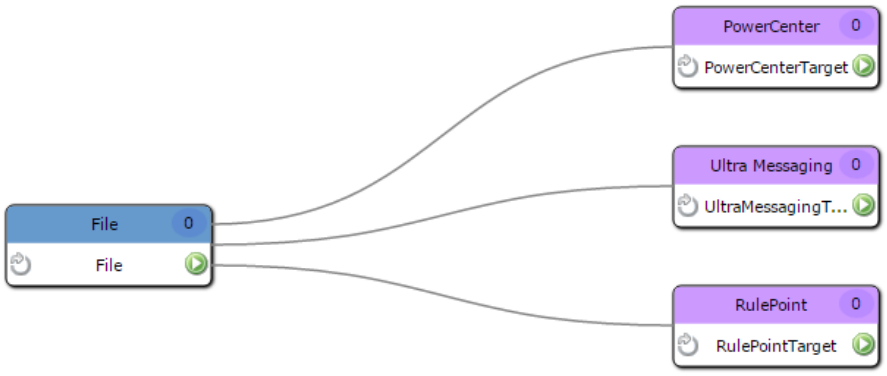
The following image shows a one-to-one data flow:



One-to-Many Data Flow

Connects one source service to multiple target services. All the target services subscribe to the data that the source service publishes.

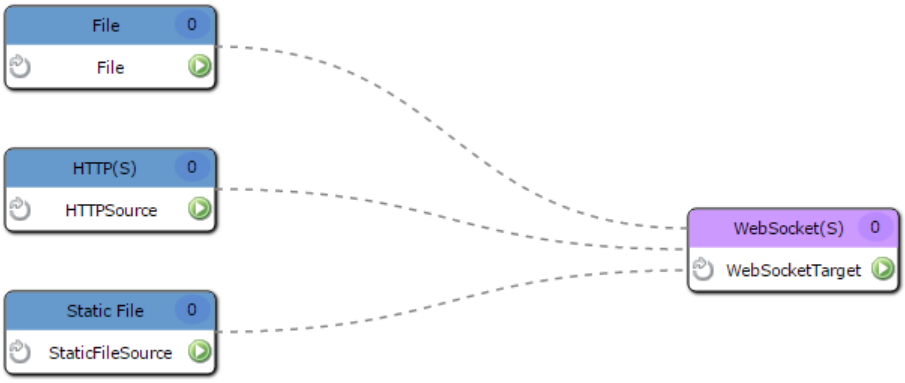
The following image shows a one-to-many data flow:



Many-to-One Data Flow

Connects multiple source services to a single target service. In a many-to-one data flow, the source services might overload the target service. You can reduce the load on the target service by mapping the target service to multiple nodes.

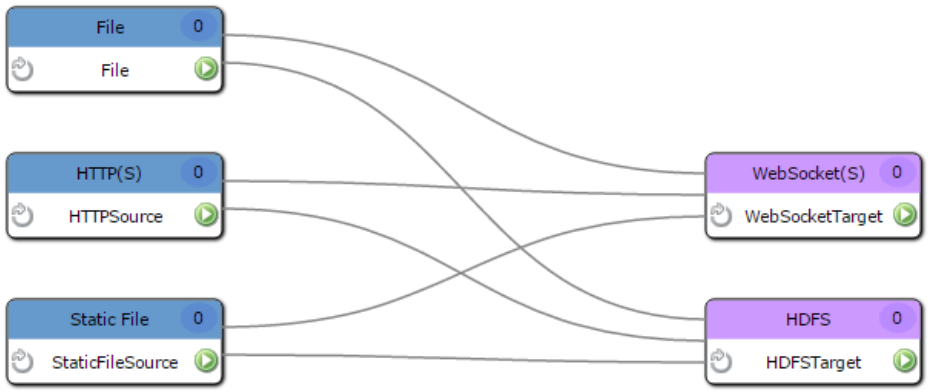
The following image shows a many-to-one data flow:



Many-to-Many Data Flow

Connects multiple source services to multiple target services.

The following image shows a many-to-many data flow:



Creating a Data Flow

Create a data flow and specify the name of the data flow.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream service.
3. Select the **Vibe Data Stream** tab.
4. In the **Data Flows** pane, right-click **All Data Flows**.
5. Click **New data flow**.

   The **Create Data Flow** dialog box appears.
6. In the **Name** field, enter a name for the data flow.
7. Click **OK**.

Related Topics:

- " Vibe Data Stream Data Flow Process" on page 18

Data Flow Design Tab

You can create, deploy, and undeploy data flows in the **Data Flow Design** tab.

The contents that appear on the **Data Flow Design** tab and the tasks you can perform vary based on the view that you select.

The following panels appear in the tab:

Data Flows

Displays a list of the data flows that you have configured on the Vibe Data Stream service. You can manage data flows in this panel.

Entity Types

Displays the entity types, that is, the source service, target service, and transformation types that you can use in data flows. The pane organizes the entity types in expandable and collapsible trees of source service types, target service types, and transformation types. This panel also displays any custom entity type that you add to VDS.

Summary View: All Data Flows

Displays the source service types and target service types used in VDS. The panel shows a summary of associations between the source service types and the target service types to which they send data.

Data Flow Designer

Appears when you select a data flow in the **Data Flows** pane. This panel displays the relationships between the entities in the data flow. This panel also displays buttons to deploy or undeploy the data flow and to delete an entity from the design.

Entity Details

Appears when you select an entity in the **Data Flow Designer** panel, and consists of a **Properties** tab and a **Nodes** tab. The **Properties** tab displays the properties of the selected entity. You can edit the properties on this tab. The **Nodes** tab lists the nodes that you have mapped to the selected service. You can map nodes to or dissociate nodes from the selected service.

Adding Entities to a Data Flow

Add source services and target services to the data flow, and then connect the source and target services. Add the transformations that you need to the connections. You can add both built-in entities and custom entities to the data flow.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator**, click the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flows** pane, click the data flow to which you want to add a source.
5. From the **Entity Types** pane, drag the source service to the **Data Flow Designer** pane.

   The **New Source** dialog box appears.
6. Specify the required properties and any of the optional properties..
7. From the **Entity Types** pane, drag the target service to the **Data Flow Designer** pane.

   The **New Target** dialog box appears.
8. Specify the required properties and any of the optional properties..
9. To connect a source to a target, in the **Data Flow Designer** pane, drag the pointer from the source to the target.

   The **Connection List** dialog box appears.
10. Select one of the following data connections from the list.

New Data Connection. You can add an Ultra Messaging data connection or a WebSocket data connection:

a. Click **New**.

b. In the **Add Connection** dialog box, specify the values for the properties of the data connection.

c. Click **OK**.

Existing Connections. If you have created a data connection, you can select it from the list of existing connections. To add the existing data connection to the data flow, click **Use**.

Related Topics:

- "Built-in Source Service Types" on page 43
- "Built-in Target Service Types" on page 67
- "Built-in Transformation Types" on page 82
- "Guidelines for Adding Transformations" on page 83

Vibe Data Stream Node Mapping

You can associate and dissociate VDS Nodes from a source service or target service. If you want a service to run on a particular node, map the service to the node. When you no longer want a service to run on a particular node, dissociate the service from the node. After you map or dissociate nodes from a service, save your changes.

Mapping Services to Vibe Data Stream Node Groups

To deploy a source service or target service on a particular VDS Node, map the service to a node group.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, select the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flow Designer** pane, select the source service or target service to which you want to map nodes.
5. In the **Entity Details** pane that appears, click the **Node Groups** tab.
6. Click **Create New Relationship**.
7. Select the node group from the list of node groups that appear.
8. Click **Apply**.
9. Optionally, to create a node group and associate it with the source service or target service, click **Create Node Group**.
10. Click one of the following node groups:
    - Dynamic Node Group. Specify the group name and the regular expression pattern for the node names.
    - Normal Node Group. Specify the group name.
11. Click **OK**.

Dissociating a Service from a Vibe Data Stream Node

Dissociate a source service or target service from a VDS Node if you no longer need the service to run on the node.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flow Designer** pane, select the service to which you want to map nodes.
5. In the **Entity Details** pane that appears, click the **Node Groups** tab.
6. To dissociate a service from a node group, select a node group from the list of associated node groups and click **Delete Relationship**.

Export and Import Data Flows

You can export data flows that you create so that other users can import them and use them. You can also import data flows after changes in configuration or when you want to recover data flows after a corruption.

To export and import data flows, use the Informatica command line program. Before you run the command line program, make sure that the Administrator Daemon and the Administrator tool are running, and that you have created a Vibe Data Stream Service.

Related Topics:

- "Running Commands" on page 180
- "Step 2: Back Up Data Flows" on page 148

Exporting Data Flows

Use the infacmd command line program to export data flows as a JSON file.

When you export data flows, the state of the data flow is not exported. For example, if the data flows are in a deployed state, when you run the command to export the data flows, they are exported in the undeployed state. When you export the data flows, the secure fields are exported as hidden text.

On UNIX, run the following command:

./infacmd.sh vds exportDataFlow

On Windows, run the following command:

infacmd.bat vds exportDataFlow

The command uses the following syntax:

```
exportDataFlow
<-DomainName|-dn> domain name
[<-UserName|-un> user name]
[<-Password|-pd> password]
<-FilePath|-fp> File path where you want to export the data flows to.
[<-Dataflows|-df> Comma separated values of dataflows. Ignore if you want to export all
dataflows.]
```

Related Topics:

- " exportDataFlow" on page 182

Importing Data Flows

Use the infacmd command line program to import data flows. When you import data flows, you have to import a file that is the JSON format. If you try to import a file that you have changed, a checksum error occurs and you cannot import the data flows.

If the data flows that you import has the same name as the existing data flows, the command throws an error. Use the command with the overwrite option to import the data flows.

When you import data flows, the data flows are in an undeployed state.

On UNIX, run the following command:

./infacmd.sh vds importDataFlow

On Windows, run the following command:

infacmd.bat vds importDataFlow

The command uses the following syntax:

```
importDataFlow
<-DomainName|-dn> domain name
[<-UserName|-un> user name]
[<-Password|-pd> password]
<-FilePath|-fp> File path where you want to import data flows from.
[<-Overwrite|-ow> Overwrite existing data flows. Default is false.
```

Related Topics:

- "importDataFlow" on page 183

Deploying a Data Flow

Deploy a data flow to start the source services and target services in the data flow. You can also deploy individual source services and target services in the data flow.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, select **Vibe Data Stream Service**.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flows** pane, select the data flow that you want to deploy.
5. In the **Data Flow Designer** pane, click **Deploy**.

   VDS deploys the data flow, and a check mark appears beside the name of the data flow in the **Data Flows** pane. The check mark indicates that you have deployed the data flow.

Undeploying a Data Flow

Undeploy a data flow when you want to change the data flow or the properties of an entity. When you undeploy a data flow, you do not affect other data flows.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream Service.
3. In the **Data Flows** pane, select the data flow that you want to undeploy.
4. In the **Data Flow Designer** pane, click **Undeploy**.

   VDS undeploys the data flow. This action stops the services in the data flow. VDS also removes the check mark beside the name of the data flow in the **Data Flows** pane to indicate that the data flow is undeployed.

Undeploying and Deploying all Data Flows

You can deploy all data flows after changes in configuration. When you have a large number of data flows, it might not be possible to deploy or undeploy each one individually.

The data flows are not automatically refreshed when there is a change in configuration. You must undeploy the data flows and deploy them again.

Optionally, to clear the ZooKeeper configuration from the data flows, perform a force clean when you undeploy all data flows.

Undeploying All Data Flows

Undeploy all data flows when you want to refresh data flows after a change in configuration or want to recover from a data corruption.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the **Vibe Data Stream Service**.
3. In the upper-right corner of the **Vibe Data Stream Service** panel, click **Vibe Data Stream**.
4. In the **Data Flows** panel, select **All Data Flows**.
5. To undeploy all data flows, click **Undeploy All** in the **Summary View: All Data Flows** panel.

   Optionally, to clear the data flow configuration or corrupted data, select **Force clean all data flow configuration from ZooKeeper**.

Deploying All Data Flows

Deploy all data flows to recover the data and the states of the data flow when you want to recover from a data corruption. The Deploy All action deploys all data flows in a sequential order. During this action, if one of the data flows is not deployed, the remaining data flows in the sequence remain undeployed.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the **Vibe Data Stream Service**.
3. In the upper-right corner of the **Vibe Data Stream Service** panel, click **Vibe Data Stream**.
4. In the **Data Flows** panel, select **All Data Flows**.

5. To deploy all data flows, click **Deploy All** in the **Summary View: All Data Flows** panel.

Editing Data Flows and Entities

Edit a data flow to change the name of the data flow. Edit an entity when you want to change the properties of the entity. Undeploy a data flow before you change the properties of the data flow or an entity in the data flow.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flows** pane, right-click the data flow whose name you want to edit.
5. Click **Edit**.
6. In the **Data Flow** dialog box, edit the name.
7. Click **OK**.
8. To edit the source service, double-click the source service in the **Data Flow Designer** pane, or click **Edit** in the **Entity Details** pane.
9. In the dialog box that appears, edit the properties of the source service.
10. Click **OK**.
11. To edit the target service, double-click the target service in the **Data Flow Designer** pane, or click **Edit** in the **Entity Details** pane.
12. In the dialog box that appears, edit the properties of the target service.
13. Click **OK**.
14. To edit the transformation, double-click the transformation in the **Data Flow Designer** pane, or click **Edit** in the **Entity Details** pane.
15. In the **Entity Details** pane, click **Edit**.
16. In the dialog box that appears, edit the properties of the transformation.
17. Click **OK**.

Cloning a Data Flow

You can create an exact copy of a data flow by cloning it. After cloning the data flow, you make changes as required.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flows** pane, right-click the data flow whose name you want to clone.
5. Click **Clone Dataflow**.

Removing Data Flows and Entities

Remove a data flow if you no longer require the data flow. Undeploy the data flow, and then remove it. If you remove a data flow, VDS removes the data flow from the Data Flows pane and all the entities that the data flow contains. If you remove a source service or target service, VDS removes the transformations and connections that you associated with the service. If you remove a transformation, VDS does not remove associated entities.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. To remove a data flow, right-click the data flow that you want to remove in the **Data Flows** pane.
5. Click **Delete**.
6. To remove an connection, select the data flow from which you want to remove the entity in the **Data Flows** pane.
7. Select the entity that you want to remove and click **Delete**.

Verifying Entity Properties

To verify the data flow design, review the properties of the entities in the design.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flows** pane, select the data flow.
5. In the **Data Flow Designer** pane, click the entity.

   The **Entity Details** pane appears.
6. To verify that you have configured the entity correctly, on the **Properties** tab, review the properties of the entity.
7. To make changes, edit the properties of the entity.
8. Click the **Nodes** tab.
9. To verify that you have mapped the services to nodes correctly, review the list of nodes that you have mapped to the entity.
10. Add or remove nodes as required, and then click **Save**.

Configuring Targets with Data Connection and Target Service Properties

VDS generates values for internal properties, such as topic name and receiver type ID, and assigns them to appropriate entities. For some target service types, such as PowerCenter and RulePoint, get values for the topic name, and configure the target service to subscribe to messages on that topic.

For example, VDS generates and assigns a topic name to a connection. If you use RulePoint as a target for VDS, get the topic name and specify the topic name in the RulePoint configuration.

Getting the Topic Name Assigned to a Connection

VDS generates the topic name for a connection and displays the topic name as a connection property.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flows** pane, select the data flow in which you want to determine the topic name.
5. In the **Data Flow Designer** pane, select the connection whose topic name you want to determine.
6. In the **Entity Details** pane, view the topic name.

   The **Topic** field displays the topic name.

Getting the Receiver Type ID of a Target Service

VDS generates the receiver type ID for a target service. VDS displays the ID as a property of the service.

1. In the Administrator tool, on the **Manage** tab, click **Services and Nodes**.
2. In the **Domain Navigator** pane, click the Vibe Data Stream Service.
3. In the upper-right corner of the Vibe Data Stream Service pane, click **Vibe Data Stream**.
4. In the **Data Flows** pane, select the data flow in which you want to determine the receiver type ID.
5. In the **Data Flow Designer** pane, select the target service whose receiver type ID you want to determine.
6. In the **Entity Details** pane, click the **Properties** tab.

   The **Receiver Type ID** field displays the receiver type ID.

Getting Entity Alerts

VDS generates alerts when the following events occur:

- A VDS Node shuts down abnormally. If a VDS Node shuts down because of an exception, VDS generates a `NODE DOWN` alert after 5 seconds.
- The machine on which the VDS Node is running shuts down. VDS generates a `NODE DOWN` alert after 5 seconds.
- ZooKeeper shuts down abnormally. VDS generates a `NODE DOWN` alert.

- A source service, or a target service shut down abnormally. VDS generates an `ENTITY DOWN` alert.

VDS does not generate alerts when you shut down the VDS Node or when a source service or target service stops when you undeploy the data flow.

The Administrator Daemon publishes the alerts on a UM topic named `vdsalerts`. To receive these alerts, you can configure the target to subscribe to the `vdsalerts` topic.

For example, to receive these alerts in RulePoint, you can specify the `vdsalerts` topic name in the RulePoint configuration. You can also use a UM receiving application and subscribe to the `vdsalerts` topic.

CHAPTER 9

Managing the Vibe Data Stream Components

This chapter includes the following topics:

Managing the Vibe Data Stream Components Overview

Vibe Data Stream allows you to manage VDS components, such as the Administrator Daemon, VDS Node, and the domain. You can verify the status of components, manage logs for, start, or stop the components. You can also configure backup for the H2 database of the Administrator Daemon.

Administrator Daemon Management

You can start, stop, or verify the status of the Administrator Daemon. You can also configure backup of the H2 database and tracing for the log files.

Verifying the Administrator Daemon Status

You can verify the status of the Administrator Daemon on Linux or Windows.

On Linux, run the following command from the `<Administrator Daemon installation directory>\admind\bin` folder:

```
./admind.sh status
```

On Windows, view the status of **Informatica Administrator Daemon** from Windows Administrative Tools.

Starting or Stopping the Administrator Daemon on Linux

To start the Administrator Daemon, run the `admin.sh` command. The `admin.sh` command also starts or stops the topic resolution daemon (LBMRD), the persistence store (UM Store), and Apache ZooKeeper along with the Administrator Daemon.

To start the Administrator Daemon, LBMRD, the persistence store (UM Store), and ZooKeeper, run the following command from the `<VDS installation directory>/admind/bin` folder:

```
./admin.sh start
```

To start one of the components, run the following command:

```
./admin.sh start <component>
```

Where, `<component>` is `admind`, `lbmrd`, `umestored`, or `zookeeper`.

To stop the Administrator Daemon, run the following command:

```
./admin.sh stop
```

To stop one of the components, run the following command:

```
./admin.sh stop <component>
```

Where, `<component>` is `admind`, `lbmrd`, `umestored`, or `zookeeper`.

Starting or Stopping the Administrator Daemon on Windows

You can start or stop the Administrator Daemon from Windows Administrative Tools.

1. From Windows **Administrative Tools**, select **Services**.
2. Right-click the **Informatica ADMIND Service 230** service.
3. Choose to start or stop the service.

Managing the Administrator Daemon Logs

The Administrator Daemon writes log messages to the `<Administrator Daemon Installation Directory>/logs/` directory. The Administrator Daemon uses the log4j API. To change how the Administrator Daemon logs messages, edit the `admind_log4j.conf` file.

1. Open the `admind_log4j.conf` file in the following directory:

   `<Administrator Daemon Installation Directory>/config`
2. Set the value of `umsm.root.logger`. You can choose to log messages at one of the following levels:
   - FATAL. Writes FATAL messages to the log.
   - ERROR. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures and service errors.
   - WARNING. Writes FATAL, ERROR, and WARNING messages to the log. WARNING errors include recoverable system failures or warnings.
   - INFO. Writes FATAL, ERROR, WARNING, and INFO messages to the log. INFO messages include system and service change messages.
   - DEBUG. Writes FATAL, ERROR, WARNING, INFO, and DEBUG messages to the log. DEBUG messages are user request logs.

   By default, the Administrator Daemon logs messages at the INFO level.
3. Set the path to the log file.

4. Save and close the file.

Vibe Data Stream Node Management

You can start, stop, or verify the status of the VDS Node. You can also configure tracing for the log files.

Verifying the Vibe Data Stream Node Status

You can verify the status of VDS Node on Linux or Windows.

On Linux, run the following command from the `<VDS Node installation directory>\node\bin` directory:

```
./node.sh status <node name>
```

To verify the status of the VDS Node installed on a remote machine, run the following command:

```
./remoteMultiHostInstall.sh status <node name>
```

Note: The node name is optional. If you do not specify node name, the command uses the host name.

On Windows, view the status of the **Informatica VDS Node** service from Windows Administrative Tools.

Starting or Stopping the Vibe Data Stream Node on Linux

To start the VDS Node, run the following command from the `<VDS Node installation directory>/node/bin` folder:

```
./node.sh start <node name>
```

To stop the VDS Node, run the following command:

```
./node.sh stop <node name>
```

To start a VDS Node installed on a remote machine, run the following command:

```
./remote-node.sh start <node name>
```

To stop a VDS Node installed on a remote machine, run the following command:

```
./remote-node.sh stop <node name>
```

Note: The node name is optional. If you do not specify it, the command uses the host name.

Starting or Stopping the Vibe Data Stream Node on Windows

You can start or stop the VDS Node service from Windows Administrative Tools.

1. From Windows **Administrative Tools**, select **Services**.
2. Right-click the **Informatica VDS Node 230 <node name>** service.
3. Choose to start or stop the service.

Managing the Vibe Data Stream Node Logs

The VDS Node writes log messages to the <VDS Node Installation Directory>/logs/ directory. The VDS Node uses the log4j API. To change how the VDS Node logs messages, edit the `log4j.properties` file.

1. Open the `log4j.properties` file in the following directory:

   `<VDS Node Installation Directory>/node/config`
2. Set the value of `log4j.rootLogger`. You can log messages at one of the following levels:
   - FATAL. Writes FATAL messages to the log.
   - ERROR. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures and service errors.
   - WARNING. Writes FATAL, ERROR, and WARNING messages to the log. WARNING errors include recoverable system failures or warnings.
   - INFO. Writes FATAL, ERROR, WARNING, and INFO messages to the log. INFO messages include system and service change messages.
   - DEBUG. Writes FATAL, ERROR, WARNING, INFO, and DEBUG messages to the log. DEBUG messages are user request logs.

   By default, the VDS Node logs messages at the INFO level.
3. Set the path to the log file.
4. Save and close the file.

Managing the Informatica Domain

To manage the Informatica domain, you need to know how to start and stop the domain and manage log files.

Starting or Stopping Informatica Domain on Linux

On Linux, run infaservice.sh to start or stop the Informatica domain. By default, `infaservice.sh` is located in the following directory:

`<Informatica Administrator Tool installation directory>/tomcat/bin`

1. Go to the directory where `infaservice.sh` is located.
2. At the command prompt, enter the following command to start the domain:

   `./infaservice.sh startup`

   To stop the domain, enter the following command:

   `./infaservice.sh shutdown`

   Note: If you use a soft link to specify the location of `infaservice.sh`, set the INFA_HOME environment variable to the location of the Informatica installation directory.

Starting or Stopping Informatica Domain on Windows

You can start or stop the VDS Node service from Windows Administrative Tools.

1. From Windows **Administrative Tools**, select **Services**.

2. Right-click the **Informatica10.0.0** service.
3. Choose to start or stop the service.

Managing the Informatica Domain Logs

The domain log messages are written to the `<VDS installation directory/UMService/umsm.log` file by using the log4j API. To change how the domain logs messages, edit the `umsm.log` file.

1. Open the `umsm.log` file in the following directory:

   `<VDS installation directory/UMService`
2. Set the value of `log4j.rootLogger`. You can choose to log messages at one of the following levels:
   - FATAL. Writes FATAL messages to the log.
   - ERROR. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures and service errors.
   - WARNING. Writes FATAL, ERROR, and WARNING messages to the log. WARNING errors include recoverable system failures or warnings.
   - INFO. Writes FATAL, ERROR, WARNING, and INFO messages to the log. INFO messages include system and service change messages.
   - DEBUG. Writes FATAL, ERROR, WARNING, INFO, and DEBUG messages to the log. DEBUG messages are user request logs.

   The WARNING level is the default level of logging.
3. Set the path to the log file.
4. Save and close the file.

CHAPTER 10

Security

This chapter includes the following topics:

Security Overview

You can secure VDS components to protect them from threats from inside and outside the network on which the components run.

To enable security in VDS, select the Custom mode of installation. You can enable security for VDS only if security is configured for the Informatica domain and if the domain uses secure network authentication to authenticate users.

The type of user authentication you can configure for VDS depends on the type of user authentication that is configured for the Informatica domain. Secure communication between the VDS components is enabled if secure communication in enabled between the domain components. You can optionally set up secure communication between the VDS components. If you choose to set up secure communication, the installer displays additional screens where you can specify the security configuration.

You can secure VDS in the following ways:

- Authenticating users and services.
- Securing communication within components.
- Securing data storage.
- Securing source services and target services.
- Assigning privileges, roles, and permissions to users.

Authentication

User authentication in VDS depends on the type of authentication that is configured for the Informatica domain.

When you install VDS, the installer detects the type of authentication that is configured for the Informatica domain and uses the same authentication for VDS. The Informatica domain can use the following types of authentication:

- Native user authentication
- Lightweight Directory Access Protocol (LDAP) user authentication
- Kerberos network authentication

Native user accounts are stored in the Informatica domain and can only be used within the Informatica domain. Kerberos and LDAP user accounts are stored in an LDAP directory service and are shared by applications within the enterprise.

If you enable Kerberos authentication during installation, you must configure the Informatica domain to work with the Kerberos key distribution center (KDC). You must create the service principal names (SPN) required by the Informatica domain in the Kerberos principal database. The Kerberos principal database can be an LDAP directory service. You must also create keytab files for the SPNs and store it in the Informatica directory as required by the Informatica domain.

If you do not enable Kerberos authentication during installation, the installer configures the Informatica domain to use native authentication. After installation, you can set up a connection to an LDAP server and configure the Informatica domain to use LDAP authentication in addition to native authentication.

You can use native authentication and LDAP authentication together in VDS. VDS authenticates the users based on the security domain. If a user belongs to the native security domain, the Service Manager authenticates the user in the secure data storage. If the user belongs to an LDAP security domain, the Service Manager passes the user name and password to the LDAP server for authentication.

You cannot use native authentication with Kerberos authentication. If the Informatica domain uses Kerberos authentication, all user accounts must be in LDAP security domains. The Kerberos server authenticates a user account when the user logs in to the network. The Informatica client applications use the credentials from the network login to authenticate users in the Informatica domain. When you install VDS, the VDS installer detects that the Informatica domain uses Kerberos authentication and prompts you to provide the path to where VDS specific keytab files are created and stored.

Native Authentication

Native user accounts are stored in the Informatica domain and can be used only within the Informatica domain.

If the Informatica domain uses native authentication, the Service Manager stores all user account information and performs all user authentication within the Informatica domain. When a user logs in, the Service Manager uses the native security domain to authenticate the user name and password.

If the Informatica domain uses native authentication, you do not have to specify additional configuration for native authentication when you install VDS.

Lightweight Directory Access Protocol (LDAP) Authentication

You can configure the Informatica domain to allow users in an LDAP directory service to log in to Informatica client applications. The Informatica domain can use LDAP user authentication in addition to native user authentication.

To enable LDAP user authentication, you must set up a connection to an LDAP server and specify the users and groups from the LDAP directory service that can have access to the Informatica domain. You can use the Administrator tool to set up the connection to the LDAP server.

When you synchronize the LDAP security domains with the LDAP directory service, the Service Manager imports the list of LDAP user accounts with access to the Informatica domain into the LDAP security domains. When you assign privileges and permissions to users in LDAP security domains, the Service Manager stores the information in the domain configuration repository. The Service Manager does not store the user credentials in the domain configuration repository.

When a user logs in, the Service Manager passes the user name and password to the LDAP server for authentication.

Kerberos Network Authentication

You can configure VDS to use Kerberos network authentication if the Informatica domain uses Kerberos network authentication to authenticate users and services on a network. You can also configure Kerberos authentication between the Administrator Daemon and the Administrator tool when the Informatica domain uses SSL certificates to secure the domain. In this scenario, the Informatica domain need not use Kerberos network authentication.

Kerberos is a network authentication protocol which uses tickets to authenticate access to services and nodes in a network. Kerberos uses a Key Distribution Center (KDC) to validate the identities of users and services and to grant tickets to authenticated user and service accounts. In the Kerberos protocol, users and services are known as principals. The KDC has a database of principals and their associated secret keys that are used as proof of identity. Kerberos can use an LDAP directory service as a principal database.

Informatica can run on a network that uses Kerberos authentication with Microsoft Active Directory service as the principal database.

The Informatica domain requires keytab files to authenticate nodes and services in the domain without transmitting passwords over the network. The keytab files contain the service principal names (SPN) and associated encrypted keys. Create the keytab files before you create nodes and services in the Informatica domain.

For more information about preparing for a Kerberos authentication setup, see the *Informatica Vibe Data Stream Installation and Configuration Guide*.

Component Security

You can configure secure communication between the VDS components and between the source and target services.

VDS uses HTTP and UM to communicate between components. It uses SSL over HTTP to secure communication between components.

When you install VDS, you can enable secure communication for the Vibe Data Stream service. You can set up secure configuration. You can also secure the connection between web application and browsers.

You can enable secure communication in the following areas:

- VDS components. You can secure HTTP communication for the Administrator Daemon, the Administrator tool, and the VDS Node.
- Data storage.
- Source services and target services.

Secure Communication Within the Components

VDS uses the SSL certificate that Informatica provides to secure the domain.

When you use SSL certificates, you secure the connections between the following components:

- The Administrator Daemon and the Administrator tool
- The Administrator Daemon and the VDS Node
- The web application and browsers

Secure Data Storage

VDS encrypts sensitive data, such as passwords and secure parameters, before it stores the data in the storage.

VDS uses the encryption key that the domain uses to encrypt and decrypt sensitive data that is stored. You must keep the encryption key file and the keyword for the encryption key in a secure location.

To encrypt and decrypt sensitive data that is stored, VDS performs the following types of encryption:

- Encryption of the secure entity fields in the storage. To store the secure fields, the Administrator Daemon uses the infakeystore.kjs and infatruststore.jks SSL certificates that the domain provides.
- Encryption of secure data sent over the UM communication channel. When the Administrator tool sends secure data to the Administrator Daemon over the UM communication channel, the Administrator tool uses a key to secure the data, since SSL is not supported over UM communication. The key is generated in-memory and is valid only till the Administrator Daemon receives data securely and decrypts it.

Update Encryption Keys

VDS uses the `infakeystore.jks` SSL keystore file and the `infatruststore.jks` truststore file that the domain uses to encrypt and decrypt sensitive data that is stored. VDS uses the encryption key configured during domain installation for symmetric encryption and decryption of secure database properties, such as database password and secure jdbc parameters. When the domain keys are updated or changed, the data in VDS has to be decrypted using the old keys and encrypted using the new keys.

To decrypt the data in VDS using the old keys and encrypt the data again with the new keys, use the infacmd command line program. The command re-encrypts data in VDS after the security keys are updated in the domain.

During installation, VDS backs up the security keys in the following location:

```
<VDS installation directory>/admind/config/backedupSecurityKeys
```

Run the command after the keys are updated on the domain. VDS performs the following tasks:

- Decrypts the data with the keys from the backed up location.
- Encrypts the data with the new domain keys and stores it in the storage.
- Replaces the keys in the back up location with the new keys from the domain. The backup location always contains the keys with which data is currently encrypted.

Updating Security Keys

Use the infacmd command line program to update the security keys.

1. Undeploy all data flows.
2. Stop the domain.
3. Run the command to update the security keys on the domain.

   For information about the updating security keys on the domain, see the *Informatica Security Guide*.
4. Start the domain.
5. Run the updateSecurityKeys command.

   On UNIX, run the following command:

   ./infacmd.sh vds updateSecurityKeys

   On Windows, run the following command:

   ./infacmd.bat vds updateSecurityKeys
6. Stop and start the Administrator Daemon.
7. Deploy all data flows that you undeployed.

Secure Source Services and Target Services

You can use secure fields when you create some source services and target services.

You can use secure fields in the following types of entities:

Built-in Source Services and Target Services

You can use secure parameters to configure fields when you create built-in source services and target services. You can set the values for parameters in the **Parameters** tab of the Administrator tool. To use the parameters, specify the parameter name in the entity properties when you add the entities to a data flow.

Custom Source Services and Target Services

You can use secure fields when you create custom entities in VDS.

When you create a custom entity type, you describe the fields of the custom entity in the VDS plug-in XML file. When you describe the fields, you can specify that the fields are secure. The secure fields are securely communicated and stored in an encrypted form.

The following example shows how you can specify a secure field:

```
<tns:textControl>
                    <tns:name>pwd</tns:name>
                     <tns:displayName>Password</tns:displayName>
                     <tns:description>Enter value for password</tns:description>
                      <tns:mandatory>true</tns:mandatory>
                      <tns:stringTextField>
```

```
                <tns:maxLength>10</tns:maxLength>
                <tns:secure>true</tns:secure>
            </tns:stringTextField>
        </tns:textControl>
```

Note: If you install VDS in secure mode, the secure fields are securely communicated using encryption over HTTPS from the web browser to the Administrator tool and in an encrypted form over UM from the Administrator tool to the Administrator Daemon stored in the database. If you install VDS in normal mode, the value in the field appears masked in the Administrator tool but is not encrypted.

For more information about creating custom entity types, see the *Vibe Data Stream for Machine Data Developer Guide*.

Privileges and Roles

You can assign privileges, roles, and permissions to users or groups of users to manage the level of access users and groups can have and the scope of the actions that users and groups can perform. VDS uses the same roles, privileges and permissions that are defined in the domain during domain installation.

You can use the following methods to manage user and group access:

Privileges

Privileges determine the actions that users can perform in the Administrator tool. You can assign a set of privileges to a user to restrict access to the Vibe Data Stream Service. You can also assign privileges to a group to allow all users in the group the same access to the Vibe Data Stream Service.

Roles

A role is a set of privileges that you can assign to users or groups. You can use roles to manage assignments of privileges to users. You can create a role with limited privileges and assign it to users and groups that have restricted access to the Vibe Data Stream service. Alternatively, you can create roles with related privileges to assign to users and groups that require the same level of access.

Permissions

Permissions define the level of access that users have to an object. A user who has the privilege to perform a certain action might require permission to perform the action on a particular object. For example, to manage the Vibe Data Stream service, a user must have the privilege to manage services and permission on the specific service.

CHAPTER 11

High Availability

This chapter includes the following topics:

High Availability Overview

High availability refers to the uninterrupted availability of VDS components. In a VDS deployment, high availability eliminates a single point of failure and provides minimal service interruption in the event of failure. High availability deployments aim to minimize downtime and increase availability of a component.

The following features make the VDS components highly available:

- Restart and failover. A component can restart on the same machine or on a backup machine after the process becomes unavailable.
- Resilience. A VDS deployment can tolerate temporary failures. The VDS components are resilient to outage.

High availability features in VDS are available if you install VDS using custom installation mode on an Informatica domain high-availability setup.

When you plan a highly available VDS environment, you must also configure high availability for the VDS components and components that are external to VDS. Internal components include the Administrator Daemon, Apache ZooKeeper, the VDS Nodes, and the data flows. External components include Informatica domain and databases.

For information about Informatica domain high availability, see the *Informatica Administrator Guide*.

Restart and Failover

To maximize operation time in the event of a failure, the VDS components can restart or fail over processes to another machine.

Based on the type of installation, you can also configure backup machines for the VDS components. You can configure restart or failover for the following components:

- Administrator Daemon
- Apache ZooKeeper
- VDS Node

Administrator Daemon Restart and Failover

Install the Administrator Daemon on the nodes where you have installed the Informatica domain.

The following situations describe how the Administrator Daemon restarts or fails over to a standby machine:

Informatica domain becomes unavailable

If the Informatica domain becomes unavailable on the master gateway node, it fails over to a backup node. The Administrator Daemon failover depends on the domain failover. When the domain fails over to the backup node, the Administrator Daemon that is running on the same node also fails over to the backup node.

Administrator Daemon process becomes unavailable

If the Administrator Daemon process becomes unavailable on the master gateway node, the process can restart.

On Linux, the Administrator Daemon process script `admind.sh` runs the `processmonitor-admind.sh` script which detects if the Administrator Daemon, LBMRD, and Apache ZooKeeper are running. If one of the processes stop unexpectedly, the `processmonitor-admind.sh` script restarts only the process that has stopped.

On Windows, the Administrator Daemon, LBMRD, and Apache ZooKeeper run as services. The recovery action for the services is set as **Restart the Service**. If one of the services stop unexpectedly, the service that has stopped is restarted.

The LBMRD runs on the same machine as the Administrator Daemon. When the Informatica domain fails over to a backup node, the Administrator Daemon detects that the backup node is the new master gateway node and uses the LBMRD that is running on the master gateway node for topic resolution.

Starting or Stopping the Administrator Daemon Process Monitor Script

You can start or stop the `processmonitor-admind.sh` script on Linux.

To start the `processmonitor-admind.sh` script, run the following command in the location where you have installed the Administrator Daemon:

```
admind.sh processmonitoron
```

To stop the `processmonitor-admind.sh` script, run the following command:

```
admind.sh processmonitoroff
```

Note: The script is started by default.

Vibe Data Stream Node Restart and Failover

To maximize operation time in the event of a failure, the VDS Node can restart on the same machine or fail over to a backup machine.

The VDS Node process script `node.sh` runs the `processmonitor-node.sh` script which detects if the VDS Node is running. If the VDS Node process stops unexpectedly, the `processmonitor-node.sh` script restarts the VDS Node.

The `./remote-node.sh` script detects if a VDS Node that is installed on a remote machine is running. If the remote VDS Node stops unexpectedly, the `./remote-node.sh` script restarts it.

When a VDS Node process fails, it fails over to one of the backup machines which share the VDS Node name. The VDS Node on the active machine starts collecting data from the data source.

Starting or Stopping the Vibe Data Stream Node Process Monitor Script

You can start or stop the `processmonitor-node.sh` script on Linux.

To start the `processmonitor-node.sh` script, run the following command in the location where you have installed the VDS Node:

```
node.sh processmonitoron <node name>
```

To stop the `processmonitor-node.sh` script, run the following command:

```
node.sh processmonitoroff <node name>
```

It is optional to specify the node name. If you do not specify the node name, the command sets the host name as default node name.

Note: The script is started by default.

Starting or Stopping the Remote Vibe Data Stream Node Process Monitor Script

You can start or stop the `processmonitor-node.sh` script on remote VDS Nodes on Linux.

To start the `processmonitor-node.sh` script, run the following command in the remote machine where you have installed the VDS Node:

```
remote-node.sh processmonitoron <node name>
```

To stop the `processmonitor-node.sh` script, run the following command:

```
remote-node.sh processmonitoroff <node name>
```

It is optional to specify the node name. If you do not specify the node name, the command sets the host name as default node name.

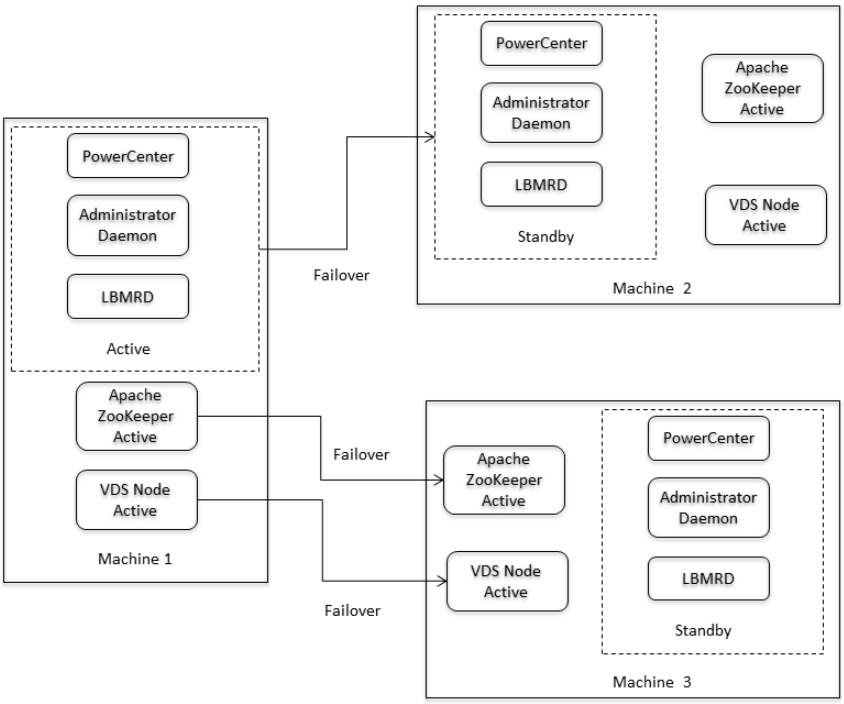
Note: The script is started by default.

ZooKeeper Failover

In a VDS high-availability environment, ZooKeeper has an active-active failover configuration. In an active-active failover configuration, all machines on which ZooKeeper is installed are in active mode. If one active machine fails, then the ZooKeeper on another active machine handles the functions of the ZooKeeper on the machine that failed.

Example

The following image shows some sample connections in a high-availability deployment:



In the deployment, the Administrator Daemon, LBMRD, Apache ZooKeeper, and the VDS Node run on Machine 1, Machine 2, and Machine 3, on which PowerCenter also runs.

The following failover scenarios can occur:

Informatica Domain on Machine 1 becomes unavailable

If the domain becomes unavailable on Machine 1, it fails over to Machine 2. The Administrator Daemon and LBMRD on Machine 1 also fail over to Machine 2. Apache ZooKeeper and the VDS Node continue to function on Machine 1.

Machine 1 becomes unavailable

If Machine 1 becomes unavailable, the Informatica domain, Administrator Daemon, and LBMRD can fail over to Machine 2 or Machine 3. The failover mechanism of Apache ZooKeeper and the VDS Node is independent of Informatica domain and Administrator Daemon failover. Each one of them could failover to either Machine 2 or Machine 3.

Resilience

A VDS deployment can tolerate temporary failures. The Administrator Daemon, LBMRD, and the VDS Node are resilient to outage. If one of the components shuts down, the `processmonitor` script tries to restart the component.

The data connections in a data flow are resilient to temporary failures. If you use a UM data connection in the data flow, you can configure multiple IP address and port combinations for LBMRD for topic resolution. If one LBMRD becomes unavailable, the Administrator Daemon uses the LBMRD that is available for topic resolution. The source services continue to send data to the target services.

Configuring High Availability in Vibe Data Stream

To minimize downtime of VDS, configure high availability for the components both at design time and at run time. Configure high availability for the Administrator Daemon, ZooKeeper and VDS Node to achieve design-time high availability. Configure data flow high availability to achieve run-time high availability.

You can configure high availability for the Administrator Daemon and ZooKeeper during installation and configure high availability for VDS Nodes and entities after installation.

For information about configuring high availability during installation, see the *Installation and configuration Guide*.

Design-Time High Availability

To achieve design-time high availability, configure high availability for the VDS components.

Configure high availability for the following VDS components:

- Administrator Daemon. Install the Administrator Daemon on the gateway nodes of the Informatica domain.
- Apache ZooKeeper. Configure multiple ZooKeeper servers during installation or after installation.
- VDS Node. Install and start VDS Node instances on multiple machines.

Administrator Daemon High Availability

You can configure high availability for the Administrator Daemon only if you selected custom installation type while installing VDS.

To configure high availability for the Administrator Daemon, perform the following steps:

1. Install the Administrator Daemon on the gateway machines of the Informatica domain so that the Administrator Daemon can fail over with the domain.
2. Create the Vibe Data Stream Service on the master gateway machine of the Informatica domain.

Note: If you use a custom source service or target service, register the service on the primary machine and upgrade the service on the backup machines.

Apache ZooKeeper High Availability

You can configure high availability for ZooKeeper either during VDS installation or after installation. To configure high availability, deploy ZooKeeper in a cluster known as an ensemble.

To configure ZooKeeper high availability, perform the following configurations:

Configure multiple ZooKeeper servers

You can configure multiple servers for ZooKeeper in the following ways:

- During installation. When you install VDS, you can specify the IP address and port combinations of the machines in the ZooKeeper ensemble. In the installer wizard, configure the **Apache ZooKeeper IP Address and Ports** property.
  The IP address and port combination has the following format:

  `<IP Address>:<port>`

  You can enter multiple comma-separated IP address and port combinations. If one instance of ZooKeeper fails, the Administrator Daemon connects to the next available ZooKeeper server.
- After installation. After you install VDS, configure the `zoo.cfg` file. Specify the IP address and port combinations of the machines in the ZooKeeper ensemble in the following format:
  `server.id=host:port:port`

For information about configuring a ZooKeeper ensemble, see the *ZooKeeper Administrator's Guide*, available at http://zookeeper.apache.org/doc/r3.4.6/zookeeperAdmin.html.

Configure VDS to work with a ZooKeeper ensemble

Specify the complete list of ZooKeeper servers in the Administrator Daemon and VDS node configuration files.

The Administrator Daemon configuration file, named `admind.cnf`, is available in the `<VDS Node installation directory>/admind/config/` directory.

The VDS node configuration file, named `node.cnf`, is available in the `<VDS Node installation directory>/node/config/` directory.

Specify the list of ZooKeeper servers, type `zkservers=` and then type the names of the ZooKeeper servers separated by commas. For example,
`zkservers=<IP_address_1>:<port>,<IP_address_2>:<port>,<IP_address_3>:<port>,...`

Update the VDS Node configuration file on each machine on which you install the VDS Node.

Vibe Data Stream Node High Availability

To configure high availability for the VDS Node, run instances of the VDS Node on multiple machines.

On Linux, start multiple VDS Nodes with the same name. On Windows, create multiple VDS Node Services and start them.

For example, if you want an entity that is running on a VDS Node named `NodeA` to be highly available, start `NodeA` on multiple machines.

Starting the Vibe Data Stream Node on Linux

To start the VDS Node, run the following command from the <VDS Node installation directory>/node/bin folder:

```
./node.sh start <node name>
```

To start a VDS Node installed on a remote machine, run the following command:

```
./remote-node.sh start <node name>
```

Note: The `<node name>` is optional. If you do not specify `<node name>`, the node name is set as the host name.

Create Multiple Vibe Data Stream Node Services on Windows

By default, the VDS Node installation process creates one Windows service. If you want to configure high availability, you can create multiple VDS Node services.

To create a VDS Node service run the following command from the `<VDS Node installation directory>\node\bin` folder:

```
node.bat install <node name>
```

Note: Do not include spaces in the node name. To create multiple services, run the command multiple times.

Run-Time High Availability

To achieve run-time high availability, configure high availability for the entities in a data flow.

Entity High Availability

You can configure high availability for the data flows if you have configured high availability for the VDS Nodes.

Ensure that the source services and target services are accessible on all the machines where the VDS Node is running.

You can configure high availability for the entities if you use the Ultra Messaging data connection in the data flow.

Configure high availability when you create a data flow and add a data connection. If you select Unicast topic resolution type, specify multiple IP address and port combinations for LBMRD when you configure the **Resolver Address** property.

Note: If you use a WebSocket data connection in the data flow, only configure high availability for the VDS Node.

CHAPTER 12

Disaster Recovery

This chapter includes the following topics:

Disaster Recovery Overview

Disaster recovery includes recovery of all VDS components and data flows after a catastrophic event. When a catastrophic event occurs, you can recover the VDS components and data flows from a disaster recovery site and resume normal operations.

Perform the following steps to recover VDS:

1. Replicate the VDS installation.
2. Back up data flows
3. Optionally, back up the node groups.
4. Set parameters and values for the entity properties.
5. Optionally, replicate source files and position files.
6. Restore VDS from the disaster recovery site.

Step 1: Replicate the VDS Installation

Replicate the VDS installation at the primary and disaster recovery sites. Install the Administrator Daemon, Administrator tool, VDS Nodes, and ZooKeeper on the primary and disaster recovery sites.

Step 2: Back Up Data Flows

Back up the data flows that you create on the primary site periodically. To back up the data flows, export the data flows from the primary site and import them to the disaster recovery site.

For example, you can back up the data flows whenever there is a change in the data flow configuration.

Related Topics:

- "Export and Import Data Flows" on page 122

Step 3: Back Up The Node Groups

Optionally, back up the node groups that you create on the primary site.

Export the node groups from the primary site and import them to the disaster recovery site. You can also use different nodes and node groups on the disaster recovery site.

Related Topics:

- "Export and Import Node Groups" on page 100

Step 4: Set Parameters

Set parameters and values for the entity properties on both the primary site and the disaster recovery site. You can set parameters and values for entity properties on the primary site. When you need to use the properties on the disaster recovery site you can change it at a single location and use the parameters.

The following properties are examples of some of the parameters that you can set:

- Unicast resolver daemon address
- Load balancer IP address

Related Topics:

- "Setting Values for Parameters" on page 93

Step 5: Replicate Source Files and Position Files

If you use File or Static File source services in your data flow, VDS creates a position file that tracks where the data has been read until. VDS saves the position file in the following directory:

```
<Source file location>/.VDSPos
```

To back up the position file directory, copy the directory to the respective source folder on the disaster recovery site periodically.

Note: If you use a Syslog UDS source in your data flows, you will not be able to recover data in the event of a catastrophic failure.

Step 6: Restore VDS from the Disaster Recovery Site

In the event of a catastrophic event, recover VDS from the disaster recovery site.

To recover VDS, perform the following tasks:

- Verify that all VDS components are running on the disaster recovery site.
- After disaster occurs, deploy the data flows on the disaster recovery site.
- Verify that the sources send data to the source services on the disaster recovery site.

CHAPTER 13

Monitoring Vibe Data Stream Entities

This chapter includes the following topics:

Monitoring Vibe Data Stream Entities Overview

You can monitor data flows, VDS Nodes, source services, and target services. You can monitor the states of the entities and view statistics for source services and target services.

You can use the VDS monitoring option in the following ways:

- Identify issues with VDS Nodes.
- Monitor threshold values for entities.
- View data connections and transformations.

You can monitor VDS data flows and entities on the **Monitoring** tab in the Administrator tool.

Viewing the Monitoring Tab

You can view the **Monitoring** tab in the following ways:

- Enter `http://<host>:<port>/administrator` in the address field of the web browser, if you have administrator privileges. Click the **Monitoring** tab.
- Enter `http://<host>:<port>/monitoring` in the address field of the web browser, if you have monitoring privileges.

Monitoring Tab Layout

When you select the Vibe Data Stream Service, the entity types appear in the **Monitoring** tab.

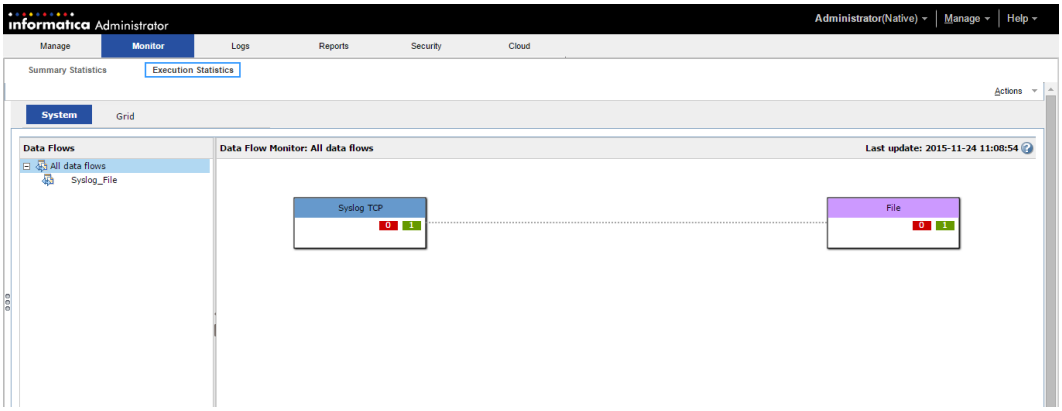
The Monitoring tab consists of the following tabs:

- System view
- Grid view

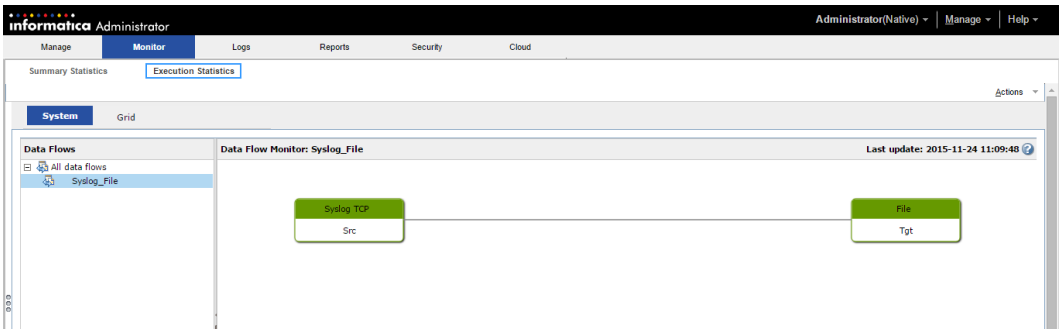
System View

The **System** view displays all the data flows in the VDS deployment.

The following image shows the **System** view with a view of all deployed data flows:



The following image shows the **Data Flow Monitor** panel with state of the data flow:



The panes that appear in the **Monitoring** tab depend on the entity that you select. The **Monitoring** tab has the following panels:

- **Data Flows** panel. Displays a list of all deployed data flows.
- **Data Flow Monitor** panel. Displays data flows and associated entities. Select **All Data Flows** in the **Data Flows** pane to view source services and target services and the states of VDS Nodes they are running on. Select a data flow to view all associated entities and states.

User Preferences

When you log in to the Administrator tool, the settings that you make in the **Grid** view of the **Monitoring** tab are preserved across different sessions.

The following preferences are preserved across different sessions:

Table preferences

The settings that you make for the data flow, nodes, source service, and target service tables persist across sessions. You can set the following table preferences:

- Table columns that you want to display. Click the arrow in the table column heading. Select **Columns** and choose the columns you want to display or hide.
- Table column sorting. Click on a column name to sort the table by status. To sort the columns alphabetically, click the arrow in the table column heading and select **Sort Ascending** or **Sort Descending**.
- Number of rows in the table.
- Active page number.
- Number of rows per page

Chart preferences

The following chart preferences that you set persist across sessions:

- Statistics for which you want to display charts.
- Time interval for which you want to view the information, such as, 15, 30, or 60 minutes.
- Size of the charts.

Filter preferences

The following filter preferences that you set persist across sessions:

- Filters that you apply.
- Associated filters.

Page preferences

The following page preference that you set persists across sessions:

- State of panels. The panels can be expanded or collapsed.

To reset the preferences that you have set, click **Clear preferences** in the **Monitoring** tab.

Monitoring Data Flows

You can monitor deployed data flows in the **Data Flow Monitor** pane of the **Monitoring** tab. You can view details of the associated VDS entities and drill down to specific information for each.

You can view the states of the source services an target services associated with a data flow. Click a data flow in the **Data Flows** pane, to view it in the **Data Flow Monitor** pane.

The state of a data flow is the aggregate of the states of the entities in the data flow. A data flow is active if the associated source services and target services are active. When you click an entity in the **Data Flow Monitor** pane, the **Data Flows** pane displays the entities associated with the data flow and the state of the data flow.

The following table describes the data flow states and colors that represent the state:

State	Color in the Data Flows pane	State Description
Active	Green	Indicates that all entities in the data flow are active.
Inactive	Red	Indicates that at least one entity in the data flow is inactive.

Monitoring Source Services

You can monitor the source services associated with a data flow in the **Data Flow Monitor** pane of the **Monitoring** tab.

You can view the following information for a source service:

- The state of the source service. A source service appears without issues when all associated VDS Nodes are active. A source service appears with issues when one associated VDS Node is not active.
- The type of source service and associated transformation.
- The data flow on which the source service is active. When you click a source service in the **Data Flow Monitor** pane, the **Data Flows** pane shows the data flows that the source service is associated with and the data flow state.
- The number of nodes on which the source service is active or inactive. When you click **All Data Flows** the count appears in colored boxes within the associated source service.
- The source service statistics. Click the source service to view the statistics in the **Entity Overview** pane.

The following table describes the source services states and colors that represent the states:

State	Color	State Description
Without issues	Green	Indicates that all VDS Nodes on which the source service is running are active.
With issues	Red	Indicates that at least one associated VDS Node is inactive.

Monitoring Target Services

You can monitor the target services associated with a data flow in the **Data Flow Monitor** pane of the **Monitoring** tab.

You can view the following information for a target service:

- The state of the target service. A target service appears without issues when all associated VDS Nodes are active. A target service appears with issues when even one associated VDS Node is inactive.
- The type of target service and associated transformation.
- The data flow on which the target service is active. When you click a target service in the **Data Flow Monitor** pane, the **Data Flows** pane shows the data flows that the target service is associated with and the data flow state.
- The number of nodes on which the target service is active or inactive. When you click **All Data Flows** the count appears in colored boxes within the associated target service.
- The target service statistics. Click the target service to view the statistics in the **Entity Overview** pane.

The following table describes the target services states and colors that represent the states:

State	Color	State Description
Without issues	Green	Indicates that all VDS Nodes on which the target service is running are active.
With issues	Red	Indicates that at least one associated VDS Node is inactive.

Monitoring Vibe Data Stream Nodes

You can monitor the VDS Nodes associated with a data flow in a VDS deployment in the **Data Flow Monitor** pane of the **Monitoring** tab.

You can view the following information for a VDS Node:

- The number of VDS Nodes associated with each source service and target service. The count appears within the associated source service and target service.
- The VDS Nodes appear as colored boxes within the source service and target service.
- The state of a VDS Node. Click a data flow in the **Data Flow Monitor** pane to see if the VDS Node is active or inactive.

The following table describes the colors associated with the VDS Node count:

Color	State Description
Green	Indicates the active VDS Nodes for a source service or target service.
Red	Indicates the inactive VDS Nodes for a source service or target service.

Monitoring Vibe Data Stream Node High Availability

You can monitor the VDS Node high availability on the **Monitoring** tab in the Administrator tool.

Select a source service or a target service in the **Data Flow Monitor** pane to see the VDS Node details in the **Entity Summary** pane. The pane displays the names of all hosts on which the VDS Node is running, and indicates the active and standby hosts.

Grid View

You can view a list of data flows, nodes and entities in the VDS deployment in the **Grid** view. The **Grid** view displays tables for each entity. Each table lists entities and related information. You can select the tables you want to view and the number of rows in each table. Click the table heading to expand or collapse a table.

You can view the following entities in the **Grid** view:

Data Flows

You can view a list of deployed data flows in the VDS deployment and related statistics. Click on a data flow name to view the data flow in the **System** view. Select a data flow to display tables for related entities.

Nodes

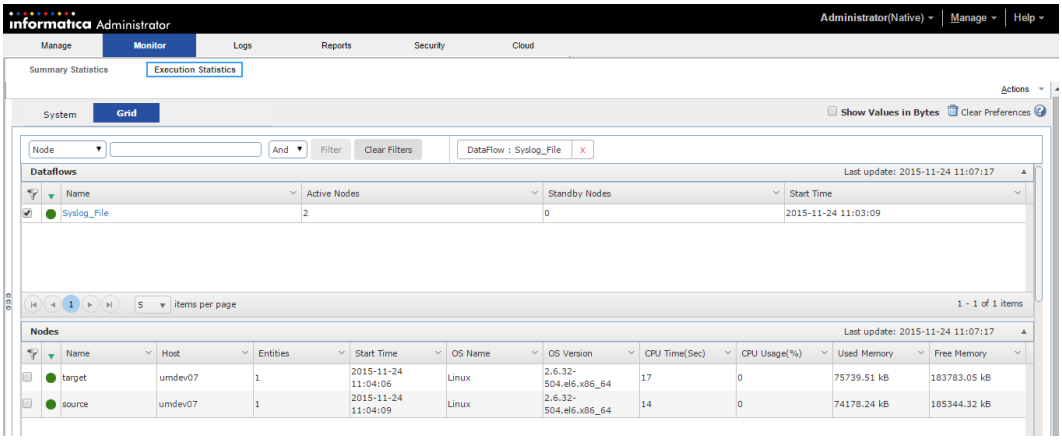
You can view the list of VDS Nodes that are mapped to the entities in the VDS deployment and the related statistics.

Entities

You can view a list of source services, target services, and transformations for deployed data flows in the VDS deployment. Filter by the node name or data flow name to view the entities. The **Entities** table displays statistics for the following entities:

- Source Services
- Target Services
- Transformations

The following image shows the **Grid** view:



Filters

You can apply a filter to drill down to specific information on VDS data flows, nodes, and entities. When you view the state or information of any component, you can use filters to view related components. The filters you apply in the **System** view or **Grid** view persist across views. You can select a component and add it to the filter.

Use the **Search** section to search for VDS data flows, nodes, and entities, and then apply filters. You can use a string or regular expression for the search. The search results match an exact string or a partial string. A regular expression describes a range or pattern of values that a filter condition can contain.

Related Topics:

- "Regular Expressions" on page 177

Entity Charts

You can view charts for source service and target service statistics in the **Charts** panel of the **Grid** view. Statistics that you can add to charts appear with a chart icon.

To add a chart, click the chart icon. The chart appears in the **Grid** view. The selected statistic either appears as a new chart or in an existing chart. Y

You can select the time interval for which you want to plot the chart, such as, 15 minutes, 30 minutes, or 1 hour. The line charts show trends in the values being mapped. The tool tip shows the value of the statistic at a specific time. The charts can show either the current time range values or values from an earlier time range. The charts display statistics in 5-second intervals for a period of 60 minutes. To stop live data update, click **Pause**.

Adding a Chart in the Grid View

On the **Charts** panel in the **Grid** view, you can add charts for the entity statistics that you want to monitor.

1. To view a list of source or target services, click the table header.

   For example, to view a list of all flat file sources, click the **Flat File** table header.

   The table displays a list of all entities. A **Chart** icon appears in the columns for the statistics that you can chart.

2. Click the **Chart** icon to add the chart to the **Charts** panel.

Chart Options

When you add a chart in the Grid view, it appears with default properties. You can change the properties of the charts.

You can set the following chart options:

Chart size

Select the chart size from the list. The chart size is a global setting. All charts appear in the size you select. You can select a small, medium, or large chart size. Default size is small.

Number of charts in a row

Select the number of charts of charts to display in a row. You can display 1, 2, 3, or 4 charts in a row.

Chart interval

The line charts plot the statistics with a granularity of five seconds for a period of one hour. You can select the time interval for which you want to plot the chart. You can select a time interval of 15 minutes, 30 minutes, or 1 hour.

Start or stop live data

When you add a chart to the chart panel, the chart displays live data. To stop live data, click **Pause**. To display live data, click **Play**.

Deleting a Chart from the Grid View

You can either delete all charts or delete them one by one from the chart panel.

To delete a chart, click **Close** on the chart or in the columns for the statistics that are charted. To delete all charts, click **Close All** on the chart panel.

Vibe Data Stream Statistics

VDS employs real-time monitoring and collects statistics from a VDS deployment.

You can view statistics for the following component and entities:

- VDS Node
- Data flows
- Aggregator
- Source Service Type
- Target Service Type
- Transformation Type

Data Flow Statistics

You can monitor the deployed data flows in the **Grid** view.

The **Data Flows** table displays the following information:

- State of the data flow.
- Name of the data flow.
- Number of active and standby nodes that are mapped to data flow.

- Time when the data flow was deployed.

Vibe Data Stream Node Statistics

You can monitor the VDS Nodes associated with a data flow in a VDS deployment in the **Grid** view.

The **Nodes** table displays the following information:

- State of the node.
- Name of the node.
- Host name or IP address on which the node is running.
- Number of source services and target services mapped to each node.
- Time when the node was started.
- The operating system of the host machine.
- The version of the operating system.
- CPU time.
- CPU usage.
- Used memory.
- Free memory.

Aggregator Statistics

You can view the following statistics for an aggregator:

Bytes Received

The number of bytes received by the aggregator.

Events Received

The number of events received by the aggregator.

Receive Rate (Per Sec)

The number of events received per second.

Bytes Sent

The number of bytes sent by the aggregator.

Events Sent

The number of events sent by the aggregator.

Send Rate (Per Sec)

The number of events sent per second.

Events Reassigned

The number of events that the source resends to the aggregator with a reassigned flag. The source sets the reassigned flag if it does not receive acknowledgment from the aggregator. The statistic displays meaningful values when you select the load balancing messaging mode in the Ultra Messaging data connection or the acknowledgment mode in the WebSocket data connection while configuring data flows.

Source Service Statistics

Filter on a data flow in the **Grid** view to view related source statistics in the **Entities** panel.

You can view the following common statistics for all source services:

State of the source service

The state appears as a colored box in the table. The source service can appear in one of the following colors:

- Green. Indicates that all VDS Nodes on which the source service is running are active.
- Red. Indicates that at least one associated VDS Node is inactive.

Name

Name of the source service.

Node Name

Name of the node to which the source service is mapped.

Data Flow Name

Name of data flow to which the source service is mapped.

Active Host

Name of the active machine on which the source service is running.

Standby Hosts

The standby machines of the source service.

Bytes Sent

Number of bytes sent by the source service.

Events Sent

Number of events sent by the source service.

Events to be Sent

Number of events that the source service is yet to send.

Send Rate (Per Sec)

Number of bytes sent every second.

Events not Delivered

Number of events not delivered to the target service. If you have included transformations on the source service in the data flow, this statistic includes the number of events that are not delivered after all the transformations are applied.

This statistic shows a value of 0 for all the source services except File source service and HTTP source service.

You can view additional statistics for the File, Static File, HTTP, TCP, MQTT, Syslog TCP, Syslog UDS, and WebSocket source services.

Note: All the statistics, except **Bytes Sent** and **Events Sent** are reset when you restart the VDS Node on which the source services are running.

File Source Service Statistics

In addition to the common source service statistics, you can view the following statistics for the File Source service:

Events Dropped

The number of events dropped while processing the source data.

Files Written to Target

The number of files sent to the target service by the source service.

Files to be Processed

The number of files to be processed. This number includes the file that is being processed and the pending files. This statistic shows a value of 0 if **File Name is Regular Expression** was not selected when the source service was created.

Error while Moving Files

The number of errors that occurred while moving the files. This statistic is applicable if the **Processed File Directory** property was specified when the source service was created.

HTTP Source Service Statistics

You can view the following statistics:

Events Dropped

The number of events dropped by the source service while processing the source data.

Events to be Processed

Maximum number of messages that can be stored in the internal queue of the source.

MQTT Source Service Statistics

You can view the following statistics:

Events dropped

Events dropped by the source service while processing the source data. This statistic increases when the length of the message that is read is greater than the event size.

Events to be Processed

Maximum number of messages that can be stored in the internal queue of the source.

Static File Source Service

In addition to the common source service statistics, you can view the following statistics for the Static File source service:

Events Dropped

The number of events dropped by the source service while processing the source data. If a read operation that a source service performs returns an event that is equal to or less than the specified event size, the source publishes the event only if the records in the event contain a delimiter. This statistic increases when there are issues with the event size, that is, the source data does not contain a delimiter.

Files Written to Target

The number of files sent to the target service by the source service. This statistic does not include the files that are ignored by the source service while the source data is being read.

Error while Moving Files

The number of errors that occurred while moving the files. This statistic is applicable if the **Processed File Directory** property was specified when the source service was created.

Files to be Processed

The number of files yet to be processed by the source service.

Files Ignored

The number of files ignored by the source service. For example, if the delimiter is EOF and the value of the **Maximum Event Size** property is less than the size of the file, this statistic is updated.

Syslog TCP and UDS Source Service Statistics

You can view the following statistics:

Events Dropped

The number of events dropped by the source service while processing the source data.

Concurrent Connections

The number of TCP clients currently connected to the source service.

Max Concurrent Clients

The maximum number of TCP clients connected to the source service since the time the source service is up.

TCP Source Service Statistics

You can view the following statistics:

Events Dropped

The number of events dropped by the source service while processing the source data.

Concurrent Connections

The number of TCP clients currently connected to the source service.

Max Concurrent Clients

The maximum number of TCP clients connected to the source service since the time the source service is up.

WebSocket Source Service Statistics

You can view the following statistics:

Events Dropped

The number of events dropped by the source service while processing the source data.

Events to be Processed

Maximum number of messages that can be stored in the internal queue of the source.

Target Service Statistics

Filter on a data flow in the **Grid** view to view target service statistics.

You can view the following statistics for target services:

State of the target service

The state appears as a colored box in the table. The target service can appear in one of the following colors:

- Green. Indicates that all VDS Nodes on which the target service is running are active.

- Red. Indicates that at least one associated VDS Node is inactive.

Name

Name of the target service.

Node Name

Name of the node to which the target service is mapped.

Data Flow Name

Name of data flow to which the target service is mapped.

Bytes Received

The number of bytes the target service receives.

Events Received

The number of events the target service receives.

Receive Rate (Per Sec)

The number of bytes the target service receives every second.

Events Reassigned

The number messages that the source service resends to the target service when it has not received acknowledgment from the target service.

You can view additional statistics for the File, HDFS, and Cassandra target services.

Cassandra Target Service Statistics

You can view the following statistics:

Events with Invalid JSON Objects

The number of events that contain JSON fields that do not match the data type of the columns in the Cassandra database.

Events with Excess JSON Fields

The number of events that contain JSON fields that do not have a corresponding column in the Cassandra database.

Events Written

The number of events written to the Cassandra database.

Events Not Written

The number of events that were not written to the Cassandra database.

File Target Service Statistics

You can view the following statistic:

Files Rolled Over

Number of files rolled over.

HDFS Target Service Statistics

You can view the following statistic:

Files Rolled Over

Number of files rolled over.

Transformation Statistics

You can monitor the transformations in a data flow in the **Grid** view.

You can view the following statistics for transformations:

State of the transformation

The state appears as a colored box in the table. The transformation can appear in one of the following colors:

- Green. Indicates that all VDS Nodes on which the transformation is running are active.
- Red. Indicates that at least one associated VDS Node is inactive.

Name

Name of the transformation.

Node Name

Name of the node to which the transformation is mapped.

Data Flow Name

Name of data flow to which the transformation is mapped.

Active Host

Name of the active machine on which the transformation is running.

Standby Hosts

The standby machines of the transformation.

Events Received

The number of events received by the transformation.

Bytes Received

The number of bytes received by the transformation.

Events Sent

The number of events sent by the transformation after transforming the data.

Bytes Sent

The number of bytes sent by the transformation after transforming the data.

Time Taken for Transformation

The time taken for transforming the data.

APPENDIX A

Troubleshooting

This appendix includes the following topics:

Troubleshooting Licenses

I get a ZooKeeper client connection exception in the Administrator Daemon logs.

This error can occur if the Administrator Daemon and Apache ZooKeeper are not running when you apply or update the VDS license. Start the Administrator Daemon and Apache ZooKeeper before you apply or update the VDS license.

The data flows do not appear correctly in the Administrator tool.

This error can occur if you have updated the Enterprise license to a Free license, and the VDS Node name is not the same as the host name. A license error occurs in the `<VDS installation folder>/node/logs/<node name>-node.log` file. When you start the VDS Node, use the host name as the VDS Node name.

I get a license error in the Administrator tool.

This error might occur in the following situations:

- The license is not valid. This error can occur if you create the Vibe Data Stream Service when the Administrator Daemon and Apache ZooKeeper are not running. Start the Administrator Daemon and Apache ZooKeeper, and apply or update the VDS license.
- The license has expired. Contact Informatica Global Customer Support to update the license.

I get the following error in the Administrator tool.

```
Admind not running.
```

This error might occur after a fresh installation if there are database connectivity issues. The license details and database configuration are not sent to the Administrator Daemon. Click **Send license and database configuration to Administrator Daemon** in the **Contents** panel.

Troubleshooting Vibe Data Stream Node Issues

I get the NoClassDefFoundError error message in the VDS Node log file.

This error occurs if the fully qualified class name for the custom source service or target service plug-in is not defined correctly in `vdsplugin.xml`. Verify that class name is defined correctly in the `pluginClass` element.

For more information about custom source services and target services, see the *Vibe Data Stream for Machine Data Developer Guide*.

When I create a UDP, TCP, Syslog, HTTP(S), or WebSocket(S) source service, I get an error in the VDS Node log file indicating that the address is already in use.

This error occurs if the source service tries to use a port that is already in use. The source service does not start. Verify that port is available and create the source service.

The VDS Node does not start after installation.

This error can occur if you have configured the VDS Node incorrectly. Check the `<VDS Installation Directory>/node/logs/<node-name>.out` log file for errors and correct them. Also see the `<VDS Installation Directory>/node/logs/<node name>-node.log` for other errors.

A VDS node logs the following message to the VDS Node log file:

```
HTTP content length exceeded <n> bytes
```

This error occurs if the content length exceeds the block size that you configure for an HTTP source service on the node. Increase the block size for the HTTP source service or reduce the content length so that it does not exceed the block size. When you configure the block size, consider any delimiters that the HTTP source might include in the message body.

I get a warning for messages dropped in the VDS Node log file.

This error might occur when you use an HTTP source service to write or send data to any target. The VDS Node log file displays warnings for dropped messages and the HTTP client receives a REQUEST_TIMEOUT error from the HTTP source service. This error occurs in the following situations:

- The target service is consuming messages at a slow rate. Verify that the target does not have bottlenecks. To make the target faster, you can configure load balancing.
- The target service is down. Verify that the target service is running.

When I start the VDS Node and use an HDFS target, the VDS Node fails to start and I get the following error:

```
JAVA_HOME is not set and could not be found
```

This error might occur when you use an HDFS target and have set the `HADOOPBASEDIR` environment variable.

To resolve this error, perform the following tasks:

- Verify that the Hadoop installation is correct.
- Configure the `JAVA_HOME` environment variable as follows:
  `export JAVA_HOME=<Java home directory>`

The HTTP client receives the following error message from the HTTP source service:

```
408 REQUEST TIMEOUT
```

When you configure an HTTP source service in synchronous mode and set the messaging mode to load balancing, the HTTP source service receives an acknowledgement from the target service after the target service receives data from the HTTP source service. However, if the target service is down, the HTTP source service sends a timeout error to the HTTP client when the client sends data to the HTTP source service. To resolve this issue, ensure that the target service is running.

I see the status of the Vibe Data Stream Service as available in the Navigator but I see the status of the service as unavailable in the Contents panel.

You might find this issue on the **Domain** tab of the Administrator tool. When you move the mouse pointer over the Vibe Data Stream Service in the Navigator, the service status appears as available. However, the service status appears as unavailable when you move the pointer over the Vibe Data Stream Service in the **Contents** panel. This service status does not impact the functionality of the Vibe Data Stream Service and you can ignore the status.

Troubleshooting Administrator Daemon Issues

I get the following error message in the Administrator tool:

```
Monitoring Web Application request to the Administrator Daemon timed out. Verify that
the Administrator Daemon is up,
verify the connectivity between the Administrator Tool and Administrator Daemon.
```

This error can occur in the following situations:

- Network latency. Reload the page. If the issue persists, contact your IT administrator.
- The Administrator Daemon is out of memory. Check the Administrator Daemon logs for `OutOfMemoryError` exception. Increase the memory or heap size allocated to the Administrator Daemon. To change the heap size, add the following configuration in the `admind.cnf` file located in the `<VDS Installation Directory>/admind/config` directory:
  `jvmOptions="-Xmx<MaximumHeapSize> -Xms<MinimumHeapSize>"`

  For example, to change the heap size to 4 GB, add `jvmOptions="-Xmx4G -Xms4G"` in the `admind.cnf` file.

The Administrator Daemon fails to start.

This error might occur if the JVM crashes and HPROF files are created in the `<VDS Installation Directory>/admind/` directory. Verify that you are using a valid license.

Troubleshooting the Administrator Tool

The Administrator tool fails to start.

This error can occur in the following situations:

- The `INFA_HOME` environment variable is set to a directory other than the correct installation directory. Clear the settings for the `INFA_HOME` environment variable and start the Administrator tool.
- On Linux, the web application is out of memory and the following exception occurs in the `<VDS installation directory>/logs/<node>/services/AdministratorConsole` directory:
  `java.io.IOException: Too many open files`

  Use the `increase ulimit -n` command to increase the limit of the number of files that are open.

The VDS Administrator tool in a PowerCenter setup goes down and the following exception occurs in the node.log file at <VDS installation directory>/logs/<node>:

```
Cannot update the data for the master gateway node [node_localhost] within the refresh interval time [<duration>]. The node will not continue as a master gateway node. Verify that the connection to the domain configuration repository database is valid.
```

This issue might occur if you set a low value for the MasterDBRefreshInterval parameter for the VDS Administrator tool.

The default value of the MasterDBRefreshInterval parameter is 8 seconds. Set the MasterDBRefreshInterval parameter to a higher value. For more information about setting up the MasterDBRefreshInterval parameter, see the Informatica Knowledge Base at: https://communities.informatica.com.

Troubleshooting Apache ZooKeeper

When I start the Apache ZooKeeper, the Apache ZooKeeper fails to start.

This error can occur if Apache ZooKeeper tries to use a port that is already in use. A BindException exception occurs in the `<VDS installation folder>/zookeeper/logs/zookeeper.out` file. Change the port or free the port, and start Apache ZooKeeper.

Troubleshooting Component Connectivity Issues

I get the following error message in the Data Flow Management view:

```
Administrator tool request to the Administrator Daemon timed out. Verify that the Administrator Daemon is up, verify the connectivity between the Administrator tool and Administrator Daemon.
```

This error might occur if the interface used for communication among the VDS components is configured incorrectly.

The interface might be configured incorrectly when you install the components in a multihome machine. To resolve this error, perform the following steps:

1. Verify that the `networkInterface` property is configured correctly in the configuration files of the Administration components, the VDS Node, and that the `clientPortBindAddress` property is configured correctly in the Apache ZooKeeper configuration file.
2. Restart all the components.
   On Windows, open a command prompt window and navigate to the `cli/bin` directory. Uninstall the Administrator Daemon, LBMRD, UMESTORED, and Apache ZooKeeper services, and then install them again.

   To uninstall the components, run the following command:

   ```
   admin_service.bat uninstall
   ```

   To install the components, run the following command:

   ```
   admin_service.bat install
   ```

I get the following error message in the Data Flow Management view:

```
Unable to communicate with Apache Zookeeper. Verify that Apache Zookeeper is up, and
verify the connectivity between the Administrator Daemon and Apache Zookeeper.
```

This error occurs in the following situations:

- Apache ZooKeeper is down.

  Navigate to the following directory:

  ```
  <ZooKeeper installation directory>\bin
  ```

  To verify the status, run the following command:

  ```
  ./zkServer.sh status
  ```

  If Apache ZooKeeper is not running, use the following command to start it:

  ```
  ./zkServer.sh start
  ```

- The Administrator Daemon is configured incorrectly.
  Navigate to the following directory:

  ```
  <VDS installation Directory>/admind/config/
  ```

  Verify that the `zkservers` field in the `admind.cnf` file is pointing to correct Apache ZooKeeper server URL.

I am unable to change the network interface.

This issue might occur after you connect to a new network.

Perform the following steps to set up the network interface after you connect to a new network:

1. Edit the `networkInterface` field in the `admind.cnf`, `lbmrd.xml`, and `umestored.cfg` files to replace the IP address with the IP address of the new connection.

   By default, the files are in the following directory:
   `<VDS Installation Directory>/admind/config`
2. On Windows, stop the following services:
   - `Informatica ADMIND Service`
   - `Informatica LBMRD Service`
   - `Informatica UMESTORED Service`
   - `Informatica VDS Nodes`

- Informatica Zookeeper Service

3. Stop the Informatica domain.
4. Update the networkInterface field in the following files to include the new IP address:
   - mnm.cnf located at <VDS Installation Directory>/mnm/config
   - node.cnf located at <VDS Installation Directory>/node/config
5. Update the clientPortBindAddress field in the zoo.cfg file to include the new IP address.
   By default, zoo.cfg is installed in the following directory:

   <VDS Installation Directory>/zookeeper/conf
6. Restart the VDS components.

When I start or stop a remote node, I get the following error message:

```
Passwordless ssh connection between the machines <host name> & <remote host name> is not
enabled.
Enable passwordless connection and try installation again.
```

This issue occurs if you have not configured passwordless SSH connections.

Before you run the remote installation, enable passwordless connection between the machines where you installed the administration components and the machines where you installed the VDS Nodes.

For more information about configuring passwordless SSH connections, see the Installation and Configuration Guide.

The Administrator tool shuts down unexpectedly after a period of inactivity and I get the following error in the tomcat log file:

```
2015-02-25 05:51:23,505 ERROR [Domain Monitor] [DOM_10095] The master gateway data in
the domain configuration repository was updated during the last refresh time interval
100001.
2015-02-25 05:51:31,392 FATAL [Domain Monitor] [DOM_10094] Cannot update the data for
the master gateway node [node01_localhost] within the refresh interval time [96000]. The
node will not continue as a master gateway node. Verify that the connection to the
domain configuration repository database is valid.
2015-02-25 05:51:31,413 INFO [Master Elect Data Writer] [DOM_10161] Stopped the writer
thread for the master gateway node election. Deleting the data from the domain
configuration repository.
2015-02-25 05:51:31,414 INFO [Master Elect Data Writer] [DOM_10146] Obtained the row
level lock on the ISP_MASTER_ELECT_LOCK table.
2015-02-25 05:51:31,415 INFO [Master Elect Data Writer] [DOM_10147] Deleting the row for
node [node01_localhost] in the domain configuration repository.
2015-02-25 05:51:31,416 INFO [Master Elect Data Writer] [DOM_10147] Deleting the row for
node [node01_localhost] in the domain configuration repository.
2015-02-25 05:51:31,417 INFO [Master Elect Data Writer] [DOM_10151] Released row level
lock on the ISP_MASTER_ELECT_LOCK table.
```

This error occurs if there is a connectivity issue.

To resolve this error, perform the following tasks:

1. In the Administrator tool, click the **Domain** tab.
2. In the **Domain Navigator** panel, select the **Domain**.
3. Click **Properties** tab.
4. Edit the Custom Properties.
5. In the dialog box that appears specify the name as MasterDBRefreshInterval and the value in seconds.
   The default value is 8 seconds.
   Informatica recommends that you specify a high value.

6. Click **OK**.
7. Restart the domain.

Troubleshooting Vibe Data Stream High Availability

I get the following exception in the VDS Node log file:

```
2013-12-31 15:27:10,524 ERROR [HDFS-41] com.informatica.um.binge.writer.TargetWrapper -
Exception in target
    org.apache.hadoop.hdfs.protocol.AlreadyBeingCreatedException: failed to create/
append/truncate file /<...>/LFdelimiter/hdfs_json.txt for
DFSClient_NONMAPREDUCE_1329143897_18 on client 10.75.40.149 because pendingCreates is
non-null but no leases found.
```

This error occurs when the new primary node does not have write access on the HDFS target. When a VDS Node goes down and another VDS Node comes up, the HDFS target does not release the lock while writing to the target service.

To resolve this error, make sure that the file names do not conflict if you configure high availability. Include the `infa.timestamp` variable in the file name in the **Destination** field of the HDFS target properties. For example: `hdfs://<HDFS hostname>:<port>/<foldername>/demo_#infa.timestamp.txt`

Troubleshooting Data Flows

The data flows do not appear correctly in the Administrator tool.

This error might occur if you have updated the Enterprise license to a Free license, and the VDS Node name is not the same as the host name. A license error occurs in the `<VDS installation folder>/node/logs/<node name>-node.log` file. When you start the VDS Node, use the host name as the VDS Node name.

The data flows are deployed, but the target services do not receive any data.

This error might occur if the network interfaces are not configured correctly. A `no interfaces matching criteria` exception occurs in the `<VDS installation folder>/node/logs/<node name>-node.log` file. Configure the correct interface in the `networkInterface` property in the `<VDS installation folder>/node/config/node.cnf` file.

The data flows are deployed, the source services and the target services are running but the target services do not receive any data.

This error might occur if the data flow uses **UM Secure** connection type in the data connection and the security related fields are not configured correctly. The VDS Node logs a security related error in the VDS Node log file. Specify correct values for the security related fields for the data connection and redeploy the data flows.

When I deploy a data flow, not all messages in the data flow are sent.

This error might occur if the machine on which the VDS Node is running is low on memory. An `OutOfMemoryError: Direct buffer memory error` occurs in the `<VDS installation folder>/node/logs/`

`<node name>-node.log` file. Increase the system memory or free up some memory by closing applications that are occupying more memory before you deploy the data flow.

In a deployed data flow, there is discrepancy in the bytes sent and bytes received.

This discrepancy in statistics might occur if you restart the source VDS Node associated with the data flow when the data flow is deployed. When you stop and start the VDS Node, the statistics are not reported for the interval during which the VDS Node is down.

The data flows are deployed but there is a discrepancy in the data that is sent and received.

This discrepancy might occur in case of database or ZooKeeper corruption. The data flows that you create and deploy using the corrupt data might be in inconsistent states or contain stale data. In this case, recover the data manually. As the data flows are not automatically refreshed, you must manually recover them.

To recover the states of the data flows, perform the following tasks:

1. Undeploy all data flows. To clear the ZooKeeper configuration or corrupted data from the data flows, perform a force clean when you undeploy all data flows.
2. Deploy all data flows.

I am using a Syslog UDS source in the data flow and unable to log in to the machine on which the node is installed.

This issue might occur if the Syslog UDS source has stopped publishing messages after the target service turns inactive.

To resolve this issue, undeploy the data flow and change the messaging mode from load balancing to streaming.

I have configured a WebSocket data connection to connect to an external load balancer. I see that the source service is not sending data to the load balancer and I get the following error message in the VDS Node log file for the source services:

```
java.lang.Thread.State: WAITING (on object monitor) at java.lang.Object.wait(Native Method)
```

This issue occurs if all the machines hosting the VDS Nodes for the target services are down.

To resolve this issue, perform the following steps:

1. If deployed, undeploy the data flow.
2. Start the VDS Nodes on the machines for the target services.
3. Deploy the data flow.

Troubleshooting Entities

I get the following error when I use a flat file source service to read from log files:

```
The process cannot access the file because it is being used by
another process.
```

This error might occur if you use Windows. The file locking mechanism of the Windows file system restricts access to a file by allowing access to only one user or process at a specific time.

To resolve this issue, perform the following steps:

1. Create a new file from the application every time it wants to create or rotate the log file.
2. Name the log file as follows:
   `<logfilename>.timestamp`
3. Configure VDS to accept the following pattern:
   `<logfilename>.*`
4. Schedule a task in Windows to delete old log files.

I use an MQTT source service and I get the following exception in the VDS Node log file:

```
Exception in thread "main" Persistence already in use (32200)
at
org.eclipse.paho.client.mqttv3.persist.MqttDefaultFilePersistence.open(MqttDefaultFilePer
sistence.java:112)
at org.eclipse.paho.client.mqttv3.MqttAsyncClient.<init>(MqttAsyncClient.java:286)
at org.eclipse.paho.client.mqttv3.MqttAsyncClient.<init>(MqttAsyncClient.java:167)
at org.eclipse.paho.client.mqttv3.MqttClient.<init>(MqttClient.java:224)
at org.eclipse.paho.client.mqttv3.MqttClient.<init>(MqttClient.java:136)

The process cannot access the file because it is being used by
another process.
```

This exception might occur when you run two MQTT sources on the same machine with the same client ID.

The MQTT source uses a persistence mechanism to store messages when they are being processed, and the client ID is used as the identifier for the persistence store. If two MQTT sources run on the same machine with the same client ID, this exception occurs when both the sources try to access the same persistence store. Configure a unique client ID for each connection.

I am writing to an HDFS target service and I see that the target file size and content are not increasing.

This behavior occurs because the HDFS target service writes to the target file only after the HDFS block size is reached.

A request from the HDFS target service to create a file, does not reach the master server in the HDFS cluster immediately. The target service initially caches the file data into a temporary local file. When the temporary file accumulates data that is more than one HDFS block size, the target service contacts the server. The server allocates a target file and communicates its identity to the target service. The target service then flushes the block of data from the temporary file to the target file.

The Syslog UDS source service fails to start and I get the following error message in the node log:

```
java.lang.IncompatibleClassChangeError: Found interface org.objectweb.asm.ClassVisitor,
but class was expected
```

This error occurs if you have configured the HADOOPBASEDIR environment variable. To resolve this error, ensure that you have not configured the HADOOPBASEDIR environment variable when you start the node on which the Syslog UDS source is deployed.

I use a Cassandra target service in the data flow to write to a Cassandra database and I get the following error in the Cassandra database log files:

```
following JSON field not present in "KEYSPACE.TABLENAME", will be dropped: [ "JSON-
FIELDNAME" ]
```

This error might occur in the following situations:

- You change the Cassandra database table by adding columns when the data flow is in a deployed state.

- The JSON input to the Cassandra target service has additional fields that were not in the Cassandra database table when you first deployed the data flow.

To resolve this error, undeploy and deploy the data flow.

I see duplicate messages in the target after I restart a VDS Node that contains the Static File source service.

This issue occurs if you specify the End-of-File (EOF) delimiter and do not specify the **Processed File Directory** property for the Static File source service. When you restart the VDS Node on which the VDS source service is running, data from the files which was already copied to the target is copied again and results in duplicate messages.

To resolve this issue, delete the files which were read by the Static File source service before you restart the VDS Node.

The client application is unable to send messages to the source service in a data flow.

This issue might occur when you use the following source services to read messages from client applications:

- TCP source service. To read messages from a TCP client application.
- Syslog TCP source service. To read messages from a Syslog TCP client application.

This issue occurs if the VDS Node on which the target service is running is down. As a result, the TCP window size between the client application and the source becomes zero and therefore the client application does not send any more data.

To resolve this issue, navigate to the following directory:

```
<VDS Node Installation Directory>/node/logs/
```

View the `<node name>-node.log` file in the target node and correct the exceptions to bring up the target service.

After the VDS Node on which the target service is running comes up, the TCP client application starts sending messages to the TCP source service and the TCP source service writes data to the target service from the point where it had stopped.

I have deleted a Vibe Data Stream Service and created a new one. When I start the Vibe Data Stream Service, I see the following exception in the Administrator Daemon log file:

```
Name is already used by an existing object
```

This issue might occur if you do not drop all the tables that were used by the previous Vibe Data Stream Service.

To resolve this issue, perform the following steps:

1. Delete the existing Vibe Data Stream Service.
2. Drop the database schema that was used by the previous Vibe Data Stream Service.
3. Ensure that the Administrator Daemon is up.
4. Create a Vibe Data Stream Service.
5. Select the appropriate database type when you specify the database properties for the new Vibe Data Stream Service.

I see an error condition for the Vibe Data Stream Service after I recycle the service.

The **Recycle** option is not supported on VDS. When you select the Vibe Data Stream Service in the Navigator after you recycle the service, the Vibe Data Stream Service icon in the Navigator displays a recycling state and the `Enabling Service` message appears in the contents panel. As a result, you are not able to access the **Vibe Data Stream** view and the **Properties** view in the contents panel.

Restart the Administrator tool. The **Services and Nodes** view of the **Domain** tab may still display an error condition for the Vibe Data Stream Service. This error does not impact the functionality of the Vibe Data Stream Service and you can ignore the message.

Alternatively, perform the following steps to delete the Vibe Data Stream Service and create a new service to resolve the error condition:

1. Restart the Administrator tool.
2. In the Navigator, right-click the Vibe Data Stream Service and click **Delete**.

   Note: When you click the Vibe Data Stream Service in the Navigator, the Vibe Data Stream Service icon in the Navigator displays a recycling status and the `Enabling Service` message appears in the contents panel. You can ignore this message.

   The **Recycle Service** window appears.
3. Click **OK**.
4. Restart the Administrator tool.
5. Create a new Vibe Data Stream Service.

I use an Unstructured Data Parser transformation in the data flow and I get the following exception in the VDS Node log file:

```
java.util.regex.PatternSyntaxException
```

This exception might occur when you have specified an incorrect pattern in the transformation.

Use the following debugger to verify if the pattern you have specified is correct:

https://grokdebug.herokuapp.com/

Troubleshooting Monitoring Tab Views

A VDS Node appears inactive on the Monitoring tab of the Administrator tool.

A VDS Node appears inactive in the following situations:

- The VDS Node might not be running. Verify the status of the VDS Node. If the VDS Node is not running, start it.
  If the VDS Node is running, navigate to the following directory:

  `<VDS Node Installation Directory>/node/logs/`

  View the `<node name>-node.log` file for exceptions and correct them.
- The Administrator Daemon might not be running. Verify the status of the Administrator Daemon. If the Administrator Daemon is not running, start it.
  If the Administrator Daemon is running, navigate to the following directory:

  `<Administrator Daemon Installation Directory>/admind/logs/`

Check the `admind.log` file for exceptions and correct them.

- The VDS Node is configured incorrectly.
  Navigate to the following directory:

  `<VDS Node Installation Directory>/node/config/`

  Verify that the `zkservers` field in the `node.cnf` file is pointing to the correct Apache ZooKeeper server URL.

A source service or a target service appears inactive on the Monitoring tab.

This can occur if the source service or a target service is configured incorrectly.

Navigate to the following directory:

`<VDS Node Installation Directory>/node/logs/`

View the `<node name>-node.log` file exceptions in source or target, and correct them.

The monitoring data indicates that the source statistics are increasing continuously, but the target statistics received are not increasing.

This error occurs in the following situations:

- The **Topic resolution type** field is set to `Unicast` in the data flow properties, and LBMRD is not running on the address specified in the **Resolver daemon** field. In the **Data Flows** pane, click the data flow whose properties you want to view. The **Entity Details** pane appears. Click **Connections** tab and click the connection. Verify that LBMRD is running on the address specified in the **Resolver daemon** field.
- The **Topic resolution** field is set to `Multicast` in the data flow properties, and the source service and target service are on different machines. Verify that the machines running the source service and target service have multicast enabled. Also verify that the machines are configured in the same multicast group.
- The transformation might not be applied correctly. Verify that you have configured the transformation correctly.

To determine whether multicast is enabled in your environment, perform the multicast capability test.

For more information about the multicast capability test, see the article *Informatica MTools* on MySupport at https://community.informatica.com/solutions/1470.

I use a Static File source service in my data flow and have specified the processed file directory. The files have been moved to the processed file directory, but the data is not received by the target service.

This situation occurs when the data is not sent to the target service because the files have been ignored. The Static File source service moves the files sent to the target service and the ignored files to the processed file directory. The Static File source service ignores the files in the following situations:

- The delimiter is EOF and the value of the **Maximum Event Size** property is less than the size of the file the source service is reading from.
- The files are older than the files already processed by the Static File source service.

The transformation statistics in the Grid view of the Administrator tool show a higher value in the events received column compared to the value in the events sent column.

This discrepancy might occur when the source service splits the events into multiple events based on the delimiter settings that you have specified for the source service. When the transformation receives the events from the source service, it also includes the split events in the events received statistic. As a result, the events received column of the transformation table displays a higher value than the value that appears in the events sent column of the source entity table.

You can ignore the difference in values as it has no impact on the functionality of the source service.

The monitoring data indicates that the Bytes Received, Events Received, Bytes Sent, and Events Sent statistics are displaying less than expected values.

This behavior occurs if you restart the VDS Node that the source service or target service is associated with. There is no workaround for this behaviour.

Related Topics:

- "Verifying the Vibe Data Stream Node Status" on page 131
- "Starting or Stopping the Vibe Data Stream Node on Linux" on page 131
- "Starting or Stopping the Vibe Data Stream Node on Windows" on page 131
- "Verifying the Administrator Daemon Status" on page 129
- "Starting or Stopping the Administrator Daemon on Linux" on page 130
- "Starting or Stopping the Administrator Daemon on Windows" on page 130

APPENDIX B

Frequently Asked Questions

This appendix includes the following topic:

Frequently Asked Questions About Vibe Data Stream

The following frequently asked questions can help you configure and use VDS more effectively.

Can I install Vibe Data Stream as a service when I have already installed Informatica Big Data Edition (BDE)?

No. We do not offer Vibe Data Stream as a service when you have installed Big Data Edition (BDE). If you install VDS on a machine where BDE is installed, you might find issues when you set up the INFA_HOME variable for VDS or BDE. Also, when you run VDS or BDE, there can be performance issues because of excessive load on the machine.

Can I use a regex filter with the asterisk (*) character in a flat file source to stream multiple files into a folder?

Yes, you can use a regex filter with the * character to stream multiple files into a folder. However, you might not be able to use a regex filter for recursive operations to stream files which are distributed across multiple folders or files placed in multiple sub-folders.

Can I recycle a Vibe Data Stream Service?

No. The **Recycle** option is not supported on VDS.

How do I change the database type assigned to a Vibe Data Stream Service?

You need to delete the existing Vibe Data Stream Service and create a new one with the database type that you want to use.

Delete the existing service. Wait for the Administrator Daemon to com up, and then create a Vibe Data Stream Service. Select the appropriate database type when you specify the database properties for the new Vibe Data Stream Service.

APPENDIX C

Regular Expressions

A regular expression describes a range or pattern of values that a filter condition can contain.

The following table describes the metacharacters that you can use in a regular expression:

Metacharacter	Description
.	Matches any single character.
[ ]	Indicates a character class. Matches any character inside the brackets. For example, [abc] matches “a,” “b,” and “c.”
^	If this metacharacter occurs at the start of a character class, it negates the character class. A negated character class matches any character except those inside the brackets. For example, [^abc] matches all characters except “a,” “b,” and “c.” If this metacharacter occurs at the beginning of the regular expression, it matches the beginning of the input. For example, ^[abc] matches the input that begins with “a,” “b,” or “c.”
-	Indicates a range of characters in a character class. For example, [0-9] matches any of the digits “0” through “9.”
?	Indicates that the preceding expression to this metacharacter is optional. It matches the preceding expression zero or one time. For example, [0-9][0-9]? matches “2” and “12.”
+	Indicates that the preceding expression matches one or more times. For example, [0-9]+ matches “1,” “13,” “666,” and similar combinations.
*	Indicates that the preceding expression matches zero or more times. For example, the input <abc*> matches <abc>, <abc123>, and similar combinations that contains <abc> as the preceding expression.
??, +?, *?	Modified versions of ?, +, and *. These match as little as possible, unlike the versions that match as much as possible. For example, the input “<abc><def>,” <.*?> matches “<abc>” and the input <.*> matches “<abc><def>.”
( )	Grouping operator. For example, (\d+,)*\d+ matches a list of numbers separated by commas such as “1” or “1,23,456.”
{ }	Indicates a match group.

Metacharacter	Description
\	An escape character, which interprets the next metacharacter literally. For example, [0-9]+ matches one or more digits, but [0-9]\+ matches a digit followed by a plus character. Also used for abbreviations such as \a for any alphanumeric character. If \ is followed by a number *n*, it matches the nth match group, starting from 0. For example, <{.*?}>.*?</\0> matches “<head>Contents</head>”. In C++ string literals, two backslashes must be used: “\\+,” “\\a,” “<{.*?}>.*?</\\0>.”
$	At the end of a regular expression, this character matches the end of the input. For example, [0-9]$ matches a digit at the end of the input.
|	Alternation operator that separates two expressions, one of which matches. For example, T|the matches “The” or “the.”
!	Negation operator. The expression following ! does not match the input. For example, a!b matches “a” not followed by “b.”

The following table describes the abbreviations that you can use in the regular expressions:

Abbreviation	Definition
\a	Any alphanumeric character, ([a-zA-Z0-9]).
\b	White space (blank), ([\\t]).
\c	Any alphabetic character, ([a-zA-Z]).
\d	Any decimal digit, ([0-9]).
\h	Any hexadecimal digit, ([0-9a-fA-F]).
\n	Newline, (\r|(\r?\n)).
\q	Quoted string, (\”[^\”]*\”)|(\‘[^\‘]*\‘).
\w	Simple word, ([a-zA-Z]+).
\z	Integer, ([0-9+]).

Related Topics:

- “Filters” on page 155

APPENDIX D

Command Line Program

This appendix includes the following topics:

Command Line Program Overview

infacmd is a command line program that allows you to administer domains, users, and services.

When you install Informatica PowerCenter, the infacmd command line program is installed by default. When you install VDS, the infacmd vds plugin is installed. The infacmd vds plugin supports the Vibe Data Stream Service.

Use infacmd vds to export and import data flows and node groups. You can run the infacmd command line program on UNIX and Windows.

infacmd vds Plugin

Each infacmd program has a plugin identifier. When you run the program, you include the plugin ID as part of the program name.

For the Vibe Data Stream, the plugin ID is vds.

Running Commands

Invoke infacmd from the command line. You can issue commands directly or from a script, batch file, or other program.

To run infacmd commands, perform the following steps:

1. At the command prompt, navigate to the directory where the infacmd executable is located. By default, infacmd is installed in the following directory:
   `<VDS InstallationDirectory>/isp/bin directory`
2. Enter infacmd on Windows or infacmd.sh on UNIX followed by the plugin ID, the command name, and the required options and arguments. The command names are not case sensitive.
   For example:

   infacmd(.sh) plugin_ID CommandName [-option1] argument_1 [-option2] argument_2...Command Options

When you run infacmd, you enter options for each command, followed by the required arguments. For example, most commands require that you enter the domain name, user name, and password using command options. Command options are preceded with a hyphen and are not case sensitive. Arguments follow the option. To enter an argument that contains a space or other non-alphanumeric character, enclose the argument in quotation marks.

Related Topics:

- "Export and Import Data Flows" on page 122
- "Export and Import Node Groups" on page 100

infacmd Return Codes

The infacmd program indicates the success or failure of a command with the following return codes:

- 0 indicates that the command succeeded.
- -1 indicates that the command failed.

Use the DOS or UNIX echo command immediately after running an infacmd command to see the return code for the command:

- In a DOS shell: `echo %ERRORLEVEL%`
- In a UNIX Bourne or Korn shell: `echo $?`
- In a UNIX C shell: `echo $status`

infacmd vds Command Reference

createService

Creates the Vibe Data Stream Service.

The infacmd vds CreateService command uses the following syntax:

```
createService
<-DomainName|-dn> domain_name
        <-NodeName|-nn> node_name
        <-UserName|-un> user_name
        <-Password|-pd> password
        <-ServiceName|-sn> service_name
        <-DbType|-dt> db_type (ORACLE, DB2, or SQLSERVER)
        <-DbUser|-du> db_user
        <-DbPassword|-dp> db_password
        <-DbUrl|-dl> db_url
        [<-DbDriver|-dr> db_driver]
        [<-DbSchema|-ds> db_schema (used for SQL Server only)]
        [<-DbTablespace|-db> db_tablespace (used for DB2 only)]
        [<-SecureJDBCParameters|-sjdbcp> secure_jdbc_parameters]
        <-licenseName|-lsn> license_name
```

The following table describes infacmd vds CreateService options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. Name of the Informatica domain. You can set the domain name with the -dn option. If you do not specify a domain, the command creates a domain named Domain_VDS by default.
NodeName -nn	node_name	Required. Node where you want to run the Vibe Data Stream Service to run.
-UserName -un	user_name	Required if the domain uses Native or LDAP authentication. User name to connect to the domain. You can set the user name with the -un option. Optional if the domain uses Kerberos authentication. To run the command with single sign-on, do not set the user name. If you set the user name, the command runs without single sign-on.
-Password -pd	password	Required if you specify the user name. Password for the user name. The password is case sensitive. You can set a password with the -pd option.
-ServiceName -sn	service_name	Required. Vibe Data Stream Service name.
-DbType -dt	db_type	Required. Values are Oracle, SQL Server, or DB2.
-DbUser -du	db_user	Required. Account for the database. Set up this account using the database client.
-DbPassword -dp	db_password	Required. Repository database password for the database user.

Option	Argument	Description
-DbUrl -dl	db_url	Required. The connection string used to connect to the database. Use one the following connection string: - IBM DB2. `jdbc:informatica:db2://<host name>:<port>;DatabaseName=<database name>` - Microsoft SQL Server. `Server: jdbc:informatica:sqlserver://<host name>:<port>;DatabaseName=<database name>` - Kerberos-enabled Microsoft SQL Server. `jdbc:informatica:sqlserver://<host name>:<port>;DatabaseName=qadb;integratedSecurity=true;authenticationScheme=JavaKerberos;encrypt=false` - Oracle. `jdbc:informatica:oracle://<host name>:<port>;SID=<database name>` - Sybase. `jdbc:informatica:sybase://<host name>:<port>;DatabaseName=<database name>`
-DbDriver -dr	db_driver	Optional. The Data Direct driver to connect to the database. For example: com.informatica.jdbc.oracle.OracleDriver
-DbSchema -ds	db_schema	Optional. The schema name for a Microsoft SQL Server database.
-DbTablespace -dt	db_tablespace	Required for a DB2 database only. When you configure a tablespace name, the Vibe Data Stream Service creates all repository tables in the same tablespace. You cannot use spaces in the tablespace name.
-SecureJDBCParameters -sjdbcp	secure_jdbc_parameters	Secure JDBC parameters that you want to append to the database connection URL.
-licenseName -lsn	license_name	License assigned to the service.

Related Topics:

- "Creating the Vibe Data Stream Service using Informatica Command Line Program" on page 39

exportDataFlow

Exports a list of data flows.

The command uses the following syntax:

```
exportDataFlow
<-DomainName|-dn> domain name
[<-UserName|-un> user name]
[<-Password|-pd> password]
<-FilePath|-fp> File path where you want to export the data flows to.
[<-Dataflows|-df> Comma separated values of data flows. Ignore if you want to export all
data flows.]
```

The following table describes infacmd exportDataFlow options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. Name of the Informatica domain. You can set the domain name with the -dn option. If you do not specify a domain, the command creates a domain named Domain_VDS by default.
-UserName -un	user_name	Required if the domain uses Native or LDAP authentication. User name to connect to the domain. You can set the user name with the -un option. Optional if the domain uses Kerberos authentication. To run the command with single sign-on, do not set the user name. If you set the user name, the command runs without single sign-on.
-Password -pd	password	Required if you specify the user name. Password for the user name. The password is case sensitive. You can set a password with the -pd option.
-FilePath -fp	file path	Required. File path where you want to export the data flows to. You can set the file path with the -fp option.
-Dataflows -df	comma separated values of dataflows	Optional. Comma separated values of data flows. Ignore if you want to export all data flows. You can specify the list of data flows with the -df option.

Related Topics:

- "Exporting Data Flows" on page 122

importDataFlow

Imports a list of data flows.

The command uses the following syntax:

```
importDataFlow
<-DomainName|-dn> domain name
[<-UserName|-un> user name]
[<-Password|-pd> password]
<-FilePath|-fp> File path where you want to import data flows from.
[<-Overwrite|-ow> Overwrite existing data flows. Default is false.
```

The following table describes infacmd importDataFlow options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. Name of the Informatica domain. You can set the domain name with the -dn option. If you do not specify a domain, the command creates a domain named Domain_VDS by default.
-UserName -un	user_name	Required if the domain uses Native or LDAP authentication. User name to connect to the domain. You can set the user name with the -un option. Optional if the domain uses Kerberos authentication. To run the command with single sign-on, do not set the user name. If you set the user name, the command runs without single sign-on.
-Password -pd	password	Required if you specify the user name. Password for the user name. The password is case sensitive. You can set a password with the -pd option.
-FilePath -fp	file path	Required. File path where you want to import the data flows from. You can set the file path with the -fp option.
-Overwrite -ow	true | false	Optional. Overwrites existing data flows. Overwrite the data flows in one of the following situations: - You want to overwrite a data flow that already exists. - You want to import a data flow that contains an entity that is associated with a node group that does not exist. Default is false.

Related Topics:

- "Importing Data Flows" on page 123

exportNodeGroup

Exports node groups.

The command uses the following syntax:

```
exportNodeGroup
        <-DomainName|-dn> Domain name
  [<-UserName|-un> User Name]
  [<-Password|-pd> Password]
  <-FilePath|-fp> File path where you want to export the node groups to.
  [<-Nodes|-nd> Export nodes associated with node group. Default is false.]
```

The following table describes infacmd exportNodeGroup options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. Name of the Informatica domain. You can set the domain name with the -dn option. If you do not specify a domain, the command creates a domain named Domain_VDS by default.
-UserName -un	user_name	Required if the domain uses Native or LDAP authentication. User name to connect to the domain. You can set the user name with the -un option. Optional if the domain uses Kerberos authentication. To run the command with single sign-on, do not set the user name. If you set the user name, the command runs without single sign-on.
-Password -pd	password	Required if you specify the user name. Password for the user name. The password is case sensitive. You can set a password with the -pd option.
-FilePath -fp	file path	Required. File path where you want to export the node groups to. You can set the file path with the -fp option.
-Nodes -nd	true | false	Optional. Export nodes associated with the node group. Default is false.

Related Topics:

- "Exporting Node Groups" on page 101

importNodeGroup

Imports node groups.

The command uses the following syntax:

```
importNodeGroup
        <-DomainName|-dn> Domain name
  <-UserName|-un> User name]
  [<-Password|-pd> Password]
  <-FilePath|-fp> File path where you want to import node groups from.
  [<-Overwrite|-ow> Overwrite existing node groups. Default is false.]
```

The following table describes infacmd importNodeGroup options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. Name of the Informatica domain. You can set the domain name with the -dn option. If you do not specify a domain, the command creates a domain named Domain_VDS by default.
-UserName -un	user_name	Required if the domain uses Native or LDAP authentication. User name to connect to the domain. You can set the user name with the -un option. Optional if the domain uses Kerberos authentication. To run the command with single sign-on, do not set the user name. If you set the user name, the command runs without single sign-on.
-Password -pd	password	Required if you specify the user name. Password for the user name. The password is case sensitive. You can set a password with the -pd option.
-FilePath -fp	file path	Required. File path where you want to import the node groups from. You can set the file path with the -fp option.
-Overwrite -ow	true | false	Optional. Overwrites existing node groups. You can use this option to import a node group that already exists. Default is false.

Related Topics:

- "Importing Node Groups" on page 101

APPENDIX E

Configuring Vibe Data Stream to Work With a ZooKeeper Observer

You can configure VDS to work with a ZooKeeper ensemble that uses a ZooKeeper observer. A ZooKeeper observer is a ZooKeeper node in an ensemble that does not participate in the leader election process in a high-availability environment.

For information about Apache ZooKeeper observers, see the ZooKeeper documentation in the following location:

http://zookeeper.apache.org/doc/trunk/zookeeperObservers.html

To configure VDS to work with a ZooKeeper observer, perform the following tasks:

1. Edit the configuration file of the ZooKeeper node that you choose as observer. Add the following configuration:
   `peerType=observer`
2. Edit the configuration files of all ZooKeeper servers in the ensemble. Add `:observer` to the server definition line of each observer.
   For example: `server.1:localhost:2181:3181:observer`
3. Edit the configuration files of the Administrator Daemon and the VDS Nodes in the deployment. Configure the `zkservers` property to point to the ZooKeeper observer.

APPENDIX F

Glossary

Administrator Daemon

Daemon process that facilitates the creation, management, deployment, and undeployment of data flows through the Administrator tool. The Administrator Daemon also aggregates statistics and state information from Vibe Data Stream Nodes in the deployment, and send the information to the Administrator tool.

data flow

Defines the path of data from source services to target services through zero or more transformations. You can create, design, deploy, and undeploy data flows in the Administrator tool. Data flows can be simple data flows, such as one-to-one, or complex data flows, such as, one-to-many, many-to-one, and many-to-many.

Informatica Administrator

Informatica Administrator (Administrator tool) is a web application that you can use to manage, monitor, deploy, and undeploy data flows.

receiver type ID

A 32-bit value that uniquely identifies the Ultra Messaging receiver.

source service

A VDS Node contains one or more specialized threads that work together to transfer data from an application host to a target data store or data engine. Source services are threads that consume and publish events generated by a source application. Source services publish data in the form of Ultra Messaging or WebSocket messages.

target service

Target services are the threads that subscribe to data published by source services and write the data to the target. Target services run on hosts that have access to the target.

topic resolution domain

The domain of UDP multicast or unicast connectivity that allows Ultra Messaging topic resolution to occur. Topic resolution enables receivers to discover sources.

unicast topic resolution daemon (LBMRD)

A daemon process that performs the same topic resolution activities as those performed by multicast topic resolution. By default, Ultra Messaging expects multicast connectivity between sources and targets. When only unicast connectivity is available, you must run one or more unicast topic resolution daemons (LBMRD).

VDS Node

A VDS Node is a Java program within which source services and target services run. You can run multiple source services and target services on a single node. You can also configure multiple nodes to run on a host machine.

INDEX

E

F

G

H

I

J

S

T

U

V

W

Z