ELEKTRON ZERO DAEMON VERSION 1.3 FOR ELEKTRON CONNECT

INSTALLATION GUIDE
# Contents

## 1 Overview

1.1 Product Description ............................................................................................................................................. 1
1.2 New Features in this Release ............................................................................................................................. 1
1.3 Supported Platforms ........................................................................................................................................... 1
1.4 Supported Features ........................................................................................................................................... 2
1.5 Audience ......................................................................................................................................................... 2
1.6 Defaults and Conventions ................................................................................................................................. 2
1.7 Related Documentation ....................................................................................................................................... 3
1.8 Documentation Feedback ................................................................................................................................. 3
1.9 Glossary of Terms and Acronyms ....................................................................................................................... 3
1.10 Included Utilities ............................................................................................................................................... 4

## 2 Installation

2.1 Install Package ................................................................................................................................................ 5
2.2 OpenSSL Installation ........................................................................................................................................... 5
  2.2.1 OpenSSL Vulnerabilities .............................................................................................................................. 5
  2.2.2 Linux Vulnerabilities .................................................................................................................................... 5
  2.2.3 OpenSSL Setup on Linux .......................................................................................................................... 6
  2.2.4 Downloading and Building Source ............................................................................................................ 6
  2.2.5 Windows WinInet ....................................................................................................................................... 6
2.3 Linux Installation ............................................................................................................................................... 7
  2.3.1 Tuning Linux TCP buffers .......................................................................................................................... 7
  2.3.2 Running Linux Install Script ....................................................................................................................... 7
2.4 Windows Installation ......................................................................................................................................... 8
  2.4.1 Configure Outgoing Internet Connections ................................................................................................ 8
  2.4.2 Tuning Windows TCP Buffers .................................................................................................................... 8
  2.4.3 Installing EZD for Windows ....................................................................................................................... 8
  2.4.4 Verify Security Protocols .......................................................................................................................... 14
2.5 Configuration ................................................................................................................................................. 15
  2.5.1 Advanced Configuration ............................................................................................................................. 15
  2.5.2 Customizing the EZD Device Channel Settings ........................................................................................ 16
  2.5.3 Customizing the RSSL Server .................................................................................................................... 17
  2.5.4 Customizing Internet Connections ........................................................................................................... 18
  2.5.5 Customizing Delivery Direct Connections ............................................................................................... 20
  2.5.6 Host Endpoint Configuration .................................................................................................................... 21
  2.5.7 User configuration file .............................................................................................................................. 21
2.6 Initialization .................................................................................................................................................... 22
  2.6.1 OpenSSL Library Loading .......................................................................................................................... 22
  2.6.2 Starting the EZD in Linux .......................................................................................................................... 22
  2.6.3 Starting the EZD in Windows .................................................................................................................... 22

## 3 Functionality

3.1 EZD Persistent Controller Channel ...................................................................................................................... 23
  3.1.1 Device Channel Failure and Reconnection .................................................................................................. 23
  3.1.2 Endpoint Host Change ............................................................................................................................... 23
  3.1.3 Password Change ...................................................................................................................................... 23
3.2 Connection Handling ......................................................................................................................................... 24
3.3 Ping Handling .................................................................................................................................................. 24
3.4 Message Handling .......................................................................................................................................... 24
3.5 Proxy support .................................................................................................................................................. 24
3.6 Required files .................................................................................................................................................. 24
1 Overview

1.1 Product Description

The Elektron Zero Daemon (EZD) provides network connectivity for existing Reuters Wire Format (RWF) and Market Feed (MF) applications using a near zero footprint datafeed delivery method. Client applications connect to the EZD to route traffic over the internet or via Delivery Direct, which uses dedicated, leased telecommunication lines. EZD uses the RSSL messaging protocol via Transport Layer Security (TLS) over TCP/IP. This allows for almost the entire existing real-estate of RSSL-consuming applications such as TREP and RWF clients to obtain data through a low footprint delivery method with no application code changes other than a few, minor configuration changes.

The Elektron Zero Daemon software allows clients to get connectivity to the Elektron Connect and Elektron Test Environment products via the internet or Delivery Direct.

1.2 New Features in this Release

EZD 1.3 provides the following new features:

- EZD now publishes the ELEKTRON_DD service instead of IDN_RDF. Datafeed client applications that require the IDN_RDF service need to tune the EZD to publish the IDN_RDF service instead. Refer to Section 2.5.4.
- EZD datafeed clients connect to the same endpoint as the EZD controller channel. EZD failback only operates on the EZD controller channel.
- Integration with Windows to operate as a service.
- Integration with Refinitiv AAA permissioning. This replaces the previous ILA values used per datafeed and allows each datafeed subscriber to use AAA token authentication. This modification requires EZD to have a permanent controller channel.
- EZD no longer uses a license file. Previously, a license purchase was required which limited the number of available connections on a per license basis.
- The Actual Experience client application for Linux is discontinued for EZD 1.3.
- EZD now requires TLS1.2 secure connections to connect to HMDS endpoints. Linux machines may need to upgrade their OpenSSL libraries in order to make connections.

1.3 Supported Platforms

- Linux RHEL 6.X 64 bit series
- Oracle Linux 6.X 64 bit series
- Oracle Linux 7.X 64 bit series
- Windows Server 2008 (and above) 64 bit

**NOTE:** 32 bit architectures are not supported at this time.
1.4 Supported Features

EZD supports the following key features:

- Encrypts and decrypts data using the Transports Layer Security protocol version 1.2 only. This replaces the Secure Sockets Layer (SSL) protocol because of known security issues.
- Supports non-authenticating HTTP internet proxy connections; you can configure the application to connect to the final endpoint via an HTTP proxy (for internet connections).
- Enables zlib compression (default) for upstream RSSL/RWF connections.
- Supports the failback feature, which allows the controller channel connected to a providing source to determine if the channel is connected to the primary host of the providing service (for Elektron Connect only).

1.5 Audience

This manual is aimed at an audience of TREP and Elektron infrastructure experts familiar with networks that host Refinitiv server software including, but not limited to, the Advanced Data Hub (ADH), Advanced Distribution Server (ADS), and RTIC. It is presumed that readers have access to and are familiar with the ADS and ADH Software Installation Manuals.

1.6 Defaults and Conventions

Throughout this manual, in tables that describe the command options available for use with a tool, the default value is listed in parenthesis. Additionally, this manual uses the following stylistic conventions:

- Path names and file names within the text of this document appear in a bold font. For example: `rmds.cnf`
- Commands to be entered by the user (exactly as shown) appear in black, Lucida Console font. For example: `export RMDS_CONFIG`
- Variables, i.e. information that must be supplied by the user, are indicated by an italic font. For example: `tar -xvf tarfile_containing_load`
- Service names, names of configuration parameters, and configuration values appear in a bold font.
- Command line syntax, command examples, file listings, code samples, or system messages appear in a plain typewriter font with gray shading; e.g.:

```c
typedef struct
{
    unsigned char flags;
    unsigned char code;
} EXAMPLE_STRUCT;
```
1.7 Related Documentation

Refer to these documents in determining how to install, set up, and configure your TREP or Elektron system:

- Advanced Distribution Server (ADS) Software Installation Manual specific to the version that you use
- Advanced Data Hub Server (ADH) Software Installation Manual specific to the version that you use

1.8 Documentation Feedback

While we make every effort to ensure the documentation is accurate and up-to-date, if you notice any errors, or would like to see more details on a particular topic, you have the following options:

- Send us your comments via email at ProductDocumentation@refinitiv.com.
- Mark up the PDF using the Comment feature in Adobe Reader. After adding your comments, you can submit the entire PDF to Refinitiv by clicking Send File in the File menu. Use the ProductDocumentation@refinitiv.com address.

1.9 Glossary of Terms and Acronyms

<table>
<thead>
<tr>
<th>TERM OR ACRONYM</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Key</td>
<td>A combination of the Stream ID and Stream ID Password.</td>
</tr>
<tr>
<td>Controller</td>
<td>An alias term for EZD that is used in account management.</td>
</tr>
<tr>
<td>DD</td>
<td>(Refinitiv) Delivery Direct</td>
</tr>
<tr>
<td>Datafeed</td>
<td>An alias term for clients that connect through EZD to consume data.</td>
</tr>
<tr>
<td>EC</td>
<td>Elektron Connect</td>
</tr>
<tr>
<td>ETE</td>
<td>Elektron Test Environment</td>
</tr>
<tr>
<td>EZD</td>
<td>Elektron Zero Daemon proxy application</td>
</tr>
<tr>
<td>Elektron Real Time ID</td>
<td>The user name credentials to authenticate the EZD device.</td>
</tr>
<tr>
<td>Elektron Real Time ID Password</td>
<td>The user password credentials to authenticate the EZD device.</td>
</tr>
<tr>
<td>Failback</td>
<td>A feature to move the currently connected channel on a secondary host to the primary host.</td>
</tr>
<tr>
<td>Failover</td>
<td>A feature to successively attempt socket connections to hosts in the host list.</td>
</tr>
<tr>
<td>RSSL</td>
<td>Reuters Sink Source Layer. A TCP/IP based protocol used by TREP. Used for user authentication.</td>
</tr>
<tr>
<td>RWF</td>
<td>Reuters Wire Feed. RWF is the format used on the wire and is used with the RSSL protocol.</td>
</tr>
<tr>
<td>SSL/SSLED</td>
<td>A Refinitiv term for Sink Source Layer. A TCP/IP based protocol used by TREP.</td>
</tr>
<tr>
<td>SSL</td>
<td>A widely recognized term for Secure Sockets Layer. An Encryption algorithm used over Internet connections.</td>
</tr>
<tr>
<td>Stream ID</td>
<td>The user name credentials for AAA login.</td>
</tr>
<tr>
<td>Stream ID Password</td>
<td>The user password credentials for AAA login.</td>
</tr>
<tr>
<td>TREP</td>
<td>Thomson Reuters Enterprise Platform</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security. An encryption algorithm used over Internet connections. Starting with EZD1.3, EZD connects to endpoints via TLS1.2 protocols.</td>
</tr>
</tbody>
</table>

Table 1: Glossary of Acronyms and Terms
1.10 Included Utilities

The `rmdstestclient` utility is included as part of the EZD package to verify EZD operation. For Linux, the `LD_LIBRARY_PATH` variable must be set to include the `libxerces` libraries that are included with the utility.

For more information on the operation of `rmdstestclient`, download the Infra Tools package and consult the appropriate documentation.
2 Installation

2.1 Install Package

Download the EZD package from the Refinitiv GSG product repository. After downloading the EZD package, unzip its contents to the following directory:

- Linux: /opt/thomsonreuters/SOFTWARE
- Windows: downloads directory

2.2 OpenSSL Installation

To use EZD on Linux, you must also have the latest version of the OpenSSL¹ tool kit release 1.0.1. EZD uses this tool kit to establish Transport Layer Security (TLS) for encryption and decryption of all content on its connections. To use EZD on Windows platforms, use TLS provided through the OS and Internet Explorer.

2.2.1 OpenSSL Vulnerabilities

Refinitiv is aware of several known vulnerabilities in the SSL protocol distributed through OpenSSL packages. Because of these vulnerabilities, Refinitiv is no longer using SSLv3 and instead uses TLS 1.2 for encrypted connections. When using OpenSSL on your machine, verify that it is using a version that does not contain these vulnerabilities. Refer to the following technical bulletins for more information:

- 7234: Refinitiv Trade Notification - SSLv3 Disablement due to Internet vulnerability, also known as POODLE (https://customers.reuters.com/a/support/NotificationService/ViewProduct.aspx?id=R0/7234)

2.2.2 Linux Vulnerabilities

Refinitiv is aware of the ‘GHOST bug in the __nss_hostname_digits_dots() function of glibc that can be triggered (locally or remotely) via the gethostbyname() functions used to resolve hostnames. Refer to the following Refinitiv Technical Bulletin available on the MyRefinitiv site for more information: https://my.refinitiv.com/content/dam/myrefinitiv/products/9936/en/Technical/TREP_TechBulletin_GHOST.pdf

¹. https://www.openssl.org/
### 2.2.3 OpenSSL Setup on Linux

Verify you are using the most secure version of OpenSSL. Contact your Linux vendor to see whether any security alerts are active and to determine the latest available OpenSSL package that they support.

- For Oracle Linux, go to http://public-yum.oracle.com and then browse the repository to locate the latest packages for the specific install version.

You can use the `yum` command to install, update, or query version information for OpenSSL. To determine whether OpenSSL is already installed on a Red Hat or Oracle Linux machine, use the `yum search openssl` command. This shows any installed OpenSSL packages.

- If the OpenSSL package is not installed, to install the package run: `yum install package_name` (for packages and their names, refer to the Red Hat or Oracle resources listed above).
- If OpenSSL is installed, to determine whether updates are available for the OpenSSL library, run: `yum check-update | grep openssl`. The command `yum update package_name` updates the specified package.

### 2.2.4 Downloading and Building Source

As an alternative approach, download the full source code and compile it for a particular platform. Source code packages are available at http://www.openssl.org/source.

After downloading, follow the included instructions to configure and build for your desired platform.

### 2.2.5 Windows WinInet

EZD on Windows does not use OpenSSL, but instead uses encryption through the Windows WinInet API. No OpenSSL libraries are required to be downloaded for Windows.
2.3 Linux Installation

Verify that you have completed the steps to perform OpenSSL installation according to your operating system.

Refer to Section 2.5, Configuration to install EZD.

2.3.1 Tuning Linux TCP buffers


Edit the `/etc/sysctl.conf` file to contain the following settings:

```plaintext
net.core.rmem_max = 8388608
net.core.wmem_max = 8388608
net.core.rmem_default = 4194304
net.core.wmem_default = 4194304
net.ipv4.tcp_rmem = 4096 4194304 8388608
net.ipv4.tcp_wmem = 4096 4194304 8388608
net.ipv4.tcp_mem = 4096 4194304 8388608
```

To make this change permanent, reboot or execute `/sbin/sysctl -p` to reread the updated values.

2.3.2 Running Linux Install Script

An install script is included to automate the installation process for Linux machines. Execute the `ezdSetup.sh` script to create the `ezd.cnf` and user configuration files. The user is prompted for non-default parameters, such as the proxy configuration.

Refer to Appendix D, Sample Install Script for Linux for an example.
2.4 Windows Installation

2.4.1 Configure Outgoing Internet Connections

It is important to note that EZD does not support loading or parsing of a proxy .pac file, which details the proxies than can be used based on the client’s region/domain/etc. Instead, the proxy must be specified directly.

2.4.2 Tuning Windows TCP Buffers


2.4.3 Installing EZD for Windows

Complete the Windows installation of EZD by performing the following steps:

1. Install EZD using the installer file EZD-setup.exe. Run the installer as Administrator by right-clicking on EZD-setup.exe and selecting Run as administrator. Refer to Figure 1.

![Figure 1. Run as Administrator](image)

The EZD InstallShield Wizard launches. Refer to Figure 2.

The installer creates (or updates) the configuration file (ezd.cnf) in the installation folder (the default directory is C:\ThomsonReuters\ezd1.3\config).

![Figure 2. EZD - InstallShield Wizard](image)
2. In the EZD InstallShield Wizard, click Next.

3. If you wish to install EZD as a Windows service, verify that the check box for **Install EZD as a Service** is selected and click Next. Refer to **Figure 3**.

![Figure 3. Install EZD as a Service](image)

4. Choose one of the following options:
   - If you are installing EZD on this computer for the first time, go to **Step 5**.
   - If a previous version of EZD is already installed on this computer but you want to update the configuration, select **Make changes to EZD.CNF** and click Next (refer to **Figure 4**). Go to **Step 5**.
   - If a previous version of EZD is already installed on this computer and you choose not to update the configuration, select **Leave EZD.CNF unchanged** and click Next (refer to **Figure 4**). Go to **Step 9**.

![Figure 4. EZD.cnf Options](image)
5. Enter the Elektron Real Time ID and Elektron Real Time Password for your account and click Next. Refer to Figure 5.

Figure 5. Elektron Login and Password

6. Select the applicable options for Hosts for Secure Internet Delivery and Hosts for Delivery Direct Over Leased Lines. If necessary, select Custom and enter a custom string in the text box provided. Click Next. Refer to Figure 6.

Figure 6. Host Input
7. Select the socket connection incoming socket servers you will be using: **SSL** (Marketfeed), **RSSL**, or both and then click **Next**. Refer to **Figure 7**.

![Incoming Socket Servers](image)

**Figure 7. Incoming Socket Servers**

8. If applicable, select the check box for **Enable failback for EZD?** and then select a day of the week or **Everyday**? and the **Time** of day (24-hour clock format). Click **Next**. Refer to **Figure 8**.

![Enable Fallback](image)

**Figure 8. Enable Fallback**
9. Choose one of the following options. Refer to Figure 9.
   • If a user list was created previously or you wish to setup the user list at a later time using the user list configuration tool, uncheck the Add Users selection and click Next. Go to Step 11.
   • If you want to setup users now, select the check box for Add Users and click Next.

![Figure 9. Add Users](image)

10. Enter the User Name, Stream ID, and Stream ID Password and then click Add User To List. Repeat this process for every user that you want to add. When you are finished adding users, click Next. Refer to Figure 10.

![Figure 10. User Information](image)
11. Select the **Destination Folder**. The default is `C:\ThomsonReuters`. If you want to change the destination folder, click **Change...** and select a different destination. Click **Next**. Refer to **Figure 11**.

![Figure 11. Select Destination](image)

12. Click **Install** to begin the installation. Refer to **Figure 12**.

![Figure 12. Install the Program](image)
13. When the installation is complete, click Finish. Refer to Figure 13.

![Figure 13. Installation Successful](image)

2.4.4 Verify Security Protocols

Refinitiv HMDS endpoints use TLS 1.2 security for the encrypted connections. Verify your security settings by going to Control Panel > Internet Settings > Advanced, and scroll down to Security settings. Verify that TLS1.2 is set.
2.5 Configuration

The EZD application requires a configuration file for operation. Refinitiv recommends using installation utilities for automatic configuration. Refer to the sections in this document regarding EZD installation for your respective OS.

2.5.1 Advanced Configuration

A configuration file (ezd.cnf) is included in the installation so that you can tune the EZD application. To connect the EZD to the hosted Refinitiv centers, you must edit this file. In addition to this, if you are using EZD to connect to Elektron Test Environments (ETE), refer to the *EZD for ETE Configuration Guide* on the MyRefinitiv site.

Refer to Table 2.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE TYPE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ezd</em>bindMainThread</td>
<td>String</td>
<td>Use this parameter to bind the EZD process thread. The parameter follows the conventions of TREP systems with the following syntax: P:0 C:0 T:0 Where P=processor number, C=core number, and T=thread number (for hyper-threaded CPU models)</td>
</tr>
<tr>
<td><em>ezd</em>userFile</td>
<td>String</td>
<td>Specifies the path to the datafeed AAA account credentials file.</td>
</tr>
<tr>
<td><em>ezd</em>libNameOpenSSL</td>
<td>String</td>
<td>Conditional. Specifies the name of the OpenSSL library installed on the machine, if different than libssl.so.10 (Linux). EZD uses this parameter to load the specified library if a newer version is installed on the machine or if the default library is not present. In some cases, the default libssl.so.10 may be linked to a newer library. Check your openSSL to ensure you have a supported TLS1.2 version. Note: Only 64 bit libraries are supported.</td>
</tr>
<tr>
<td><em>ezd</em>libNameCrypto</td>
<td>String</td>
<td>Conditional. The name of the cryptographic library installed on the machine, if different than libcrypto.so.10 (Linux). EZD uses this parameter to load the specified library if a newer version is installed on the machine or if the default library is not present. In some cases, the default libcrypto.so.10 may be linked to a newer library. Check your openSSL to ensure you have a supported TLS1.2 version. Note: Only 64 bit libraries are supported.</td>
</tr>
</tbody>
</table>

Table 2: Main EZD Configuration Parameters
2.5.2 Customizing the EZD Device Channel Settings

EZD 1.3 introduces a new persistent device channel to authenticate the EZD device. The device channel includes tunable parameters governing timeouts to the upstream host. The device channel must be connected prior to any datafeed clients connecting to EZD.

2.5.2.1 Parameters

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE TYPE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ezd</em>connectionTimeout</td>
<td>Number</td>
<td>The period of time provided for a channel connection when attempting to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>connect to hosts within the hostlist until timeout. Default value is 60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>seconds.</td>
</tr>
<tr>
<td><em>ezd</em>controllerChannelRetryTime</td>
<td>Number</td>
<td>The expiration time for the EZD controller channel to attempt the next</td>
</tr>
<tr>
<td></td>
<td></td>
<td>connection when the controller channel is disconnected. Default value is 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>seconds.</td>
</tr>
<tr>
<td><em>ezd</em>downloadDictionary</td>
<td>Boolean</td>
<td>Determines the ability of EZD to download the dictionary from the source</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADS. The dictionary is downloaded each time the EZD connects or reconnects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to the upstream ADS. The dictionary is needed for SSL applications to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>connect to the EZD. If set to False, the EZD loads the local RDMFieldDictionary file instead. Default value is True.</td>
</tr>
<tr>
<td><em>ezd</em>dictDownloadTimeout</td>
<td>Number</td>
<td>The expiration time for the EZD device channel to download the dictionary.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the dictionary is not downloaded within the dictDownloadTimeout time,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the dictionary is loaded via file. Default value is 60 seconds.</td>
</tr>
</tbody>
</table>

Table 3: EZD Device Channel Settings
### 2.5.3 Customizing the RSSL Server

The EZD application creates an RSSL server component to accept RWF/RSSL client connections. The RSSL server uses the configuration file to configure the incoming RWF/RSSL client connections. For advanced configuration settings, refer to the *Advanced Distribution Server (ADS) Software Installation Manual* specific to the version that you use or contact your local Refinitiv support representative.

By default, the RSSL Server listens to connections on port 14002. If you want to use a different port, edit the `rsslPort` parameter (in `ezd.cnf`) to set the desired port number.

```plaintext
*ezd*snkRsslServer*rsslPort : 14002
```

EZD can compress data on the downstream RSSL connection to client applications. Compression is disabled by default. Enable compression by setting the `compressionType` field to 1.

- For SSL connections:
  ```plaintext
ezd*snkSslServer*compressionType: 1
```
- For RSSL connections:
  ```plaintext
ezd*snkRsslServer*compressionType: 1
```

**NOTE:** Compression is a CPU-intensive activity that impacts the performance of the EZD application.
2.5.4 Customizing Internet Connections

Every connection to the EZD creates an outgoing connection to a hosted Refinitiv system. By default, an outgoing connection compresses data using zlib compression techniques and uses Transport Layer Security (TLS) encryption for internet connections. An outgoing connection must be configured to connect to the Refinitiv hosted system.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE TYPE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ezd</em>secureConnection*hostList</td>
<td>List of Hostnames</td>
<td>Specifies the list of hosts to connect the EZD. The first host in the list specifies the primary host connection and is used as the primary host for the failback feature. If the connection to the primary host fails, EZD attempts connections to the subsequent hosts in the list until a connection is successful. For details on configuring this parameter, refer to Section 2.5.6.1.</td>
</tr>
<tr>
<td><em>ezd</em>secureConnection*port</td>
<td>Number</td>
<td>Specifies the port number on the hosted Refinitiv system that accepts connections. Refinitiv recommends using 443 for the configuration parameter. Default setting is 443.</td>
</tr>
<tr>
<td><em>ezd</em>secureConnection*proxyHostName</td>
<td>String</td>
<td>Specifies the hostname, IP address, or URL of an outgoing proxy. For Windows connections, proxyHostName should not be configured in ezd.cnf. EZD uses the proxy configuration settings from Internet Explorer.</td>
</tr>
<tr>
<td><em>ezd</em>secureConnection*proxyPort</td>
<td>Number</td>
<td>Specifies the port number of the proxy. For Windows connections, proxyPort should not be configured in ezd.cnf. EZD uses the proxy configuration settings from Internet Explorer.</td>
</tr>
<tr>
<td><em>ezd</em>secureConnection*reencodeSrcDir</td>
<td>Boolean</td>
<td>Sets whether EZD re-encodes incoming source directory messages to change service names unknown to clients into names that clients can understand. Default is True. WARNING! Do not change this parameter.</td>
</tr>
<tr>
<td><em>ezd</em>secureConnection*tcpRecvBufSize</td>
<td>Number</td>
<td>Specifies the number of bytes to allocate to the TCP receive buffer. For this setting to take full effect, OS's TCP receive buffer may require tuning. Refer to the respective OS Installation chapter in this document for TCP buffer tuning. Default setting is 2048000.</td>
</tr>
<tr>
<td><em>ezd</em>secureConnection*tcpSendBufSize</td>
<td>Number</td>
<td>Specifies the number of bytes to allocate to the TCP send buffer. For this setting to take full effect, OS's TCP receive buffer may require tuning. Refer to the respective OS Installation chapter in this document for TCP buffer tuning. Default setting is 2048000.</td>
</tr>
<tr>
<td><em>ezd</em>secureConnection*tryNextHostTime</td>
<td>Number</td>
<td>If the current host fails or is unavailable, this parameter specifies (in seconds) the time EZD waits before trying the next host in the hostList. Default setting is 25.</td>
</tr>
</tbody>
</table>

Table 4: Secure Connections
Table 4: Secure Connections (Continued)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE TYPE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ezd</em>secureConnection</td>
<td>String</td>
<td>This parameter changes the incoming service name provided in the source directory message to a known name used by the clients. Starting in EZD 1.3, the incoming service name is ELEKTRON_DD. Many clients connect to service IDN_RDF; however, the hosted site might not provide the IDN_RDF service. This parameter allows the incoming service (e.g. ELEKTRON_DD) to be renamed (e.g. IDN_RDF). This prevents client applications from changing their configuration files. In version 1.0, clientServiceName was named appServiceName. It was renamed to reduce confusion from corresponding TREP operations.</td>
</tr>
<tr>
<td>*incomingserviceName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*clientServiceName: outgoingserviceName</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.5.5 Customizing Delivery Direct Connections

By default, the EZD installation configures EZD to connect to the internet. Internet connectivity allows clients to remove their leased telecommunication lines that are connected to Refinitiv. However, a client may wish to retain these leased lines and use EZD to connect directly to the Refinitiv hosted endpoint. This connection method is called Delivery Direct. When configuring the EZD to use the Delivery Direct method, a few parameters must be changed. Refer to Table 5.

NOTE: Only one exclusive connectivity mode is used, either secure internet or delivery direct upstream connection.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE TYPE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ezd</em>secureConnection*hostList:</td>
<td>List of Hostnames</td>
<td>Refer to Section 2.5.6.2, Hosts for Delivery Direct Connections.</td>
</tr>
<tr>
<td><em>ezd</em>secureConnection*port:</td>
<td>Number</td>
<td>Specifies the port number on the hosted Refinitiv system that accepts connections. Default setting is 443.</td>
</tr>
<tr>
<td>I <em>ezd</em>secureConnection*proxyHostName</td>
<td>IP Address or URL</td>
<td>Comment out this parameter by inserting a “!” at the beginning of the line. If you switch to the secure internet connectivity mode, re-enable this parameter.</td>
</tr>
<tr>
<td>I <em>ezd</em>secureConnection*proxyPort</td>
<td>Number</td>
<td>Comment out this parameter by inserting a “!” at the beginning of the line. If you switch to the secure internet connectivity mode, re-enable this parameter.</td>
</tr>
</tbody>
</table>

Table 5: Delivery Direct Connection Parameters
2.5.6 Host Endpoint Configuration

The particular Refinitiv hosted system that you connect to depends on the region where your site is located. You should also use a backup host. Configure the host in the `ezd*secureConnection*hostList` parameter.

The following hosts are for Elektron Connect only. For Elektron Test Environments hosts, refer to the EZD for ETE Configuration Guide on the MyRefinitiv site.

2.5.6.1 Hosts for Secure Internet Connections

Use the following host configuration depending on your region:

- **EMEA**: emea1-ec-hmds.cp.thomsonreuters.com, emea2-ec-hmds.cp.thomsonreuters.com
- **AMERS**: amers1-ec-hmds.cp.thomsonreuters.com, amers2-ec-hmds.cp.thomsonreuters.com
- **APAC**: apac1-ec-hmds.cp.thomsonreuters.com, apac2-ec-hmds.cp.thomsonreuters.com

2.5.6.2 Hosts for Delivery Direct Connections

Use the following host configuration depending on your region:

- **EMEA**: emea1-ec-hmds.cp.extranet.thomsonreuters.biz, emea2-ec-hmds.cp.extranet.thomsonreuters.biz
- **AMERS**: amers1-ec-hmds.cp.extranet.thomsonreuters.biz, amers2-ec-hmds.cp.extranet.thomsonreuters.biz
- **APAC**: apac1-ec-hmds.cp.extranet.thomsonreuters.biz, apac2-ec-hmds.cp.extranet.thomsonreuters.biz

2.5.7 User configuration file

The user configuration file maps user client names (i.e. datafeed Stream ID) to associated AAA account permissions (i.e. authentication keys). The user configuration file path must be specified in the EZD configuration file (`ezd.cnf`). For details on setting up a user configuration file, refer to Chapter 4, Permissioning/Authentication.

Example:

```
*ezd*userFile : user_list.txt
```
2.6 Initialization

2.6.1 OpenSSL Library Loading

The EZD application uses dynamic loading of the OpenSSL libraries. In Linux environments, the OpenSSL libraries may not be part of the library search path. To fix this, export the environment variable (LD_LIBRARY_PATH) and edit it to contain the path to the libssl and libcrypto libraries.

The EZD application expects default libraries of libssl.so.10 and libcrypto.so.10. libcrypto.so.10 and libssl.so.10 should symbolically link to newer openssl libraries. If newer libraries are installed, refer to Section 2.2 for details on loading non-default OpenSSL libraries for the EZD application.

2.6.2 Starting the EZD in Linux

**NOTE:** Before starting the EZD, complete Section 2.5.7 to ensure that the user_list.txt file is configured.

▶ Run the EZD by entering the following command:

Specify the start file on the command line:

```
start_ezd
```

2.6.3 Starting the EZD in Windows

**NOTE:** Before starting the EZD, complete Section 2.5.7 to ensure that the user_list.txt file is configured.

2.6.3.1 Run the EZD as a service:

When the EZD is installed as a service, the EZD process automatically starts upon installation and any time the machine reboots. If the EZD shuts down due to error, the EZD process automatically restarts within one minute. Refer to your system Event Logger for EZD event notifications.

**NOTE:** If the Microsoft C++ runtime libraries are not installed, EZD cannot start and the Event Logger cannot provide specific information as to why the program terminated.

2.6.3.2 Run the EZD as a stand-alone process

▶ Run the EZD by entering the following command:

In Windows, open a command window in Administrator mode. This is necessary to connect to shared memory.

Start EZD on the command line:

```
WARNING! If entering ezd.exe on the command line, you must specify either the -nodaemon flag or the -c config_file option to prevent a startup error.
```

```
ezd.exe -nodaemon
```
3   Functionality

3.1   EZD Persistent Controller Channel

The new AAA authentication present at the HMDS site endpoints requires that each connecting Refinitiv controller has a persistent connection with Elektron Real Time ID and Elektron Real Time Password credentials (authentication key). EZD 1.3 fulfills these criteria by having a persistent controller channel that is connected to an HMDS endpoint. All subsequent datafeed client connections connect to the same endpoint as EZD 1.3. This is different from prior EZD versions where the datafeed client connections could connect to different hosts in the hostlist. The controller channel is used to send EZD username and password credentials, as well as retrieve the data dictionary from the HMDS endpoint. The data dictionary is retrieved each time the controller channel establishes a connection to the host endpoint. Refer to Section 6.2 for more information about the dictionary download.

3.1.1   Device Channel Failure and Reconnection

If the EZD device channel disconnects during EZD operation, the HMDS endpoint may close all the associated datafeed client connections. If the controllerChannelRetryHostListTime parameter is set above 0 (disabled by default), the EZD performs auto-recovery on the controller channel by attempting to reconnect to the same previously connected HMDS endpoint every 15 seconds. If auto-recovery continues to fail, the EZD considers other hosts in the hostlist after the controllerChannelRetryHostListTime expires. While the EZD controller channel is down, the EZD disables the ability for datafeed clients to connect. Once the EZD controller channel is established, datafeed clients may reconnect.

3.1.2   Endpoint Host Change

The EZD may connect to a different host through auto-recovery or failback. Once the EZD changes the endpoint host, all datafeed client connections are disconnected. The clients need to perform connection recovery.

3.1.3   Password Change

EZD 1.3 uses a controller authentication key to login into the HMDS endpoint. The client site has control over the EZD password permissioning. If the client site wants to change the EZD password, perform the following steps:

1. Invoke another instance of EZD by entering the following command:

   ezd -encryptPW <clear text password>

2. The command outputs an encrypted string password. Copy this text.

3. Modify the EZD configuration file by replacing the value in *ezd*controllerAccountPassword: with the new password value.

4. Upon the next authentication cycle, the EZD is required to login with the new authentication key.

   The upstream permissioning service discovers that the EZD authentication key is no longer valid and forcibly logs out the EZD controller channel. This action may cause the datafeed client channels to disconnect as well.
3 Functionality

3.2 Connection Handling

The EZD application includes the option to create an RSSL server to handle incoming RSSL/RWF client connections and/or an SSL server to handle incoming SSL 4.0/4.0BE client connections (SSL 4.5 client connections are not compatible with EZD at this time). Each connection creates a corresponding upstream RSSL connection over TLS encryption to a Refinitiv hosted site. If EZD fails to connect to a Refinitiv hosted system, EZD sends a failure status message to the client. EZD logs incoming and outgoing connections/drops into a log file. Clients are responsible for reconnecting.

Starting in EZD 1.3, outgoing connections use the Transport Layer Security (TLS) Encryption version 1.2 protocol.

3.3 Ping Handling

EZD implements incoming and outgoing pings for the connection channels. You can configure the connection parameters to adjust the frequency of pings. Ping messages are not sent in instances that contain the same traffic flow. For example, an outgoing ping message is not sent while EZD is sending outgoing messages. However, incoming ping messages on that channel should be expected because the datafeed client rarely sends messages upstream.

3.4 Message Handling

EZD is a pass-through proxy for incoming and outgoing messages. If needed, you can configure EZD to assign an alias to the incoming service name (provided by the Refinitiv hosted systems) in source directory messages to names used by clients. For details on the aliasing operation, refer to Section 2.5.4, Customizing Internet Connections. Starting in EZD1.3, the incoming service name is ELEKTRON_DD.

| ezd*secureConnection*ELEKTRON_DD*clientServiceName: IDN_RDF |

3.5 Proxy support

EZD supports non-authenticating HTTP internet proxy connections. Set the proxy information in the configuration file.

NOTE: For EZD on Windows, the proxy settings are set in Internet Explorer and not the configuration file.

3.6 Required files

The EZD application requires the following files:

- Configuration file (ezd.cnf)
- User datafeed account mapping file (user_list.txt)
4 Permissioning/Authentication

Datafeed client devices connecting to EZD must have an entry in the `user_list.txt` that specifies the Stream ID and Stream ID Password credentials. The file includes five columns delimited by commas with the following format:

- Column 1: The username provided by the datafeed client application.
- Column 2: The translated Stream ID.
- Column 3: The value for this column is 1.
- Column 4: The value for this column is 1.
- Column 5: The Stream ID password. The password must be encrypted.

Example file:

```
Bob,DF-UUID-X,1,1,DsIB25IF_XfodozOZEtQ9_7pRMZGrEzGvrxhjM5Q52Z03woJ5cbS_rDqgp2Vt01U_t1gSL8TtQJMv5_eSNuL5aPKr7bk
Katie,DF-UUID-X,1,1,F6k58tk91BTHeZaru5cF1rNjSziZuZi3D7VRStcB8zavnLg7iPm9B1UeFprIJo9zDMUVunrGlxbZxKUT8f7kbexXha
```

4.1 Password Change

The authentication system used in EZD allows clients to change their datafeed passwords. When the datafeed client requires a password change, the authentication system hosted by Refinitiv must be updated and the local file `user_list.txt` must be modified.

Consult the Refinitiv AAA License Manager documentation to change your datafeed client password on the Refinitiv system or consult Refinitiv support.

To change the local file, use the password management utility included as part of the EZD installation. Refer to Appendix F for more information.
5 Configuring ADH or ADS/POP Connections

You can connect your ADH or ADS/POP through an EZD by configuring a route in the ADH or ADS/POP configuration file. Create a route by using standard route configuration parameters (edit the route options listed below).

The following example creates a route called “route1.route”.

*adh*route1.route*hostList : EZD hosted machine, typically localhost
*adh*route1.route*port : EZD rsslServer port, typically 14002
*adh*route1.route*protocol : rssl
*adh*route1.route*requestTimeout : 30
*adh*route1.route*serviceList : ELEKTRON_DD
*adh*route1.route*userName : A permissioned user name with an associated Stream ID in the user_list.txt file.
*adh*route1.route*SERVICE*cacheLocation: ssl
# 6 Reuters SSL Connections

## 6.1 Introduction

EZD includes support for Reuters SSL 4.0 and 4.0BE connections from connecting datafeed applications. EZD is responsible for converting Reuters SSL messages received from the application to RSSL and then sending them upstream over a secure channel. EZD also converts RSSL messages received over secure channels to Reuters SSL before handing them over to the application.

**NOTE:** EZD does not support SSL 4.5 connections.

Refer to [Table 6](#) through [Table 9](#) for supported messages.

<table>
<thead>
<tr>
<th>SSL MESSAGES</th>
<th>CORRESPONDING RSSL MESSAGES</th>
</tr>
</thead>
</table>
| IPC_PT1_MOUNT_REQ      | RssslRequestMessage  
msgClass : RSSL_MC_REQ  
domainType : RSSL_DMT_LOGIN |
| IPC_PT1BE_SRC_TABLE_REQ| RssslRequestMessage  
msgClass : RSSL_MC_REQUEST  
domainType:RSSL_DMT_SOURCE |
| IPC_PT1BE_CONFIG_REQ   | Dictionaries are pre loaded for EZD so EZD echoes back with the dictionaries it already has loaded. |
| IPC_PT1BE_PING_RESPONSE| Pings from application are responded to by EZD they are not forwarded upstream. |
| IPC_PT1BE_OPEN_REQ     | RssslRequestMessage  
msgClass: RSSL_MC_REQUEST  
domainType : RSSL_DMT_MARKET_PRICE |
| IPC_PT1BE_CLOSE_REQ    | RssslCloseMessage  
msgClass : RSSL_MC_CLOSE  
domainType: RSSL_MC_MARKET_PRICE |
| IPC_PT1BE_PRIORITY_REQ | RssslRequestMessage  
msgClass : RSSL_MC_REQUEST  
domainType : RSSL_DMT_MARKET_PRICE  
Following flags are set on request message  
RSSL_RQMF_NO_REFRESH, RSSL_RQMF_HAS_PRIORITY |
| IPC_PT1BE_INSERT       | RssslPostMessage  
msgClass : RSSL_MC_POST  
domainType : RSSL_DMT_MARKET_PRICE |

**Table 6: Supported SSL Messages from APP to EZD**
<table>
<thead>
<tr>
<th>RSSL MESSAGES FROM UPSTREAM</th>
<th>CORRESPONDING SSL MESSAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>RsslRefreshMessage</td>
<td>IPC_PT1_MOUNT_ACK</td>
</tr>
<tr>
<td>msgClass : RSSL_MC_REFRESH</td>
<td></td>
</tr>
<tr>
<td>domainType : RSSL_DMT_LOGIN</td>
<td></td>
</tr>
<tr>
<td>RsslStatusMessage</td>
<td>IPC_PT1_MOUNT_NAK</td>
</tr>
<tr>
<td>msgClass: RSSL_MC_STATUS</td>
<td></td>
</tr>
<tr>
<td>domainType: RSSL_DMT_LOGIN</td>
<td></td>
</tr>
<tr>
<td>RsslRefreshMessage</td>
<td>IPC_PT1BE_SERVICE_INFO</td>
</tr>
<tr>
<td>msgClass: RSSL_MC_REFRESH</td>
<td></td>
</tr>
<tr>
<td>domainType: RSSL_DMT_SOURCE</td>
<td></td>
</tr>
<tr>
<td>RsslUpdateMessage</td>
<td></td>
</tr>
<tr>
<td>msgClass: RSSL_MC_UPDATE</td>
<td>IPC_PT1BE_SERVICE_INFO</td>
</tr>
<tr>
<td>domainType: RSSL_DMT_SOURCE</td>
<td>IPC_PT1BE_GROUP_STATUS</td>
</tr>
<tr>
<td></td>
<td>IPC_PT1BE_GROUP_MERGE</td>
</tr>
<tr>
<td>RsslRefreshMessage</td>
<td></td>
</tr>
<tr>
<td>msgClass: RSSL_MC_REFRESH</td>
<td>IPC_PT1BE_IMAGE</td>
</tr>
<tr>
<td>domainType: RSSL_DMT_MARKET_PRICE</td>
<td></td>
</tr>
<tr>
<td>RsslUpdateMessage</td>
<td></td>
</tr>
<tr>
<td>msgClass: RSSL_MC_UPDATE</td>
<td>IPC_PT1BE_UPDATE</td>
</tr>
<tr>
<td>domainType: RSSL_DMT_MARKET_PRICE</td>
<td></td>
</tr>
<tr>
<td>RsslStatusMessage</td>
<td></td>
</tr>
<tr>
<td>msgClass: RSSL_MC_STATUS</td>
<td>IPC_PT1BE_STATUS</td>
</tr>
<tr>
<td>domainType: RSSL_DMT_MARKET_PRICE</td>
<td></td>
</tr>
<tr>
<td>RsslAckMessage</td>
<td></td>
</tr>
<tr>
<td>msgClass: RSSL_MC_ACK</td>
<td>IPC_PT1BE_INSERT_ACK</td>
</tr>
<tr>
<td>domainType: RSSL_DMT_MARKET_PRICE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPC_PT1BE_INSERT_NAK</td>
</tr>
</tbody>
</table>

Table 7: Supported SSL Messages from EZD to APP

EZD sends IPC_PT1BE_PING message to the application periodically based on the value of the parameter pingInterval.
## RSSL Status Code to SSL Status Code

<table>
<thead>
<tr>
<th>RSSL Status Code</th>
<th>SSL Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note: The following SSL Status Codes are SSL constants. Do not rename them.</td>
<td></td>
</tr>
<tr>
<td>RSSL_SC_NONE</td>
<td>SSL_INFO_NONE</td>
</tr>
<tr>
<td>RSSL_SC_NOT_FOUND</td>
<td>SSL_INFO_NONE</td>
</tr>
<tr>
<td>RSSL_SC_TIMEOUT</td>
<td>SSL_INFO_NONE</td>
</tr>
<tr>
<td>RSSL_SC_NOT_ENTITLED</td>
<td>SSL_INFO_NO_PERMISSION</td>
</tr>
<tr>
<td>RSSL_SC_INVALID_ARGUMENT</td>
<td>SSL_INFO_NONE</td>
</tr>
<tr>
<td>RSSL_SC_USAGE_ERROR</td>
<td>SSL_INFO_NONE</td>
</tr>
<tr>
<td>RSSL_SC_PREEMPTED</td>
<td>SSL_INFO_PREEMPTED</td>
</tr>
<tr>
<td>RSSL_SC_JIT_CONFLATION_STARTED</td>
<td>SSL_INFO_NONE</td>
</tr>
<tr>
<td>RSSL_SC_REALTIME_RESUMED</td>
<td>SSL_INFO_NONE</td>
</tr>
<tr>
<td>RSSL_SC_FAILOVER_STARTED</td>
<td>SSL_INFO_FAILOVER_START</td>
</tr>
<tr>
<td>RSSL_SC_FAILOVER_COMPLETED</td>
<td>SSL_INFO_FAILOVER_COMPLETE</td>
</tr>
<tr>
<td>RSSL_SC_GAP_DETECTED</td>
<td>SSL_INFONONE</td>
</tr>
<tr>
<td>RSSL_SC_NO_RESOURCES</td>
<td>SSL_INFO_NO_RESOURCES</td>
</tr>
<tr>
<td>RSSL_SC_TOO_MANY_ITEMS</td>
<td>SSL_INFO_TOO_MANY_ITEMS</td>
</tr>
<tr>
<td>RSSL_SC_ALREADY_OPEN</td>
<td>SSL_INFO_NONE</td>
</tr>
<tr>
<td>RSSL_SC_SOURCE_UNKNOWN</td>
<td>SSL_INFO_SRC_UNKNOWN</td>
</tr>
<tr>
<td>RSSL_SC_NOT_OPEN</td>
<td>SSL_INFO_NOT_OPEN</td>
</tr>
<tr>
<td>RSSL_SC_NON_UPDATING_ITEM</td>
<td>SSL_INFO_NON_UPDATING_ITEM</td>
</tr>
</tbody>
</table>

**Table 8: Mapping Table for RSSL Status Code to SSL**

## RSSL Data State to SSL Data State

<table>
<thead>
<tr>
<th>RSSL Data State</th>
<th>SSL Data State</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSSL_DATA_NO_CHANGE</td>
<td>SSL_S_NO_CHANGE</td>
</tr>
<tr>
<td>RSSL_DATA_OK</td>
<td>SSL_S_OK</td>
</tr>
<tr>
<td>RSSL_DATA_STALE</td>
<td>SSL_S_STALE</td>
</tr>
</tbody>
</table>

**Table 9: Mapping Table for RSSL Data State to SSL**
6.2 Dictionary Download

When Reuters SSL client connections are enabled via the `enableSslServer` parameter, EZD loads data dictionaries to support RWF to Marketfeed conversion. EZD reads the `enum` file from the location provided via the following parameter:

```
*ezd*enumFile : ./enumtype.def
```

If EZD cannot find a valid `enum` file to load, it disables SSL connections.

By default, EZD attempts to download `RDMFieldDictionary` from upstream. Reuters SSL client connections are disabled until the dictionary is successfully loaded. The `dictDownloadTimeout` (Default is 60 seconds) parameter determines how long EZD waits to download the dictionary. If EZD cannot download the dictionary within that time, then EZD attempts to load the dictionary from the file specified by the configuration parameter `fieldDictionary`. If EZD is unable to load the dictionary via download or from the file, then it logs failure to download dictionary and disables SSL client connections.

If both RSSL and SSL connections are enabled for EZD, failure to download the dictionary or read it from the file disables only SSL connections; EZD continues to accept RSSL connections.

**NOTE:** If the EZD reconnects or changes host endpoints, the EZD re-downloads the data dictionary. Note that this does take a considerable amount of bandwidth for a brief amount of time.

6.3 Parameters for EZD SSL

The following parameters support the SSL connections:

```
*ezd*enableSslServer : True
*ezd*snkSslServer*sslPort : 8101
*ezd*snkSslServer*guaranteedOutputBuffers : 200
*ezd*snkSslServer*maxOutputBuffers : 400
*ezd*snkSslServer*numInputBuffers : 10
*ezd*pingInterval : 20
*ezd*pingKillInterval : 60
*ezd*snkSslServer*reservedFileDescriptors : 64
*ezd*snkSslServer*allowCompMode : True
*ezd*snkSslServer*tcpRecvBufSize : 524288
*ezd*snkSslServer*tcpSendBufSize : 524288
!*ezd*snkSslServer*compressionType : 1
!*ezd*snkSslServer*zlibCompressionLevel : 5
```
7 EZD Failback

7.1 Overview

The failback feature allows the EZD controller channel to detect if the channel is connected to the primary host of the providing service. In some cases, the connected host may be a backup or secondary source providing data. This feature only applies to the EZD controller channel. All datafeed client channels use the same host endpoint as the EZD controller channel.

The EZD seeks a northbound connection to the Elektron HMDS via the \texttt{*ezd*secureConnection*hostList} parameter. The EZD controller channel attempts a single connection to the first host in the list and then successively to each host thereafter when failures to the respective hosts have occurred. The hosts at the end of the \texttt{*ezd*secureConnection*hostList} may reference hosts that are distant for the application. In version 1.0, if the northbound channel is connected to a distant host, the EZD process must be restarted in order to have all clients attempt their connections to the desired near host. The failback feature is an automated way to move clients to a desired host.

7.2 Operation

The EZD Failback feature operates via a timer or through manual intervention. The EZD also includes configurable variables to test the failback of all connections or individual connections.

The failback timer operates similarly to a Linux cron timer. The timer is set to expire at a certain time of the day, either once a day or on a certain day of the week. The timer expiration settings are configurable, but the timer cannot be turned off. The timer settings expire at a planned maintenance window interval.

\textbf{NOTE:} The timer settings are based on GMT time.

The second method of failback operation is via manual intervention.

The failback feature logs messages regarding connection attempts, successes, and failures. Starting in EZD 1.3, the failback feature is only applicable to the EZD controller channel. If the controller channel is connected to the primary host, a failback test connection is not attempted. If the controller channel is not connected to the primary host, a failback test connection occurs. If the failback test connection succeeds, the original controller channel connection and all datafeed client connections to the EZD are terminated. The EZD controller channel performs auto-recovery to connect to the primary host site. The datafeed client connections must be terminated in order to open streaming items to the new connection.

7.3 Configuration

Add the list of hosts to that connect to EZD to the \texttt{*ezd*secureConnection*hostList} parameter. The first host in the list specifies the primary host connection and is used as the primary host for the failback feature. If the connection to the primary host fails, EZD attempts connections to the subsequent hosts in the list until a connection is successful.
## 7.3.1 Variables for Timer Operation

To support Failback, add the Failback configuration variables listed in Table 10.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE TYPE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ezd</em>enableFailbackMgr</td>
<td>Boolean</td>
<td>Enables the Failback timing and manual intervention feature. If false, the timer does not execute and the user is unable to manually perform failback operations.</td>
</tr>
<tr>
<td><em>ezd</em>failbackTime</td>
<td>Number</td>
<td>Sets the timer to the desired hour and minute for expiration. The allowable times are 0 - 23 for the hour and 0 - 59 for the minute. Times are specified in GMT. The default setting is 00:00.</td>
</tr>
<tr>
<td><em>ezd</em>failbackDayOfWeek</td>
<td>Number</td>
<td>Sets the day of the week on which the failback timer expires. Allowable values are 0 - 8, where: 0 - 6 = Sunday through Saturday (respectively), 7 = Sunday (to match cron), 8 = everyday. The default setting is 0.</td>
</tr>
<tr>
<td><em>ezd</em>failbackChannelUptime</td>
<td>Number</td>
<td>Sets how long (in seconds) the test failback channel connection remains connected. The uptime indicates connection health, especially in cases where the connection could be intermittent. The test connection cannot exceed 60 seconds because a login message is not sent to the upstream source. The upstream source removes connections after 60 seconds if a login message is not sent. The default setting is 45.</td>
</tr>
<tr>
<td><em>ezd</em>failbackChannelTimeout</td>
<td>Number</td>
<td>Sets how much time the EZD spends trying to connect to the primary host. If the specified time expires, the failback test connection fails. The default setting is 45.</td>
</tr>
</tbody>
</table>

Table 10: Failback Timer Parameters
7.3.2 Manual Operation

Initiate manual failback by changing the following parameters in ezdmon (ManagedProcess.ezd object). Refer to Appendix G for more information regarding the use of ezdmon.

▶ Test failback ability for all channels by entering the following command:

Change the following parameter in the HTTPSrcChannel object in the mob.

```
FailbackAllToPrimaryHosts
```

▶ Test failback ability for a single connection by entering the following command:

Change the following parameter in the HTTPSrcChannel object in the mob.

```
FailbackToPrimaryHost
```

The HTTPSrcChannel Stats screen contains new variables that show:

- The primary host name
- A true/false indicator to indicate if the connection is connected to the primary host
8 Logging and Monitoring

The EZD application maintains a log file (`ezd.log`) regarding application events such as login attempts and datafeed client disconnects. It also publishes statistics into shared memory. You can use the `ezdmon` monitoring tool to look into the shared memory segment. To connect to the shared memory segment, you must provide `ezdmon` the path to the configuration file. By default, the EZD monitor opens in the managed object browser (mob) mode.

**NOTE:** When specifying the configuration file using Windows, be sure to use the forward slash (`/`) as the directory separator as opposed to the backslash (`\`).

```
ezdmon -c ezd.cnf
```
### 9 Endpoint Site Maintenance

Sites that host EZD connections undergo periodic maintenance that may affect EZD connectivity. The following maintenance schedule is posted for reference. EZD does not provide an alert for external site maintenance activity. If EZD connections are affected, EZD attempts to reconnect through failback.

#### 9.1 Regional Low Usage (RLU)

Maintenance in RLU is considered low risk and should not impact service. Maintenance windows occur daily from 22:00 to 02:00 local time per region. The following table shows the maintenance windows across the regions:

<table>
<thead>
<tr>
<th>REGION</th>
<th>MAINTENANCE WINDOW IN GMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMERS</td>
<td>03:00 GMT - 07:00 GMT</td>
</tr>
<tr>
<td>APAC</td>
<td>14:00 GMT - 18:00 GMT</td>
</tr>
<tr>
<td>EMEA</td>
<td>22:00 GMT - 02:00 GMT</td>
</tr>
</tbody>
</table>

**Table 11: Maintenance Windows**

#### 9.2 Global Low Usage (GLU)

Maintenance in GLU carries higher risk and may be service impacting. EZD may not be able to connect to preferred sites during these times. The following maintenance windows are used during GLUs:

- GLU1: Friday 23:00 GMT > 07:00 GMT Saturday
- GLU2: Saturday 14:00 GMT > 04:00 GMT Sunday
Appendix A  Troubleshooting

Problems with EZD are typically broken down into two areas:

- Initial connection of EZD to the internet. The initial connection includes complexity due to the following requirements:
  - Specify the internet proxy
  - Configure the controller and datafeed client profiles
  - Complete the SSL handshake
- Maintaining EZD connection to the internet. Problems maintaining the connection to the internet are typically related to the quality of the ISP. Details are provided for determining the reliability and bandwidth required for the link.

A.1  Startup Issues: Windows Shared Memory

EZD must run in a command window with administrative privileges in order to attach to shared memory. Verify that your command window has administrative privileges.

A.2  Problems with Initial Connection to the Internet

A.2.1  Proxy Configuration

Most user sites do not have direct access to the internet. In this case, an internet proxy is used for connectivity. The internet proxy is configured via the `proxyHost` and `proxyPort` configuration parameters.

Proxy settings for Windows machines are configured through Internet Explorer. Windows users should not configure the `proxyHost` and `proxyPort` configuration variables.

It is important to note that EZD does not support loading or parsing of a proxy `.pac` file, which details the proxies than can be used based on the client's region/domain/etc. Instead, the proxy must be specified directly. If a client configures `proxyHost` to be a web server that contains the proxy `.pac`, the following is logged:

```
<ezd.1.ezd: Error: Tue Aug 19 14:39:38.630202 2014>
Channel initialization failed for socket 10 to host emea1-ec-hmds.cp.thomsonreuters.com error code -1.
Text: <Impl/ripcsrvr.c:5949> Error: 1002 Could not read Proxy HTTP Ack received. System errno: (11)
Text: HTTP/1.1 400 Bad Request^M
Content-Type: text/html; charset=us-ascii^M
Server: Microsoft-HTTPAPI/2.0^M
Date: Tue, 19 Aug 2014 14:44:48 GMT^M
Connection: close^M
Content-Length: 324^M
^M
<!DOCTYPE HTML PUBLIC "-/W3C//DTD HTML 4.01//EN""http://www.w3.org/TR/html4/strict.dtd">^M
<HTML><HEAD><TITLE>Bad Request</TITLE>^M
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></HEAD>^M
<BODY><h2>Bad Request - Invalid URL</h2>^M
<hr><p>HTTP Error 400. The request URL is invalid.</p>^M
</BODY></HTML>^M
```

Test a proxy by configuring a web browser with a proxy. In Internet Explorer, click Tools > Internet Options > Connections > LAN Settings. Click the Proxy server box and enter the hostname and port of the proxy. If the web browser connects to the internet, these same settings should work to connect the EZD to the internet with the internet proxy.
A.2.2 Hostname Lookup

If the EZD is unable to resolve the hostname in the hostList file, the following is logged:

```
Failed to connect secure Channel to host hostname. Error:<Impl/ripcsrvr.c:xx>
Error: xx rpcHostByName() failed.
Hostname is incorrect. System errno: (24).
```

Typically, this denotes a typographical error in the hostList. However, it also occurs if the internet connection is down; in which case, access to DNS is lost. As long as a simple ping works to reach the hostname, then this error log should not be present.

A.2.3 Connection Timeouts

Connection timeouts are the most common scenario for connection problems. The EZD includes a timeout parameter. If there is no response after 25 seconds, then the EZD connection to the specified host did not complete. This could be due to intermittent Internet connectivity, excessive delay when reaching the host, or other possibilities. If other hosts are specified in the hostlist, successive host connections are attempted. When a connection timeout occurs, the following is logged:

```
Socket # trying other hosts due to host connection timeout for host hostname.
```

Generally, a ping to the proxy or server host is the next step to reestablish basic connectivity. If an internet connection is detected, then verify the port settings for both the proxy and the server host. In most cases, server host is port 443 and the proxy is either 8080 or 8081.

A.2.4 Downstream Disconnects

Downstream application disconnects generate the following text:

```
Disconnecting client 8 from source hostname. Removing secure channel socket #.
Reason : client disconnect; text from client: rsslRead failed with code -1 and system error 0. Text: <Impl/ripcsrvr:xxxx> Error : 1002 ripcRead() failure.
Connection reset by peer.
```

Downstream disconnects may occur for normal reasons such as component restart, bandwidth exhaustion, or network outage. In some cases, the EZD upstream connection timeout is longer than the downstream connection timeout. In this case, the above text is logged as client disconnected and reconnected later. More specifically for the ADH, the default disconnect timeout is 10 seconds. If EZD cannot connect upstream, the ADH times out the login request and reconnects 10 seconds later. In this case, Refinitiv recommends that you configure the ADH requestTimeout parameter to 30 seconds.

For recommended TCP tuning parameters for Linux, refer to Section 2.3.1, Tuning Linux TCP buffers.
A.2.5 Misconfigured UserName

All users that connect to EZD must specify a mapping from incoming user name to datafeed account credentials. This is controlled with the user_list.txt file. For some applications, the transmitted user name is not configurable.

If a user attempts to connect and does not have account credentials mapping in the user_list.txt file, the user name is not mapped and is denied a login. Alternatively, if the user name is mapped to proper account credentials and the account credentials are no longer valid, the user cannot login. In either case, the following login status message is displayed in the ezd.log:


A.2.5.1 Client Application Logs

The login denied message is not displayed by the EZD. Instead, the following message is printed to adh.log or the application log file:

Session disconnected (route "adh2_rdf", host "ezd2", port 14002). Login request for user "#219687" has been denied. Text:
DACS Gateway connection is down
Connection request will be sent periodically.
Message will be logged when channel login succeeds.

EZD detects the following login issues and prints the associated status message to the ezd.log:

- EZD datafeed client incorrect login name and/or password:

  secureChannel.3 The EZD process has intercepted a login response failure to the client from host
  oaklx8655-vm2 on socket 11. Reason from source:Login Denied:Failed to authenticate:401:
  <END>

The DACS Gateway connection is down text is not very informative, but it does note that the user's credentials were incorrect. Verify credential configuration is correct by ensuring that the user name, Stream ID, and Stream ID password in the user_list.txt file is correct.

At this point, verifying the user name specified in the login is necessary. The user name is configured in the userName parameter in the ADH configuration. The username should map to a Stream ID and Stream ID password contained in the user_list.txt file on the EZD machine. The name must be present in the user_list.txt along with the proper user credentials. To ensure EZD reads this file correctly, verify the RegisteredUserDatabase with mob. The bottom of the mob screen displays user entries in the format of "userXX", where XX is a number. Find the user entry which corresponds to the user name and Stream ID.

If you are still unable to login with the user name, connect to EZD with rmdstestclient using the ADH or application user name along with the -l stdout option and save the results to a file. Send the file to Refinitiv support for further analysis.
A.3  Content Issues

Content received from EZD is identical to what is received from EED Full Tick and EED Optimized. However, there may be a handful of indirect differences.

A.3.1  Lack of Content Due to Region

Some small subsets of content are only available in EMEA, AMERS, and APAC. As such, failover from APAC to EMEA, for example, could result in content being unavailable despite the fact the user has the required permissions for that content. The only workaround is to failback to the primary data center as soon as it is available.

Permission failures take the form of status closed responses with the info code and status text of PERMISSION_DENIED.

A.3.2  Eikon Quote Lists Missing fields

Eikon relies on consolidated fields (such as fields in dictionary that start with CF_) to populate the quote list. These fields are generated in one of the following ways:

- By the RDF when consolidated field mapping logic is enabled (The default is disabled). EED does not have this logic
- Within HMDS by a component called the field mapping server, but only for hosted Eikon
- Within the Eikon desktop itself in the event the previous two are not available

The first two methods come at a CPU and bandwidth premium for all downstream devices and are not the desired direction.

The third method works, but only if the service is configured as the IDN feed. Refer to Figure 14.

![Figure 14. IDN Feed Configuration](image)

For clients that have RDF and Eikon, the expectation is that they have IDN_RDF as their IDN or Elektron Feed and add a new feed for Elektron Zero Daemon to compare the two. This does not work for quote lists, because Eikon can have only a single IDN or Elektron feed.
A.4 Connectivity Issues – Client/EZD Disconnects

A.4.1 Internet Connectivity Issues

After the connection is established, the most common problem a client experiences is a disconnect due to internet issues or high latencies for data.

EZD disconnects from the internet are reported as rsslRead or rsslWrite failures in the ezd.log file. Channel disconnects are detected by the OS when writing or reading to a socket fails.

**NOTE:** The EZD maintains a one-to-one mapping between datafeed client channels and upstream channels. If the upstream channel is lost, it causes the downstream datafeed client connection to disconnect.

Internet disconnects can occur even if the internet link appears healthy. However, because this is TCP/IP, sudden, heavy latency spikes, heavy packet loss, or exceeding bandwidth at an instant in time where bandwidth is shared with other clients can result in the upstream ADS cutting the channel to avoid unmarked stale data. EZD connects to ADS with traffic management enabled such that just in time conflation takes place during periods of sudden congestion. However, just in time conflation is trade-safe, so trade spikes alone can cause channel disconnects.

From the client side, www.speedtest.net may be used to determine the ping and bandwidth characteristics of the internet connection. Contact a Refinitiv representative for bandwidth requirements per watchlist size.

Loss is determined by measuring duplicate or selective ACKS reported by the OS. The greater the number of duplicate or selective ACKS, the greater the likelihood that TCP cannot recover and results in a channel cut.

The **netstat -s -f tcp** command on Linux platforms may be used to determine the number of selective or duplicate ACKS reported by the OS.

A.4.2 EZD Device Channel disconnection

If the EZD controller channel is suddenly disconnected, all datafeed clients may be disconnected by the host endpoint.

Errors in the upstream authentication system may forcibly disconnect the EZD controller channel. You should ensure that the EZD controller account credentials are correct. If the credentials are correct and you continue to have connectivity issues, contact your local Refinitiv representative and notify them of this issue. Below is a sample message where the EZD controller channel was forcibly disconnected.

```
secureChannel.7 EZD persistent controller channel has received a failure message on the LOGIN stream.
   Error logging into the system.
<END>

Disconnecting secure channel secureChannel.7. Reason: Force Logout from DACS.
<END>
```
Appendix B  FAQs

1. How do I check Watchlist for each user in EZD?
   There is no watchlist in the EZD, because it is a proxy with no knowledge of what users request. Anything that is opened on the EZD is immediately passed upstream.

2. Why is there no IPC session in shared memory for my connection?
   If the datafeed application connects to EZD, but EZD is unable to connect to the ADS or proxy upstream, the downstream datafeed connection is disconnected. The event is logged within EZD, and the datafeed application is disconnected. EZD attempts to reconnect to the application.

   When a connection is established on both ends, there are two IPC sessions in shared memory:
   - The RIPCClient or SIPCClient object that represents the connection from EZD to datafeed client application (such as ADH). This contains the IP address of the connected application.
   - The HTTPSrcChannel object that represents the internet connection from EZD to upstream ADS.

   All shared memory statistics may be viewed in ezdmon -mob.

3. What services does EZD provide?
   The only service name that EZD provides is ELEKTRON_DD. However, this may change as EZD evolves. ELEKTRON_DD represents a 330ms trade-safe conflated service.

4. Why do I have to remap the user name to an account in the user_list.txt file?
   This logic is provided for applications that do not make user names configurable. The EZD maps the user name to a Stream ID account listed in the user_list.txt file. The account credentials are transmitted to the upstream ADS for permission credentials.

5. Why did my application get disconnected from EZD after it successfully connected?
   The most likely scenario is a problem with the internet connection. When the internet connection is lost, the EZD disconnects the downstream user as well. The EZD might also disconnect the downstream user if the application is slow reading or the LAN connection is bad.

   Contact the ISP to verify the health of the internet connection and ensure adequate bandwidth is available. A disconnect on market open or market close is typically indicative of bandwidth problems either within the internet or on the client site. Bandwidth through a corporate internet proxy is also known to have limits lower than what the internet connection might provide.

   The most common indication of outright network connectivity problems is if on reconnection, the EZD prints the following error:

   Failed to connect secure Channel to host hostname. Error:<Impl/ripcsrvr.c:xx>
   Error: xx ripcHostByName() failed. Hostname is incorrect. System errno: (24).

   This log message is a direct result of EZD calling gethostbyname for the hosts in hostList. If EZD is unable to obtain the IP address of the host after it was able to resolve on startup, then EZD cannot access the internet DNS, which is most likely due to loss of internet connection.
Appendix C  Configuration File

!
Elektron Zero Daemon (EZD) Sample configuration file
!

!!
!! Shared memory configuration
!!
maxSemaphores : 6
admin*maxClients : 10
ezd*admin*semaphoreKey : 90
ezd*admin*sharedMemoryKey : 90
ezd*admin*sharedMemorySize : 1000000
ezd*admin*enableChildStats : False
ezd*admin*enableRootStats : True
ezd*admin*enableMemoryStats : True

!!
!! Logging information
!!
*ezd*logger*file : ./ezd.log
*ezd*logger*install_file_action : True
*ezd*logger*install_stderr_action : False
*ezd*logger*install_system_action : False
*ezd*logger*logDtMsgTogether : False
*ezd*logger*max_bytes : 3000000
*ezd*logger*max_swapfiles : 0
*ezd*logger*microSecTimeStamps : true
*ezd*logger*selector : *.debug
*ezd*logger*timeStampFormat : %a %b %d %H:%M:%S %Y

!!
!! Tracing information
!!
*ezd*debugTrace : False
*ezd*dumpIncoming : False
*ezd*dumpOutgoing : False

!!
!! EZD program information
!!
*ezd*userFile : user_list.txt
*ezd*bindMainThread : P:0 C:0 T:0
!*ezd*libNameCrypto :
!*ezd*libNameOpenSSL :
*ezd*controllerAccountUUID: <machine account>
*ezd*controllerAccountPassword: <machine password>
*ezd*deviceChannelRetryTime: 15
*ezd*deviceChannelRetryHostListTime: 1200
!!
!! Dictionary Parameters
!!
*ezd*downloadDictionary : True
*ezd*enumFile : ../../etc/enumtype.def
*ezd*fieldDictionary : ../../etc/RDMFieldDictionary

!!
!! Failback Parameters
!!
*ezd*enableFailbackMgr : True
*ezd*failbackTime : 00:00
*ezd*failbackDayOfWeek : 0

!!
!! Client sink connection information
!!
*ezd*enableRsslServer : True
*ezd*snkRsslServer*rsslPort : 14002
*ezd*snkRsslServer*clientToServerPings : True
*ezd*snkRsslServer*sessionStatsWindow : 1
*ezd*snkRsslServer*guaranteedOutputBuffers : 200
*ezd*snkRsslServer*fdExhaustTimer : 60
*ezd*snkRsslServer*flushInterval : 20
*ezd*snkRsslServer*maxOutputBuffers : 400
*ezd*snkRsslServer*minPingTimeout : 6
*ezd*snkRsslServer*numInputBuffers : 10
*ezd*snkRsslServer*pingTimeout : 30
*ezd*snkRsslServer*reservedFileDescriptors : 64
*ezd*snkRsslServer*rsslTrace : False
*ezd*snkRsslServer*rsslConnectionTimeout : 60
*ezd*snkRsslServer*serverToClientPings : True
*ezd*snkRsslServer*timedWrites : True
*ezd*snkRsslServer*tcpNoDelay : False
*ezd*snkRsslServer*tcpRecvBufSize : 524288
*ezd*snkRsslServer*tcpSendBufSize : 524288
*ezd*snkRsslServer*compressionType : 0
*ezd*snkRsslServer*zlibCompressionLevel : 3
*ezd*snkRsslServer*forceRsslCompression : False
!end rsslServer params

*ezd*enableSslServer : True
*ezd*snkSslServer*sslPort : 8101
*ezd*snkSslServer*guaranteedOutputBuffers : 200
*ezd*snkSslServer*maxOutputBuffers : 400
*ezd*snkSslServer*numInputBuffers : 10
*ezd*snkSslServer*pingInterval : 20
*ezd*snkSslServer*pingKillInterval : 60
*ezd*snkSslServer*reservedFileDescriptors : 64
*ezd*snkSslServer*sessionStatsWindow : 1
*ezd*snkSslServer*allowCompMode : True
*ezd*snkSslServer*tcpRecvBufSize : 524288
*ezd*snkSslServer*tcpSendBufSize : 524288
*ezd*snkSslServer*compressionType : 0
*ezd*snkSslServer*zlibCompressionLevel : 5
!end sslServer params

!begin secure connection params
!!
!! Outbound Encrypted Internet connection information
!!
*ezd*secureConnection*hostList : amers1-ec-hmds.cp.thomsonreuters.com,
amers2-ec-hmds.cp.thomsonreuters.com
*ezd*secureConnection*port : 443
*ezd*secureConnection*proxyHostName :
*ezd*secureConnection*proxyPort :
*ezd*secureConnection*reencodeSrcDir : True
*ezd*secureConnection*IDN_RDF*clientServiceName: ELEKTRON_DD
*ezd*secureConnection*compressionType: 1
*ezd*secureConnection*guaranteedOutputBuffers : 200
*ezd*secureConnection*numInputBuffers : 10
*ezd*secureConnection*pingTimeout : 30
*ezd*secureConnection*tcpNoDelay : False
*ezd*secureConnection*tcpRecvBufSize : 2048000
*ezd*secureConnection*tcpSendBufSize : 2048000
*ezd*secureConnection*tryNextHostTime : 25
!end secure connection params
Appendix D  Sample Install Script for Linux

Loading Linux 6 binaries,..
1: Install EZD in default path at /opt/thomsonreuters/SOFTWARE/ezd1.3
2: Enter custom path
Where would you like to install EZD? 1

-----------------------------------------------------------------------------------------------
1: Install EZD CONFIGURATION FILE in default path at /opt/thomsonreuters/SOFTWARE/globalconfig
2: Enter custom path
Where would you like to install the configuration file? 1

-----------------------------------------------------------------------------------------------
We will now configure the list of users to connect to EZD.
Install will now prompt for User Name, Stream ID, and Stream ID Password
User Name : test1
Stream ID : test1@thomsonreuters.com
Stream ID Password: testpass
user test1 added.

-----------------------------------------------------------------------------------------------
Would you like to add any more users? [y/n] n

-----------------------------------------------------------------------------------------------
ezd1.3 requires your EZD Credentials.
Enter your EZD Credentials:
Elektron Real Time ID: test_machine
Elektron Real Time Password: test_machine

-----------------------------------------------------------------------------------------------
Enter the number for which region EZD will be connecting to

====Hosts for Secure Internet Delivery =====
1: EMEA (emea1-ec-hmds.cp.thomsonreuters.com, emea2-ec-hmds.cp.thomsonreuters.com)
2: APAC (apac1-ec-hmds.cp.thomsonreuters.com, apac2-ec-hmds.cp.thomsonreuters.com)
3: AMERS (amers1-ec-hmds.cp.thomsonreuters.com, amers2-ec-hmds.cp.thomsonreuters.com)

====Hosts for Delivery Direct over leased lines =====
4: EMEA [emea1-ec-hmds.cp.extranet.thomsonreuters.biz, emea2-ec-hmds.cp.extranet.thomsonreuters.biz]
5: APAC [apac1-ec-hmds.cp.extranet.thomsonreuters.biz, apac2-ec-hmds.cp.extranet.thomsonreuters.biz]
6: AMERS [amers1-ec-hmds.cp.extranet.thomsonreuters.biz, amers2-ec-hmds.cp.extranet.thomsonreuters.biz]

7: Custom Host
hostList: 1

-----------------------------------------------------------------------------------------------
Will you be routing through a proxy? [y/n] y

-----------------------------------------------------------------------------------------------
Please enter the proxy host name: test.proxy.thomsonreuters.com

-----------------------------------------------------------------------------------------------
Please enter the proxy port: 443
1: SSL
2: RSSL
3: SSL & RSSL
Would you like to enable SSL, RSSL, or both? [enter 1, 2, or 3] 3

Enable failback for ezd? [y/n]
Default is [y]
Enable failback: y

Please set the failback day of the week. Default is [8]
0: Sunday
1: Monday
2: Tuesday
3: Wednesday
4: Thursday
5: Friday
6: Saturday
8: Everyday
Day of the week: 0

Please enter the failback time in 24hr format. Default is 12:00AM
NOTE: The time should be GMT
ex) 00:00
Failback time: 00:00

WARNING: ezd.cnf already exists in /opt/thomsonreuters/SOFTWARE/globalconfig/
WARNING: Renaming current ezd.cnf to ezd.cnf.old
Installed in EZD in /opt/thomsonreuters/SOFTWARE/ezd1.3
Installed EZD configuration file in /opt/thomsonreuters/SOFTWARE/globalconfig
Installation complete!
Appendix E  Configuration Example

Figure 15 illustrates how the EZD server is connected to an application server and the Refinitiv servers.

Figure 15. EZD Sample Configuration
Appendix F  EZD User Password Management Utility

EZD 1.3 is packaged with a password management utility to help facilitate changing user passwords.

F.1 Executing the Password Management Utility

F.1.1 Linux

Execute the password management utility on Linux by entering the following command:

```
java -jar ezd_userlist.jar
```

F.1.2 Windows

Execute the password management utility in Windows by executing the `ezd_userlist.exe` binary.

F.2 Configure the EZD Path

The utility needs a path to the EZD binary file in order to encrypt the password in the user configuration file.

▸ Configure the path to the EZD binary file by performing the following steps:

1. Click on Browse Path. A file manager window displays.
2. Select the EZD binary file `ezd.exe` (for Windows) or `ezd` (for Linux).
3. Click Open.

F.3 Using the Password Management Utility

F.3.1 Open an Existing File

▸ Open an existing file by performing the following steps:

1. Click on File and select Open from the drop-down list. A file manager window displays.
2. Navigate to and select the existing file.
3. Click Open.

F.3.2 Add a user:

▸ Add a user by performing the following steps:

1. Enter the user name in the User Name: box.
2. Enter the user Stream ID in the Stream ID: box.
3. Enter the user Stream ID password in plain text in the Stream ID Password: box.
4. Press the Add User button.

F.3.3 Change an Existing Password

▸ Change an existing password by performing the following steps:

1. Select the user in the list and click Change Password. A dialog window opens and prompts you for the new password.
2. Enter the new password in plain text.
F.3.4 **Save Data**

Changes in the user configuration file are saved by clicking the **Save Data** button or by selecting **Save** in the **File** drop-down menu.

![User Password Management Utility](image)

**Figure 16. User Password Management Utility**
Displaying the Password Utility from Remote Systems

If you are on a Linux machine and remotely configuring your EZD installation on another Linux machine, you can have the EZD password utility appear on your local machine.

Display the EZD password utility on your local machine by performing the following steps:

1. Execute `xhost +` on your local machine.
2. On the remote machine, export the DISPLAY variable to the local machine's display. Example:

```bash
export DISPLAY=localmachine:0
```
Appendix G  EZD Monitoring Tool - ezdmon

G.1  Introducing EZD Monitoring Tool

ezdmon is a utility that displays configuration and statistics information that the EZD application publishes to shared memory. ezdmon is located in the same directory where the EZD application is installed.

G.2  Using the EZD Monitoring Tool

G.2.1  Launch ezdmon

Launch ezdmon by performing the following steps:

1. Set the environment variable RMDS_CONFIG to the path of the EZD configuration file.
2. Perform the one of the following as applicable:
   - Linux: Default directory: /opt/thomsonreuters/SOFTWARE/ezd1.3/rhel6_x86_64/bin
     From a command line, navigate to the location of the ezdmon binary and enter the following command:

     ```bash
     ./ezdmon
     ```
   - Windows: Default directory:C:\ThomsonReuters\ezd1.3\win_x86_64\bin.
     Open a DOS prompt window, navigate to the location of the ezdmon binary, and enter the following command:

     ```bash
     .\ezdmon
     ```
**Example of Using EZD Monitoring Tool**

*ezdmon* uses vim-like navigation, so the arrow keys are not used to navigate. Navigation is done with H (left), K (up), J (down), L (right), Esc (previous menu), CTRL-H (backspace). For example, to change *FailbackAllToPrimaryHosts* under HTTPSrcChannel, perform the following steps:

1. Launch *ezdmon* by according to the procedure in **Section G.2.1**. The *ezdmon* window displays.
2. Press ENTER to view the **Browse Object Directory** screen, and then press TAB to navigate to the **Classes in System** section.

![Classes in System](image)

3. Press the J key five times, then press ENTER to view the **Instances of class HTTP Tunnel Manager** screen.

![Instances of class "HTTP Tunnel Manager"](image)
4. Press ENTER to select the **HTTP Tunnel Manager** instance.

![Instance Display]

- **Parent Instance:** caksxvnhost5-vmi.1.csd
- **Name:** HTTP Tunnel Manager
- **Class:** HTTP Tunnel Manager
- **State:** Init <Init> ""

**Variables**

- LoginAttempts: 17
- LoginRejects: 49414
- totalClientLostM: 0
- totalClientTOM: 8
- pingInterval: 0
- pingKillInterval: 0

**Children**

- registered2Users
- secureChannel.17
- secureChannel.14

5. Press **J** to navigate to the lowest-numbered **secureChannel** in the **Children** section and press ENTER. In this example, **secureChannel.14** is the lowest-numbered.

![Instance Display]

- **Parent Instance:** caksxvnhost5-vmi.1.csd.HTTP Tunnel Manager
- **Name:** secureChannel.14
- **Class:** HTTP Src Channel
- **State:** Init <Init> ""

**Variables**

- isOnPrimaryHost: True
- FailbackToPrimary: False
- clientLostMsg: 0
- resBias: 60
- clientIOR=resP=qu: 0

**Children**
6. Press the space bar to navigate to the screen where **FailbackToPrimar** is seen. Use the J key to select **FailbackToPrimar** and press ENTER.

7. Press CTRL + U to delete the text **False**, and then type **True** and press ENTER.
8. To exit `ezdmon`, press the Esc several times to navigate back through the menu tree until you return to the command prompt.