Informatica

Informatica® Data Integration Hub
10.4.0

High Availability Guide

Informatica Data Integration Hub High Availability Guide
10.4.0
February 2020

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

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

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

Publication Date: 2020-05-21

Table of Contents

Preface

Use the Data Integration Hub High Availability Guide to learn how to configure a high availability cluster on which you can install and run Data Integration Hub. The guide also includes troubleshooting instructions, a description of log messages, and a port number reference.

Informatica Resources

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

Informatica Network

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

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

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

Informatica Knowledge Base

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

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

Informatica Documentation

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

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

Informatica Product Availability Matrices

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

Informatica Velocity

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

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

Informatica Marketplace

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

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

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

CHAPTER 1

Introduction to High Availability

This chapter includes the following topics:

High Availability Overview

When you use Data Integration Hub to process publications and subscriptions on a large scale, you need to provide continuous service even if one component fails. You set up a high availability environment with failover mechanisms that can handle a failed server or service.

You can configure a high availability cluster with multiple machines in a network or with a single multi-processor machine. A high availability cluster includes multiple installations of typical Data Integration Hub components that communicate with each other in the background while providing a single point of operation for users.

Cluster Configuration Types

You determine which cluster type to use based on the organization requirements and available hardware resources. Most of the configuration steps are identical for both cluster types.

You can set up and configure the following high availability cluster types:

- Multi-machine cluster. Consists of several computers that contain identical installations of Data Integration Hub. Each computer in the cluster can take over in case the other computer fails.
- Single-machine cluster. Consists of multiple Data Integration Hub installations on a single multi-processor computer. When you configure a single-machine cluster, you modify port numbers in each subsequent instance to prevent conflicts.

Cluster Components

The high availability cluster consists of multiple instances of Data Integration Hub and related component. You use a similar configuration for multi-machine and single-machine clusters.

The cluster normally consists of the following main parts:

- Server cluster. Handles document and event processing between Data Integration Hub and PowerCenter.
- Operation Console cluster. Handles browser requests from Operation Console users with a load balancer.

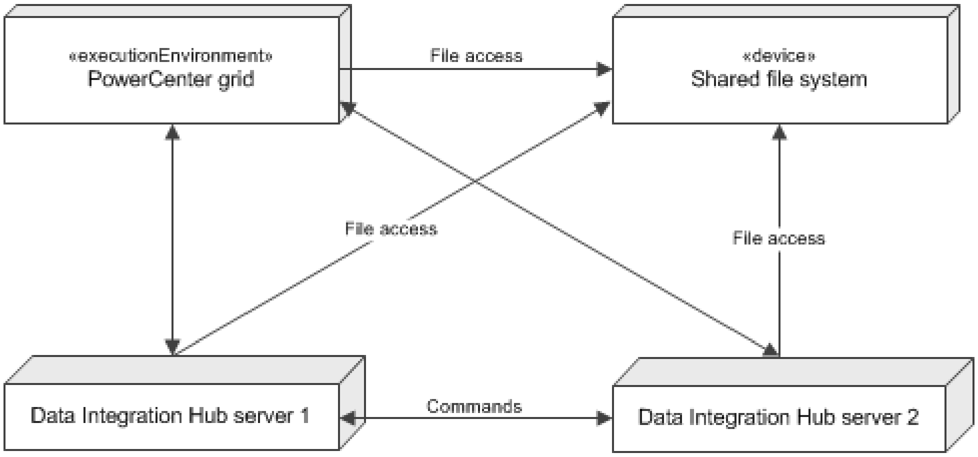
Data Integration Hub Server Cluster Components

The Data Integration Hub server cluster handles document and event processing between the Data Integration Hub server, JMS Broker, Managed File Transfer, and PowerCenter.

The Data Integration Hub server cluster typically consists of the following components:

- Data Integration Hub server. Processes events and documents.
- Shared file system. Contains the document store, File Receive and File Send endpoint documents, archive service, and the JMS messages data directory.
- PowerCenter grid. Processes documents from the Data Integration Hub server with workflows.

The following image shows an example of an active/active Data Integration Hub server cluster configuration:



This cluster consists of two Data Integration Hub server instances, a PowerCenter environment, and a shared file system. The Data Integration Hub server instances also include the optional Managed File Transfer component. The communication lines between the Data Integration Hub servers and the PowerCenter grid represent the RMI and Web Service message flows.

This cluster configuration does not require you to install the Data Integration Hub server on each of the PowerCenter nodes. The installation location has no impact on the necessary configuration changes.

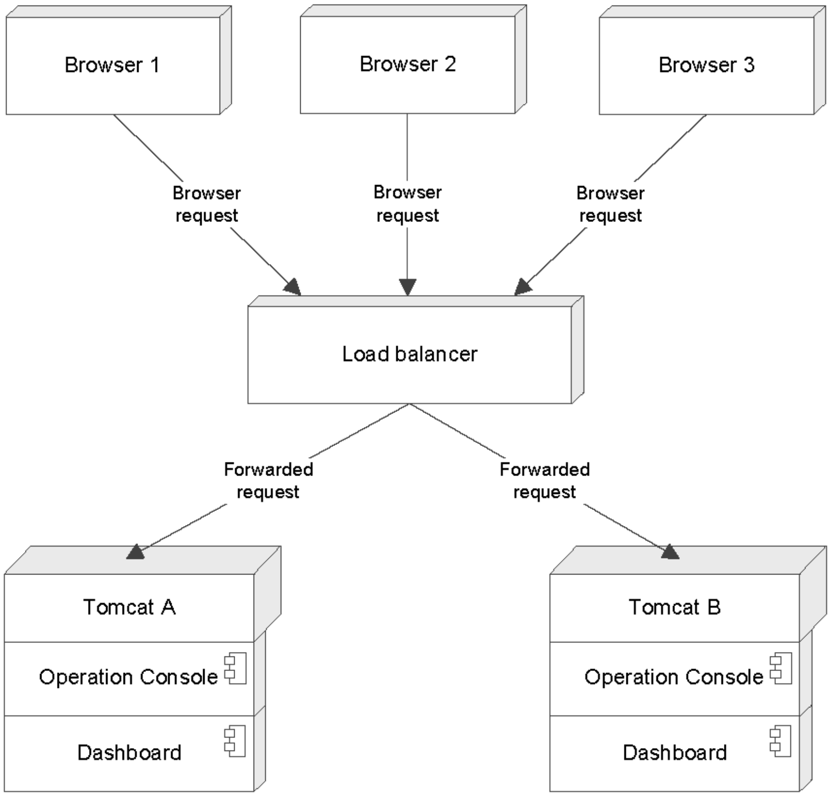
Operation Console Cluster Components

The Operation Console cluster handles browser requests to and from Operation Console users. The cluster uses a load balancer to distribute the requests between multiple Tomcat instances.

The Operation Console cluster typically consists of the following components:

- Browser clients. Data Integration Hub users log in to the browsers and perform actions in the Operation Console.
- Load balancer. Receives requests from each Operation Console browser and forwards the requests to the available Tomcat server instance.
- Tomcat Operation Console server. Receives and processes forwarded requests from the load balancer.

The following figure shows an example of the Operation Console cluster configuration:



This cluster consists of three browser clients, one load balancer, and two Tomcat instances that run the Operation Console and the Dashboard. The communication lines between the components represent actions that users perform in the Operation Console that the load balancer distributes between the Tomcat instances. Synchronize the clocks of all machines in the cluster to within 30 seconds of each other to ensure proper handling of browser requests.

Note: The Dashboard is available if you installed the Dashboard and Reports component.

Configuration Checklist

Follow the high-level steps to configure the high availability environment. The actual steps may vary according to the cluster type and the Data Integration Hub components that you use.

To configure the high availability environment, complete the following tasks:

1. Complete the prerequisites tasks and set up the high availability environment.
2. Configure RMI connections for the Data Integration Hub server and the Power Center Integration Service.
3. Configure the Operation Console. For single-machine clusters, modify Tomcat ports for each subsequent Operation Console instance.
4. Configure an HTTP load balancer to distribute data transfer between the Operation Console browsers and the Tomcat server.
5. Configure the Dashboard and reports system properties and Logi Info settings. This task is required only if you installed the Dashboard and Reports component.

CHAPTER 2

Set Up the High Availability Environment

This chapter includes the following topics:

Prerequisites

Before you set up the high availability environment, verify that your system meets the following prerequisites:

- It is recommended that the Data Integration Hub repository is on a high availability database, such as Oracle Real Application Clusters.
- The Data Integration Hub repository and the document store are on high availability storage systems, such as Network-Attached Storage (NAS).
- The high availability environment includes a clustered file system, such as Global File System (GFS) or Veritas Cluster File System) (VxCFS).
- The document store is on a shared file system that is accessible from all of the cluster components using the same directory path.
- The cluster consists of multiple active Data Integration Hub server nodes.
- The PowerCenter real-time workflows are active and running.
- A load balancer is installed, such as the F5 Load Traffic Manager or the Apache HTTP Server.
- The clocks on all Data Integration Hub nodes are synchronized to within 30 seconds of each other. Use the Network Time Protocol (NTP) to synchronize the clocks.
- The ping latency between the nodes in the cluster is 10 ms or smaller.

- All nodes in the cluster use the same operating system platform. Clusters with nodes that use both UNIX and Microsoft Windows operating systems are not supported.

Set Up a High Availability Storage Solution

When you set up the high availability environment, you determine which storage solution to use for the Data Integration Hub repository database and the document store.

A high availability storage system typically consists of a group of reliable and fast hard drives on which you install the database or file system. To increase reliability, use Network-Attached Storage (NAS) or Storage Area Network (SAN) storage systems.

In the high availability storage system, you can mount any high-performance hard drives, such as RAID, SSD, or SCSI. To optimize performance, use RAID 1+0 hard drives.

Set Up a Clustered File System

Install and configure a clustered file system on which to store shared components, such as the document store.

A clustered file system is a distributable file system that utilizes the high availability physical storage configuration. The clustered file system you install depends on the type of high availability storage solution.

When you use an SAN storage solution, you need to install the clustered file system separately. For example, you can use the Veritas Cluster File System (VxCFS) file system or the Global File System (GFS). The recommended file system is VxCFS.

Configure the clustered file system to use hardware-based I/O fencing for all of the nodes in the cluster.

Configure Cluster Management Software

Use cluster management software to monitor the status of the services that run on each node in the cluster and troubleshoot service failure.

You configure the software to start or stop services on each node in the cluster in case one or more services fail. For example, if you use an active/passive cluster, you can configure the cluster manager to stop all of the services in the active node and start all of the services in the passive node in case one of the services in the active node fails.

A cluster manager typically consists of the following components:

- Main application that manages the entire cluster and sends commands to start, stop, or verify the availability of the services. You can install the application anywhere in the cluster. To improve reliability, install the application outside of the cluster.
- Agent that monitors and reports the status of the services in each node to the main application. The agent can start or stop the services with commands from the main application. You install the agent on each node in the cluster and define which services to monitor.

Install Data Integration Hub

You install Data Integration Hub either on a multi-machine cluster or a single-machine cluster. Choose the type of installation based on the cluster type that you set up.

Ensure that the Data Integration Hub components that you install are identical between all the nodes in the cluster.

Install Data Integration Hub in a Multi-Machine Cluster

In a multi-machine cluster, you install Data Integration Hub on each computer in the cluster in the same way that you install a single Data Integration Hub instance.

Install the same Data Integration Hub components on each node in the cluster.

Install Data Integration Hub on a Single-Machine Cluster

After you install the first Data Integration Hub instance on a machine with multiple processors, you can install up to four additional instances to create a single-machine cluster. Perform additional steps to configure the subsequent instances within the cluster. You can install up to five Data Integration Hub instances on a single computer.

By default, the Data Integration Hub installer assigns port numbers beginning with 18xxx. In subsequent installations, you can use port numbers beginning with 28xxx for the second instance, 38xxx for the third instance, and so on. For a list of port numbers in Data Integration Hub, see Appendix C, “Port Number Reference” on page 33.

1. Install the first copy of Data Integration Hub as described in the *Data Integration Hub Installation and Configuration Guide*.
2. Stop all of the Data Integration Hub processes.
3. Run the Data Integration Hub installer with the following changes:
    a. In the **Installation Location** screen, select a different installation directory for each subsequent installation.

    b. In the **Database Connection** screen, choose to use an existing Data Integration Hub repository and enter the same database connection properties as the first installation.

    c. In the **Web Server** screen, modify the default HTTP or HTTPS connector port number and the server shutdown listener port number.

    .
4. After you install all of the Data Integration Hub instances, modify the port numbers in each of the cluster components to prevent conflicts.

Configure Database Connections for Oracle

If you use an Oracle database, define the maximum number of database connections for all of the Data Integration Hub server nodes in the cluster.

In all copies of the dx-configuration.properties files, set the maximum number of database connections in the dx.jdbc.maxPoolSize property. For example, to define a maximum of 50 database connections, use the following syntax: `dx.jdbc.maxPoolSize=50`

For general information about configuring the JDBC URL, see the *Progress DataDirect Connect for JDBC User's Guide*.

Note: Data Integration Hub does not support Oracle TNS files.

Define System Properties for PowerCenter Workflows

You modify system properties in Data Integration Hub to enable running workflows from multiple nodes in the cluster to PowerCenter.

For more information about Data Integration Hub system properties, see the *Data Integration Hub Administrator Guide*.

Modify the values of the following properties:

- pwc.domain.gateway. If you use multiple Informatica gateway nodes, add all of the nodes to the property.
- pwc.webservices.url. If you use batch workflows, configure the load balancer such that batch web service requests go to the same node and if the node fails, service requests must go to other node.

Configure a Shared File System

You configure Data Integration Hub components on a shared file system to improve performance and reliability when Data Integration Hub processes documents and events. Each system component can use a different shared file system as long as every server can access it using the same file path.

Install and configure the document store on a shared file system.

Configure the Document Store

Install the document store on a shared network drive so that it is accessible by all the Data Integration Hub Server instances.

The Informatica domain nodes that run workflows need to use the same file references to access the shared file system. All Data Integration Hub and Informatica domain nodes require access to the same file server and all nodes must use the same path to the file server. For example, the path `\\shared\storage_1\DataIntegrationHub\document_store\file_one.txt` must point to the same file from all nodes.

If you upgrade a single instance of Data Integration Hub Server to a cluster environment and the document store is not accessible by all the nodes in the cluster, you must move the document store to a shared location.

In the repoutil command line utility, run the moveDocumentStore command with the following syntax in a single line:

```
repoutil -c moveDocumentStore -t dih -l <Data Integration Hub repository jdbc URL> -u
<user name> -p <password> --docStore <new document store location>
```

The following example shows a repoutil script for moving the document store in a node that uses a UNIX operating system:

```
./repoutil.sh -c moveDocumentStore -t dih
-l "jdbc:informatica:oracle://xsvcshacl03:1521;ServiceName=drep02_taf" -u dihadmin -p
mypassword --docStore="/u02/app/infa_shared/DIH_doc_store"
```

For more information about the repository utility, see the *Data Integration Hub Administrator Guide*.

Note: Do not move the document store manually. If you manually move the document store, Data Integration Hub will not reference document attachments for events correctly.

CHAPTER 3

Configure JMS Discovery

This chapter includes the following topics:

JMS Discovery Overview

When you set up the high availability environment, you can configure the cluster to send and receive JMS messages. The JMS messages consist of documents and events that Data Integration Hub and PowerCenter process.

The following JMS discovery modes are available:

- Unicast. Static discovery mode in which all of the machines in the cluster receive all of the incoming documents. Choose unicast mode if your network environment does not support multicast communication. Unicast discovery mode is strongly recommended.
- Multicast. Dynamic discovery mode in which a single machine receives and routes all of the documents to all of the machines in the cluster. Choose multicast mode if you want to utilize a single point of entry to the cluster and do not want to manage the IP address of each node in the cluster separately.

To use multicast mode, you perform the multicast capability test to determine whether your network environment supports multicast communication.

Multicast Capability Test

The multicast capability test determines whether multicast communication is enabled in your network environment.

1. Download the **mtools** package from the following location:
   https://community.informatica.com/solutions/1470
2. Extract the package to each computer that you want to test.

3. Navigate to the directory `mtools/<OS Version>/`. The supported operating systems are Windows, Linux (x86), AIX-5-powerpc64, SunOS-5.10-i386, SunOS-5.10-sparc.
4. If you use default multicast address (224.252.253.254) and its default multicast port (18162), run the following commands:
    - On the first machine: `mdump 224.252.253.254 18162`.
    - On all other machines: `msend -b1 -m20 -n5 224.252.253.254 18162`

If the test is successful, the output displays the following information:

- The output in the first machine indicates that five messages, numbered 0 to 4, were received. Each message should be 20 bytes long.
- The output in the second machine indicates that five messages were sent. Each messages should be 20 bytes long.

Data Integration Hub Server Configuration Properties

You configure JMS discovery for each Data Integration Hub server node in the cluster based on the mode that you want to use.

To configure JMS discovery, you modify configuration properties in all copies of the dx-configuration.properties file. The Operation Console file is typically located in the following directory: `<DIHInstallationDir>\conf\`. The Data Integration Hub server file is located in the following directory: `<DIHInstallationDir>\DataIntegrationHub\tomcat\shared\classes`.

The following table describes the JMS discovery configuration properties:

Property	Description
dx.console.jms.unicastAddress	IP address and port number of the Operation Console to use in unicast mode. Default value: `0.0.0.0:18050` For single-machine clusters, each Data Integration Hub server instance must have a different port number.
dx.jms.multicastAddress	IP address and port number of the Operation Console to use in multicast mode. Default value: `224.252.253.254:18000` For single-machine clusters, each Data Integration Hub server instance must have a different port number.
dx.cluster.name	Logical name of the Data Integration Hub server cluster. Must be identical for all of the nodes in the cluster.

Configuring Multicast JMS Discovery

Modify properties in different cluster components to enable multicast JMS discovery in the high availability cluster.

Make sure you configure the same JMS discovery mode in every file. You cannot configure some properties for unicast mode and other properties for multicast mode.

1. In all copies of the Data Integration Hub server configuration files, add comment indicators to the dx.console.jms.unicastAddress property. This property is relevant only for unicast mode.
2. If you want to process documents by reference, modify the value of the dataDirectory property to point to a shared storage directory.

Configuring Unicast JMS Discovery

Modify properties in different cluster components to enable unicast JMS discovery in the high availability cluster.

Make sure you configure the same JMS discovery mode in every file. You cannot configure some properties for unicast mode and other properties for multicast mode.

1. In all copies of the Data Integration Hub server configuration files, add the following properties and replace the syntax example with the actual values:
   - dx.AMQ.discovery=b2bDxAMQBrokerStatic
   - dx.AMQ.static.discovery.address=static:(tcp://<host1Name>:<OperationConsoleJMSPort>,tcp://<host1Name>:<DIHServerJMSPort>, tcp://<host2Name>:<OperationConsoleJMSPort>,<host2Name>:<DIHServerJMSPort>)

   For example: `dx.AMQ.static.discovery.address=static:(tcp://host1:18100,tcp://host1:18050, tcp://host2:18100,tcp://host2:18050)`
2. Add comment indicators to the following properties that are relevant only for multicast mode:
   - dx.cluster.name
   - dx.jms.multicastAddress

CHAPTER 4

Configure RMI Connections

This chapter includes the following topics:

RMI Connections Overview

The Data Integration Hub server and PowerCenter Integration Service use Remote Method Invocation (RMI) connections to communicate.

You configure the RMI connections to enable the PowerCenter Integration Service to communicate with all of the Data Integration Hub server nodes in the high availability cluster.

Configuring RMI Connections

Configure RMI connections to enable the PowerCenter Integration Service and all of the Data Integration Hub nodes in the high availability cluster to communicate.

1. In the Informatica Administrator tool, open the **Processes** tab of the PowerCenter Integration Service.
2. In the environment variable list, enter the IP addresses and ports for each of the Data Integration Hub nodes in the `DX_SERVER_URL` environment variable in the following format:

   ```
   rmi://<host1>:<port>[;rmi://<host2>:<port>]
   ```

   For example: `rmi://dih1:18095;rmi://dih2:18095`

   Make sure that the port number matches the value in the dx.rmi.port configuration property.
3. Configure any other node-dependent PowerCenter Integration Service properties, such as the Java system properties, to use IP addresses and ports of the Data Integration Hub server cluster nodes.
4. For single-machine clusters, enter a unique value in the dx.rmi.port property of the `dx-configuration.properties` file for each subsequent Data Integration Hub installations. The default value is `18095`.

   The file is typically located in the following locations:

   - `<DIHInstallationDir>/conf/dx-configuration.properties`
   - `<DIHInstallationDir>/DataIntegrationHub/tomcat/shared/classes/dx-configuration.properties`

CHAPTER 5

Configure the Operation Console

This chapter includes the following topics:

Configuring a Proxy Server for the Operation Console

You configure a proxy server for the Data Integration Hub Operation Console to enable load balancing and failover handling.

Load-balancing software for the Operation Console acts as a proxy server for the browsers connecting to the Operation Console. The browsers see a single proxy server and are not aware to which physical server they are connected. Configure the load balancer for sticky sessions.

If you choose to use Apache HTTP Server for load balancing, use version 2.2.15 or higher.

Configuring Tomcat for Single-Machine Clusters

For single-machine clusters, you modify ports in the Tomcat configuration file for each subsequent Data Integration Hub instance in the cluster. You do not need to modify ports for the first instance.

You edit the HTTP or HTTPS ports to prevent the Data Integration Hub server or the Operation Console from creating an incorrect URL for viewing the contents of event blobs and for advanced exception handling.

The `server.xml` file is typically located in the following directory: `<DIHInstallationDir>/DataIntegrationHub/tomcat/conf/`

1. In the `server.xml` file, modify the values of the following properties:

Property	Description
Shutdown port	Port to use when you shut down Tomcat with an external process. This property contains the attribute `port` and listens only to the local loopback address `127.0.0.1`. Default is: `18005`
HTTP connector port	Port to use when Tomcat connects to the Operation Console. This property contains the attribute `port`. Default is: `18080`
HTTPS connector port	Port to use when Tomcat connects to the Operation Console with a secure protocol. This property contains the attribute `port`. Default is: `18443`

2. In the Operation Console, modify the dx.console.url system properties according to the HTTP or HTTPS port that you defined.
3. For Windows operating systems, modify the following shortcuts and bookmarks:
   - Remove the Start menu entry: **Start** > **All Programs** > **Informatica** > **Data Integration Hub**.
   - Configure the Data Integration Hub Windows services for manual startup.
   - Create a shortcut to the folder `<DIHInstallationDir>\bin\dihstartup.bat` for each node in the cluster. This shortcut starts the node.
   - Create a shortcut to the folder `<DIHInstallationDir>\bin\dihshutdown.bat` for each node in the cluster. This shortcut shuts the node down.

CHAPTER 6

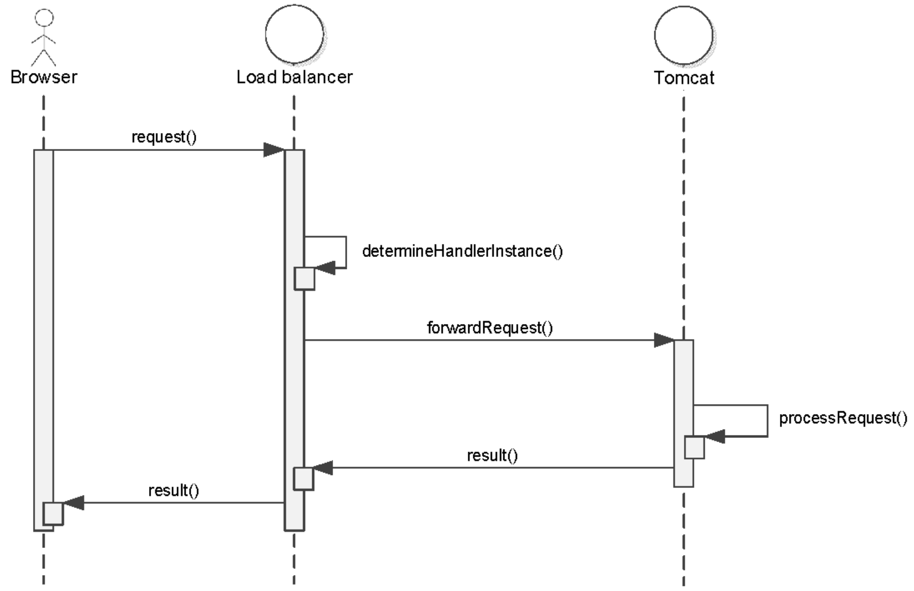
Configure an HTTP Load Balancer

This chapter includes the following topics:

HTTP Load Balancer Overview

You use a load balancer in the high availability cluster to distribute the workload evenly over two or more servers and to achieve optimal network performance.

The following figure shows how the load balancer forwards browser requests to one of the Tomcat instances:



In this diagram, the browser sends the request to the load balancer. The load balancer determines which Tomcat instance will handle the request and forwards the request to the Tomcat instance. The Tomcat instance processes the request and sends the result back to the load balancer. The load balancer forwards the result back to the browser.

The load balancer determines which Tomcat computer handles each request. This computer might vary from request to request and depends on the configuration.

You can use any load balancing software or hardware. This chapter includes examples for configuring load balancing with a newly installed copy of Apache HTTP Server.

Configuring Load Balancer Properties

Configure Data Integration Hub system properties for the load balancer.

1. In the Data Integration Hub Operation Console, from the Navigator, open **Administration** > **System Properties**.
2. Change the value of the `dx.console.url` property to the load balancer URL, in the following format: `http://<load_balancer>:<load_balancer_port>/dih-console` .

   For example: `http://host1:80/dih-console`

Configuring Sticky Sessions

A sticky session is a session that uses a single server to handle all user traffic for a specific browser. In a sticky session, after the browser creates a session, the same server that received the first session continues to process all requests from that session until the user logs out of the browser.

When you enable sticky sessions, the load balancer forwards all requests from a specific browser to the same server until the user ends the session, regardless of the load on the servers in the cluster. For example, if browser 1 starts a session and the load balancer forwards the initial request to Tomcat A, all subsequent requests are sent to Tomcat A.

Sticky sessions do not require multicast transmission to multiple destinations and do not generate additional network traffic or reduce performance. However, if a server fails, all existing sessions on that server end without a failover mechanism and all users must manually log in to a different server.

Configuring Tomcat for Sticky Sessions

Configure properties in the file `<DIHInstallationDir/DataIntegrationHub/tomcat/conf/server.xml`. Each item that needs to be changed is marked with an "INFA change cluster" XML comment in `server.xml`. Search for this string to find the required locations.

1. Create a backup of the file `server.xml` and call it `server.xml.bak`.
2. Enable AJP connector by removing the comment indicator from the line `<connection port="18009" enableLookup="false" redirectPort="18443" protocol="AJP/1.3"/>`.
   This opens a socket that the load balancer uses to forward requests.

3. Change the value of the attribute `jvmRoute` on the element `Engine` to the physical computer name, for example, `Tomcat-A`. The value of the `jvmRoute` attribute must be unique for each Tomcat instance, therefore, use the computer name. Otherwise, the load balancer cannot consistently route the requests and sessions to the correct machine.

   Note: Failure to edit `jvmRoute` might cause unexpected results.

Sticky Session Configuration Example

To configure configuring load balancing with Apache HTTP Server, follow the sample instructions.

Configuring Apache HTTP Server for Sticky Sessions

To configure the Apache HTTP Server for sticky sessions, you need to make changes in the cluster configuration.

1. Include cluster configuration. Edit the file `<apache installation>/conf/httpd.conf` and add `Include conf/dx-gui-cluster.conf` at the end.
   This Apache directive indicates that an additional configuration file must to be loaded.
2. Add the cluster configuration. Create a file called `dx-gui-cluster.conf` under `<apache installation>/conf/` and copy the following code section into it:

```
## Load the proxy module, if is not loaded already
<IfModule !proxy_module>
  LoadModule proxy_module modules/mod_proxy.so
</IfModule>

## Load the http proxy protocol module, if it is not loaded already
## This module enables load balancing between nodes
<IfModule !proxy_http_module>
  LoadModule proxy_http_module modules/mod_proxy_http.so
</IfModule>

## Load the proxy balancer module, if it is not loaded already
## This module enables load balancing between nodes
<IfModule !proxy_balancer_module>
  LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
</IfModule>

## Load the ajp proxy protocol module, if it is not loaded already
## This module enables the communication between Apache and Tomcat via AJP
<IfModule !proxy_ajp_module>
  LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
</IfModule>

## Add an additional location to access the current load balancing configuration
## and statistics. Access is only allowed from localhost
## Only set the location if status module (mod_status) is loaded
<IfModule status_module>
  <Location /balancer-manager>
    SetHandler balancer-manager

    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1
  </Location>
</IfModule>

## Configure the ‘proxy’ for the Data Integration Hub Operation Console instances
## Note that this is a static list, all the machines should be explicitly listed here
## The value for the attribute ‘route’ must exactly match that which is
## used when configuring Tomcat
<Proxy balancer://dx-gui-cluster>
  BalancerMember ajp://tomcat-a:18009 route=dxTomcat-A
  BalancerMember ajp://tomcat-b:18009 route=dxTomcat-B
</Proxy>

# Disable ProxyRequest as only the ProxyPass and ProxyPassReverse are used.
# This prevents the server from processing requests from spammers (sending e-mails)
ProxyRequests Off

## Configure the forwarding of the request.
## Client browser must connect to <apache http server>:<port>/dih-console/
## Note that it must be a single line
ProxyPass /dih-console/ balancer://dx-gui-cluster/dih-console/
stickysession=JSESSIONID|jsessionid

## Configure the return route from back-end to front
ProxyPassReverse /dih-console/ balancer://dx-gui-cluster/dih-console/
```

Note: The URLs in the `ProxyPass` and `ProxyPassReverse` directives must end with a slash. If the slash is missing, routing a message to and from the back-end nodes might fail.

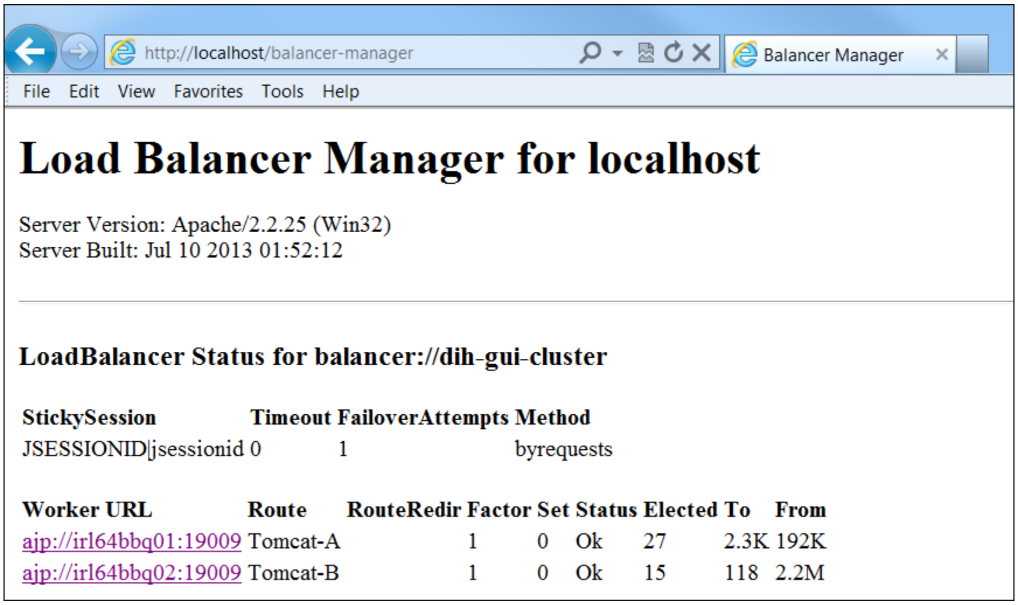
3. Edit the cluster configuration. In the file that you created, edit the copied code segment as follows:
   - Delete the two lines beginning with `BalancerMember`. You can find them between the lines `<Proxy balancer://dx-gui-cluster>` and `</Proxy>`.
   - Insert lines, one for each Operation Console (Tomcat) server that is part of your cluster. Each line must have fields for a specific node (Tomcat instance), the machine name, and the `jvmRoute` value set for that specific Tomcat instance, for example, `Balancer-Member ajp://<operation console machine name>:18009 route=<jvmRoute value as defined in web.xml>`
4. Enable load balancer status monitoring. In the file `<apache installation>/conf/httpd.conf`, remove the comment indicator before the following line: `LoadModule status_module modules/mod_status.so` and restart the Apache HTTP Server.

Testing the Sticky Sessions Configuration in Apache HTTP Server

To test the sticky sessions configuration changes, start the Tomcat instances and the Apache HTTP server and follow the test procedure in this section.

1. Test the back-end machines. Go directly to the Data Integration Hub Operation Console on the back-end machines. Verify that the behavior in a clustered environment is the same as in a single-server environment. Verify that the login and logout are successful.
   Repeat this step for each back-end machine.
2. Verify that the Balancer Manager works properly. Go to `http://host:80/balancer-manager`. This page shows the status of all the configured workers. After restarting Apache, it should show that all the workers are OK and that no data has been transmitted or received.
   You can only perform this step if you enabled monitoring in the Apache HTTP Server configuration file.
3. Log in to the Data Integration Hub Operation Console 1. Go to `http://host:80/dih-console/`. The Data Integration Hub Operation Console login page should be displayed. Log in into the Data Integration Hub Operation Console.
4. Verify the sticky session. Go back to the balancer manager page at `http://host:80/balancer-manager`. This page should show that all the session requests have been forwarded to a single back-end machine.
5. Log in to the Data Integration Hub Operation Console 2. Go to `http://host:80/dih-console/`, using a different browser than before, and log in to the Data Integration Hub Operation Console.

6. Verify sticky session 2. Go back to the balancer manager page at `http://localhost:80/balancer-manager`. This page should show that all the session requests have been forwarded to the other back-end machine.
The following image shows the expected results in the Balancer Manager:



Load Balancer Manager for localhost

Server Version: Apache/2.2.25 (Win32)
Server Built: Jul 10 2013 01:52:12

LoadBalancer Status for balancer://dih-gui-cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONID|jsessionid	0	1	byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	To	From
ajp://irl64bbq01:19009	Tomcat-A		1	0	Ok	27	2.3K	192K
ajp://irl64bbq02:19009	Tomcat-B		1	0	Ok	15	118	2.2M

CHAPTER 7

Configure the Dashboard and Reports

This chapter includes the following topics:

Dashboard and Reports Configuration Overview

To enable high availability for the Dashboard and the operational data store, you configure the security settings of the LogiXML security tokens and modify settings in Data Integration Hub and in the `LogiXML_Settings.lgx` file to use the load balancer instead of the regular connection strings.

The Dashboard must be able to access all nodes of the load balancer.

Configuring the Dashboard and Reports to Use the Load Balancer

If you installed the Dashboard and Reports component, change settings in Data Integration Hub and in Logi Info Studio to use the load balancer for the Dashboard in Data Integration Hub.

Before you configure the Dashboard, configure sticky sessions for the Operation Console and verify that the dx.console.url system property contains the load balancer URL.

1. In the Data Integration Hub Operation Console, from the Navigator, open **Administration** > **System Properties**.
2. In the `dx.dashboard.url` system property, enter the load balancer URL in the following format: `http://<load_balancer>:<load_balancer_port>/dih-dashboard`

   For example:

   ```
   http://host1:80/dih-dashboard
   ```
3. Open the following file:

   ```
   <DIHInstallationDir>/ DataIntegrationHub/tomcat/shared/classes/
   dx_dashboard_configuration.xml
   ```

4. In the `dx_dashboard_configuration.xml` file, configure the following properties:

Property	Description
DX_CONSOLE_URL	Load balancer URL in the following format: `http://<load_balancer>:<load_balancer_port>/dih-console` For example: `http://host1:80/dih-console`
DASHBOARD_SAVEFOLDER	Shared folder for the saved Dashboard applications. You must create the folder before you configure this property. For example: `//NetworkDir/SavedDashboards`
AuthenticationClientAddresses	Load balancer IP addresses separated by commas. Enter IPv4 and IPv6 addresses. For example: `10.40.0.135,192.178.147.1,192.268.30.1,127.4.0.1,193.168.147.1`
LogonFailPage	Load balancer URL in the following format: `http://<load balancer>:<dih_port>/dih-console/logout.jsp` For example: `http://host1:80/dih-console/logout.jsp`

5. In the `_Settings.lgx` file, add a `SecureKeySharedFolder` property to the `Security` element and set the value of `SecureKeySharedFolder` to the path of a shared location on the network to store the SecureKey file.

   For example: `SecureKeySharedFolder="//NetworkDir/SecureKeys"`

   Note: You must create the folder before you add this property.

6. Configure the operational data store (ODS) workflow file with the standard PowerCenter high availability configuration. The name of the workflow file depends on type of database on which the ODS is installed.

Database Type	Workflow Location and Name
Oracle	`<DIHInstallationDir>/powercenter/ETL/DX_ETL.xml`
Microsoft SQL Server	`<DIHInstallationDir>/powercenter/ETL/DX_ETL_SQLSERVER.xml`

7. Restart the Tomcat server.
8. If you customized the Dashboard, deploy the customization on all of the machines in the cluster.

APPENDIX A

Troubleshooting High Availability

This appendix includes the following topic:

Troubleshooting High Availability

This section contains solutions to common problems that you might encounter when you configure a high availability cluster.

After installing and configuring the Data Integration Hub server, the service does not start.

In the `startup.bat` or `startup.sh` file, change the value of the shutdown port to `28095`. The file is located in the following directory: `<DIHInstallationDir>\DataIntegrationHub\bin`. If the cluster is installed on multiple hosts, change the value of the shutdown port on all the cluster hosts.

One of the services in the cluster failed.

Restart all of the Data Integration Hub services on the node where the service failed.

When one of the nodes in the cluster fails while publications are running, the publications remain in Processing status indefinitely.

Perform the following actions:

1. In the **Event List** page of the Data Integration Hub Operation Console, discard the publication events that are in Processing status.
2. Run the publications again.

Perform the following actions:

1. In the **Event List** page of the Data Integration Hub Operation Console, discard the publication events that are in Processing status.
2. Run the publications again.

When one of the nodes in the cluster fails while subscriptions are running, the subscriptions remain in Processing status indefinitely.

Perform the following actions:

1. For each subscription that is in Processing status, inspect the target to verify that some or all of the subscription data does not exist on the target. If the data exists, delete the data.

2. In the **Event List** page of the Data Integration Hub Operation Console, reprocess the subscription events that are in Processing status.

APPENDIX B

High Availability Log Messages

This appendix includes the following topic:

High Availability Log Messages

The dxserver.log file of each node in the network contains certain messages indicating that dx-configuration.properties file is configured properly.

If dx-configuration.properties under Data Integration Hub server is configured properly the following messages (order may vary a bit) should be seen in dxserver.log of each node. In this example there are two hosts called host1 and host2 and the ports are Data Integration Hub default ports.

```
INFO  [org.apache.activemq.network.DiscoveryNetworkConnector]
{Notifier-MulticastDiscoveryAgent-listener:
DiscoveryNetworkConnector:localhost:BrokerService
[b2bDxInternalCommandBroker]}
Establishing network connection from vm://b2bDxInternalCommandBroker
to tcp://host1:18100

INFO  [org.apache.activemq.network.DiscoveryNetworkConnector]
{Notifier-MulticastDiscoveryAgent-listener:
DiscoveryNetworkConnector:localhost:
BrokerService[b2bDxInternalCommandBroker]} Establishing network
connection from vm://b2bDxInternalCommandBroker
to tcp://host2:18050

INFO  [org.apache.activemq.network.DemandForwardingBridge]
{StartLocalBridge: localBroker=vm://b2bDxInternalCommandBroker:0#0}
Network connection between vm://b2bDxInternalCommandBroker:0#0 and
tcp://localhost/127.0.0.1:18100(b2bDxInternalCommandBrokerGUI) has
been established.

INFO  [org.apache.activemq.network.DiscoveryNetworkConnector]
{Notifier-MulticastDiscoveryAgent-listener:DiscoveryNetworkConnector:
localhost:BrokerService[b2bDxInternalCommandBroker]}
Establishing network connection from vm://b2bDxInternalCommandBroker
to tcp://host2:18100

INFO  [org.apache.activemq.network.DemandForwardingBridge]
{StartLocalBridge: localBroker=vm://b2bDxInternalCommandBroker:0#2}
Network connection between vm://b2bDxInternalCommandBroker:0#2 and
tcp://host2/host2IP:18050(b2bDxInternalCommandBroker) has been
established.

INFO  [org.apache.activemq.network.DemandForwardingBridge]
{StartLocalBridge: localBroker=vm://b2bDxInternalCommandBroker:0#4}
Network connection between vm://b2bDxInternalCommandBroker:0#4 and
```

```
tcp://host2/host2IP:18100(b2bDxInternalCommandBrokerGUI) has been
established.
```

APPENDIX C

Port Number Reference

This appendix includes the following topic:

Port Number Reference

The Data Integration Hub server and the Operation Console use default port numbers. You can modify the port numbers to prevent conflicts in single-machine clusters.

Note: The 0.0.0.0 IP address indicates that all addresses assigned to the server should be used.

The following table lists the default Operation Console port numbers:

Listening IP	Port	Cluster Type	Description
127.0.0.1	18005	- Multi-machine - Single-machine	Tomcat shutdown port. This port cannot be disabled.
0.0.0.0	18009	- Multi-machine	Tomcat AJP/1.3 port. This port is disabled by default.
0.0.0.0	18080	- Multi-machine - Single-machine	HTTP listener port.
0.0.0.0	18443	- Multi-machine - Single-machine	HTTPS listener port. This port is disabled by default.

The following table lists the default Data Integration Hub server port numbers:

Listening IP	Port	Cluster Type	Description
0.0.0.0	18095	- Multi-machine - Single-machine	(Bootstrap) RMI Registry port.

APPENDIX D

Glossary

active/active

A configuration where all nodes are active. In the event of a failure, the remaining active nodes assume responsibility for all processing tasks.

active/passive

A configuration with an active node and one or more passive nodes. The passive nodes are used only if the active node fails. Only one node is active at a time.

active node

The server in a Data Integration Hub cluster that is currently processing user transactions. If the active node fails unexpectedly, a passive node takes over.

AJP

Apache Jservlet Protocol is a binary packet-oriented communication protocol used to reduce the communications overhead between a web server (Apache HTTP Server) and a servlet container (Apache Tomcat).

browser

The Web interface that the Operation Console user uses to log on to the Operation Console. The HTTP load balancer forwards he actions that the Operation Console users perform are to the available Tomcat instance in the Operation Console cluster.

client

See **browser**.

failover

The migration of a service, process, or task to another node when the original node becomes unavailable, for example, if the original node shuts down unexpectedly.

grid

An alias assigned to a group of nodes that run sessions and workflows. When you run a workflow on a grid, you improve scalability and performance by distributing session and command tasks to service processes running on nodes on the grid.

load balancer

Hardware or software that balances the incoming requests to different back-end machines or nodes.

recovery

The automatic or manual completion of tasks after an application service is interrupted.

resilience

The ability of services to tolerate transient failures, such as loss of connectivity to the database or network failures.

sticky session

A session that uses a single server to handle all user traffic for a specific browser.

Tomcat instance

The node that runs Tomcat with the Data Integration Hub Operation Console.

INDEX