TECHNICAL SERVICE MANAGEMENT

FTSE Distribution Interface

Implementation Guide – Java (Unix)

API v2.0.0

FTSE

FTSE

Contents

INTRODUCTION

This document forms an implementation guide for developers utilising the **FTSE Distribution Interface (FDI)**.

Developer tips and hints have been provided on core areas of the service and of specific data handling considerations.

It also provides details of how to compile and utilise the Java (Unix) sample applications which are components included in the FDI API.

This document should be read in conjunction with the following FTSE technical documentation

- FTSE Distribution Interface Overview
- FTSE Distribution Interface Conformance Testing

A Java (Windows) version of the API is also available for which the following documentation exists:

- FTSE Distribution Interface Implementation Guide – Java (Windows)

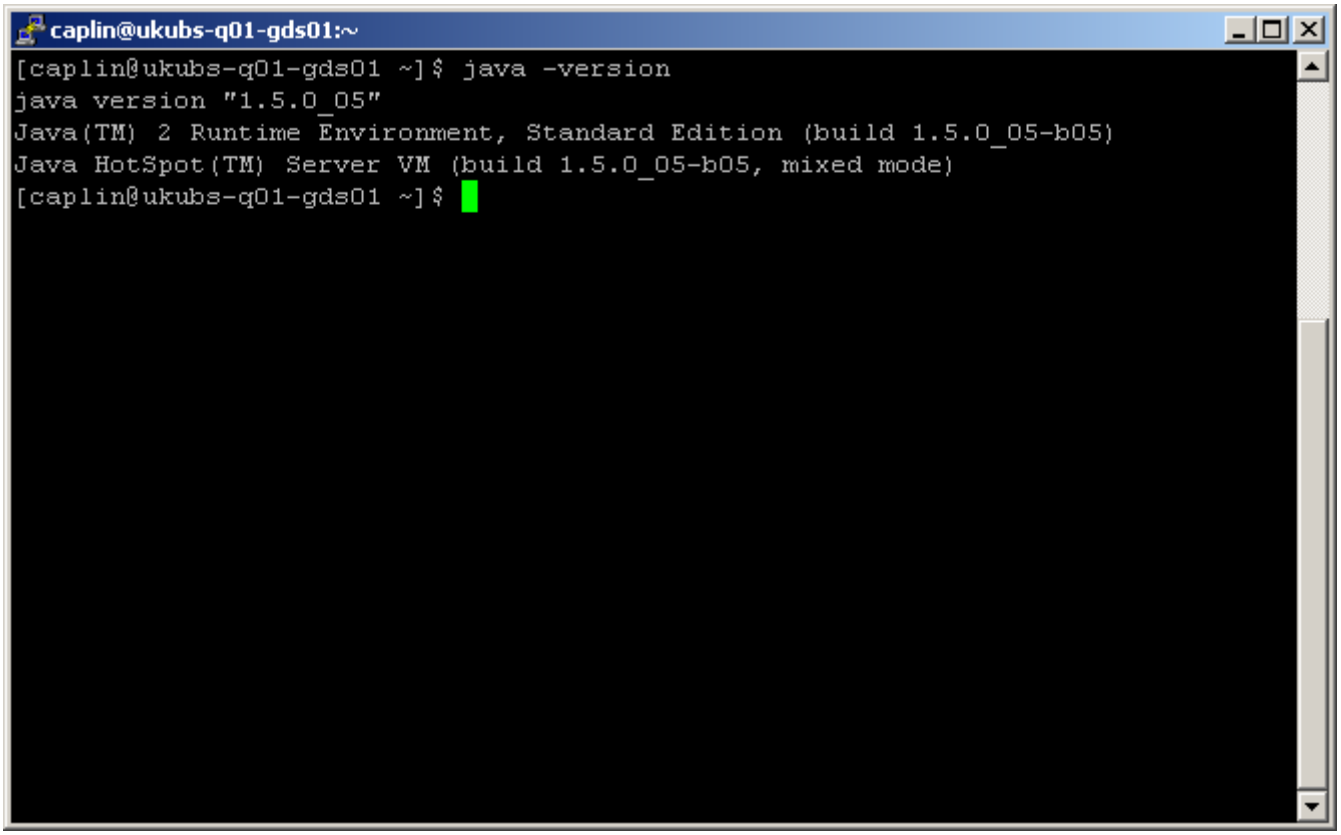
A Microsoft .NET version of the API is also available for which the following documentation exists:

- FTSE Distribution Interface Implementation Guide – . NET

RUNTIME & DEVELOPMENT ENVIRONMENT

Before progressing open a command line session and check the Java runtime environment and compiler on the computer ***[java –version].***

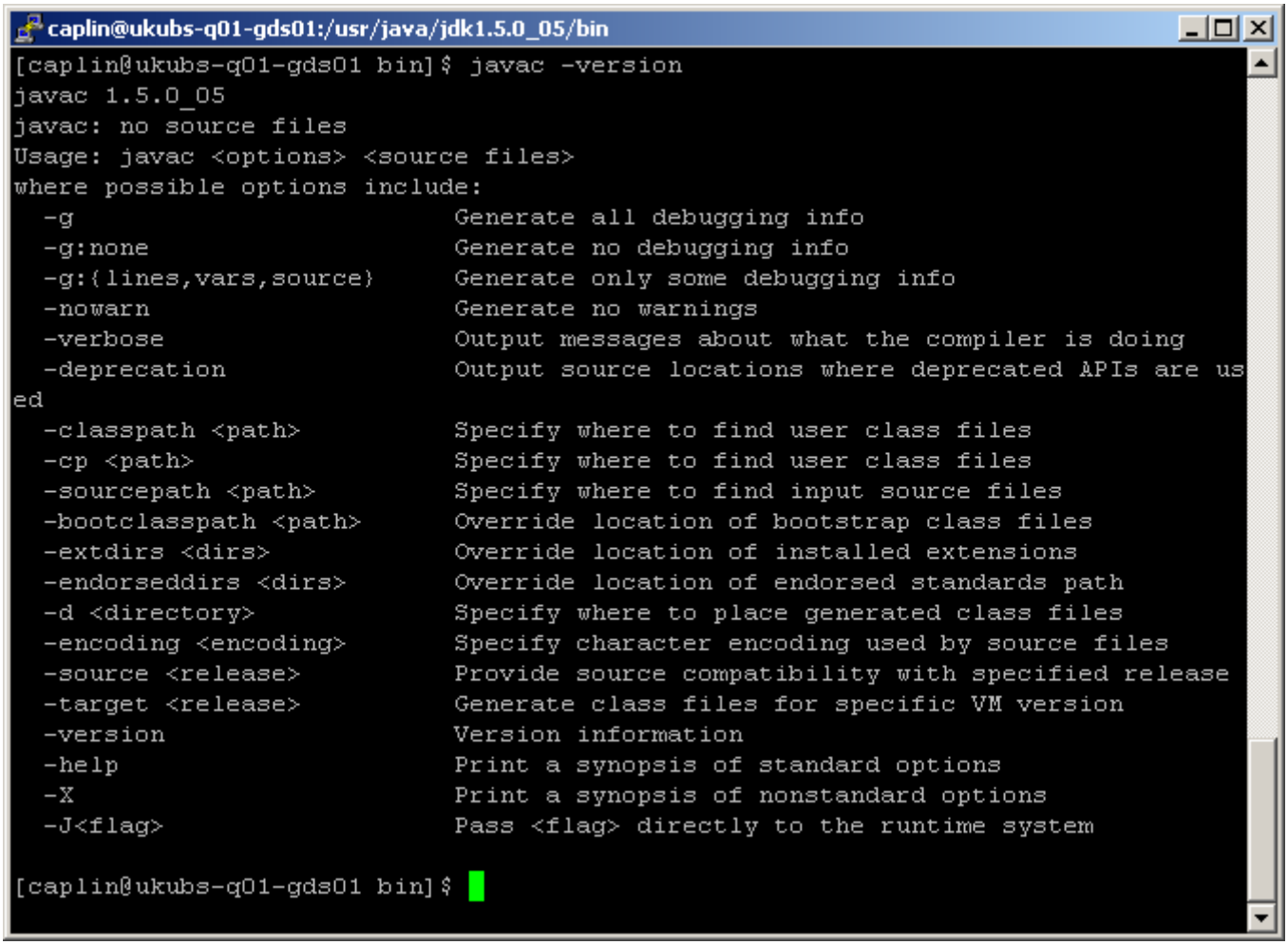
The software API is designed to run Java at a minimum of v5.0, i.e. 1.5.0_xxx as reported.



CLIENT DEVELOPMENT

If you want to develop applications then the Java compiler will need to be present.

This can be checked for by running the following from the command prompt [***javac –version***] as per example below.



If required the JSDK can be downloaded from the following URL:

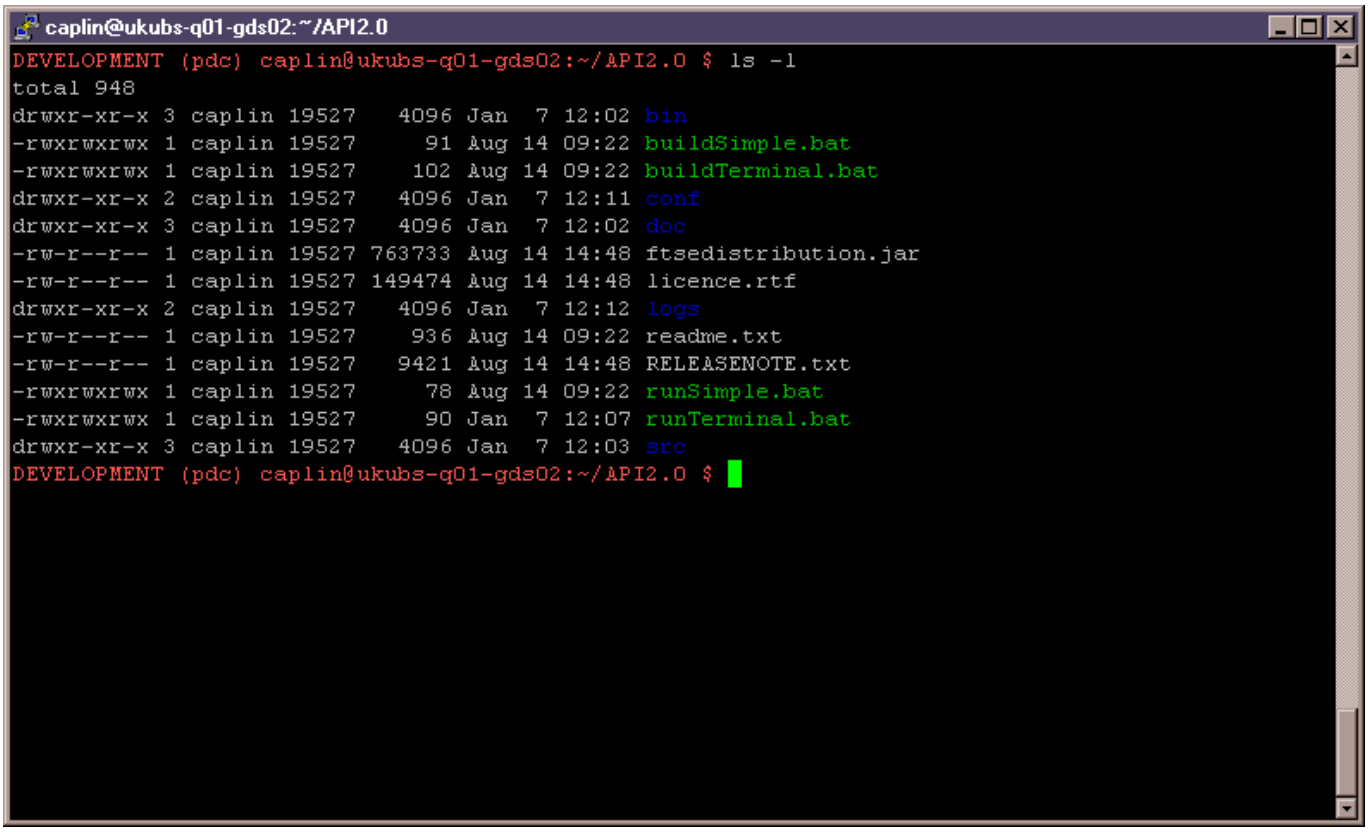
http://www.oracle.com/technetwork/java/javase/downloads/index-jdk5-jsp-142662.html

DOWNLOADING THE API

Please contact info@ftse.com for the latest edition of the API.

The API is distributed as a zip file.

In this example the FDI has been unpacked to directory ***API2.0***



NOTE: It would be advisable to remove any previously installed API before commencing installing the latest version.

The key files in the FDI directory are:

buildSimple.bat	Batch file to build the Simple java application*
buildTerminal.bat	Batch file to build the Terminal java application*
ftsedistribution.jar	The FDI jar
runSimple.bat	Batch file to run the Simple application**
runTerminal.bat	Batch file to run the Terminal application**

*

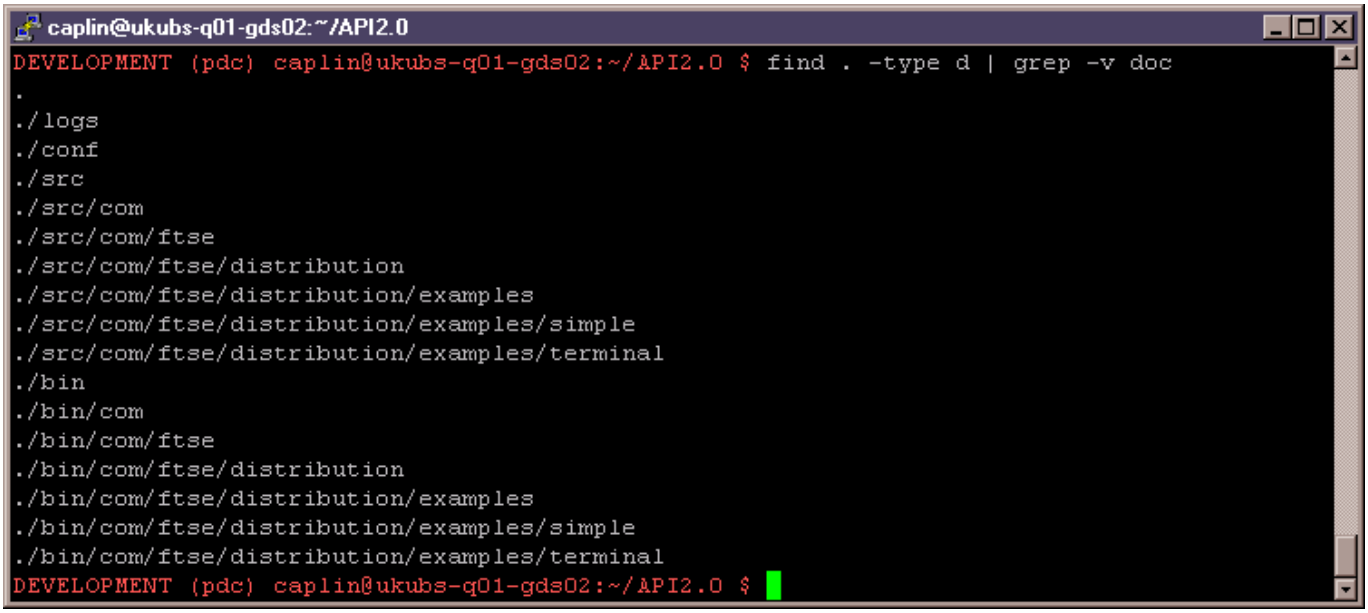
NOTE: These files will need to have the path separator changed from DOS "\" to Unix "/".

**

NOTE: These files will need to have the semi-colon ";" replaced with a colon ":" and execute permissions added

LOCATION OF INSTALLED SOFTWARE

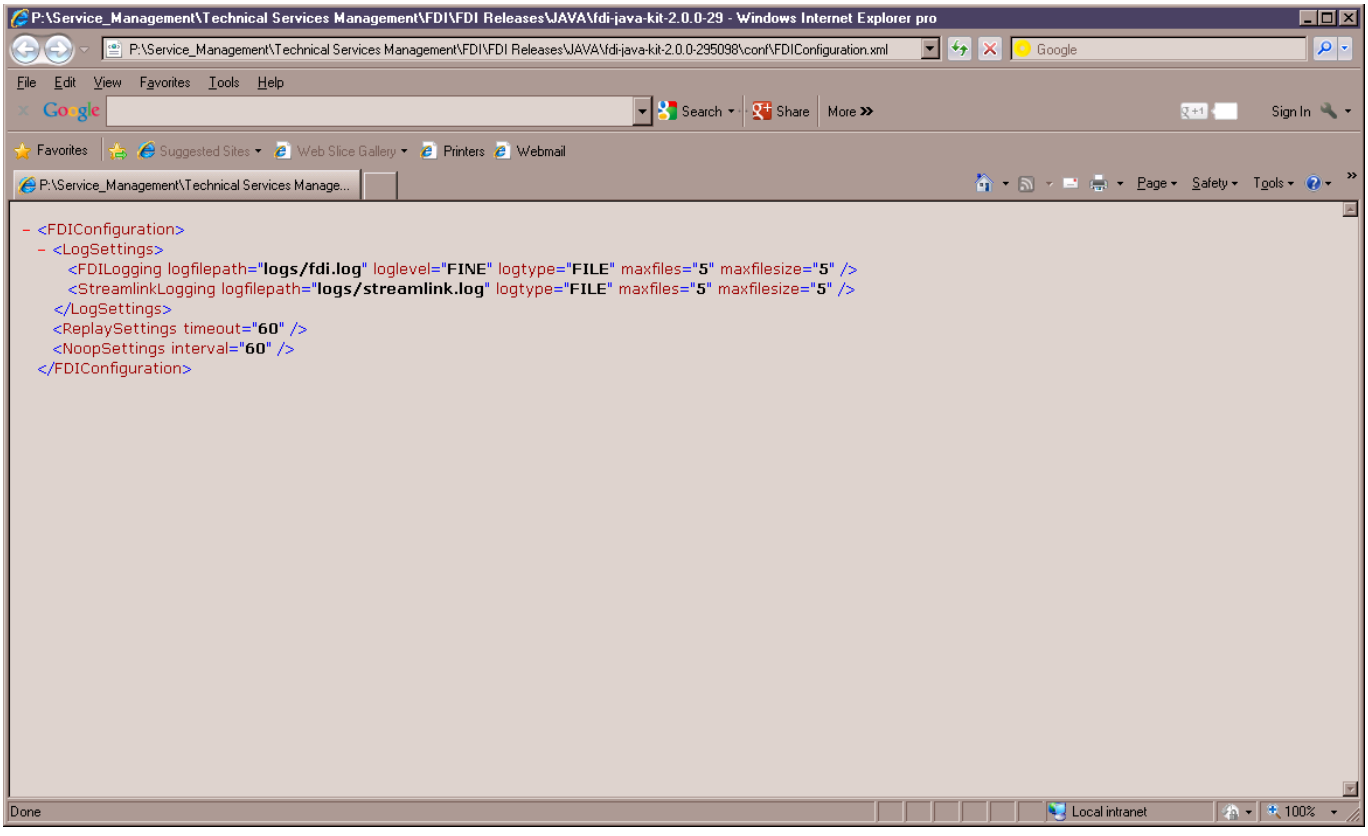
The unpacked directory structure is as follows (excluding documentation):



The ***conf*** folder contains two xml configuration files.

FDICONFIGURATION.XML

This configuration file helps to define the behaviour of the logging and by default it looks like this:



By sticking to the default path and filenames logs are generated in the ***logs*** directory relative location to where the API resides.

fdi.log and ***streamlink.log*** size and version limits are controlled by altering some of the parameters, for example:

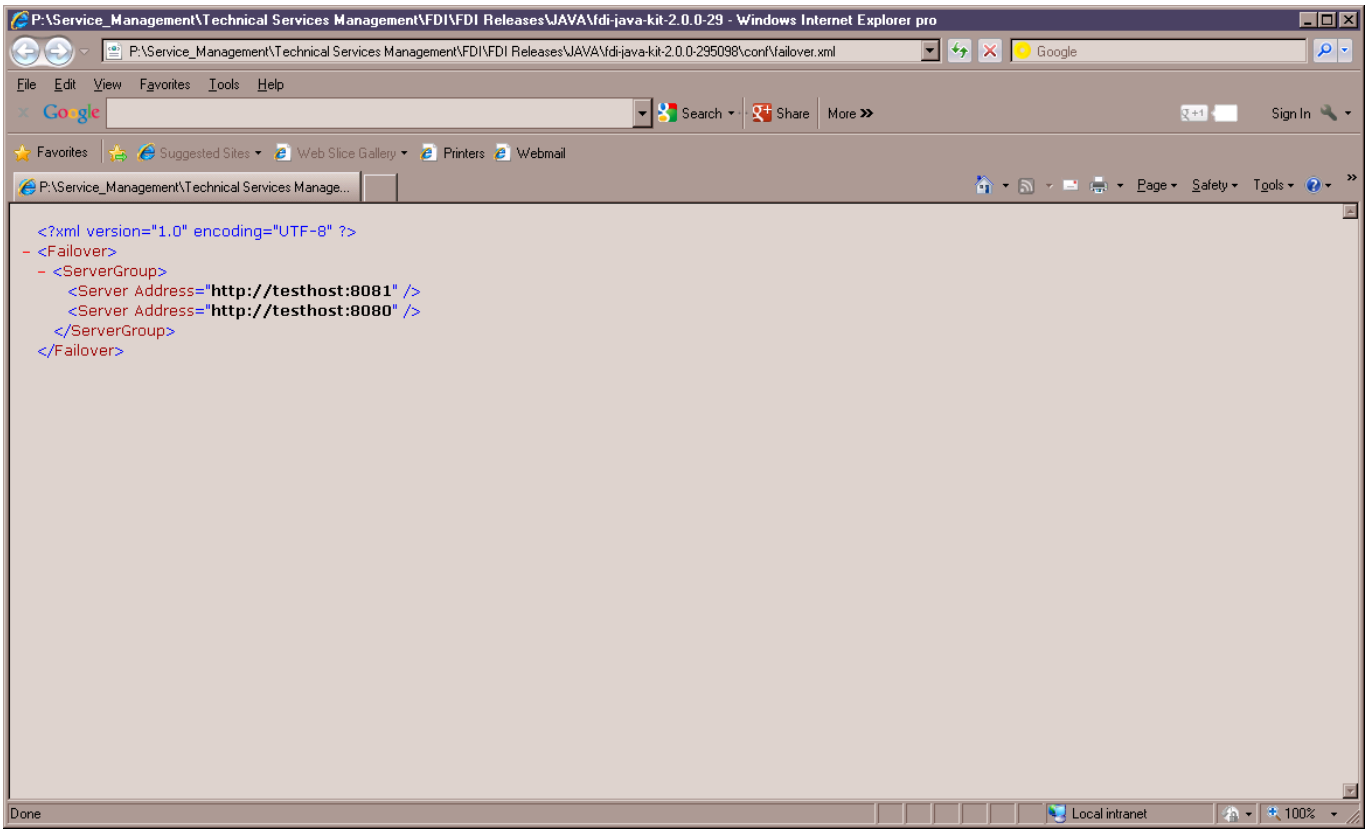
```
<FDIConfiguration>
  <LogSettings>
    <FDILogging logfilepath="logs/fdi.log" logtype="FILE" maxfiles="10" maxfilesize="25"/>
    <StreamlinkLogging logfilepath="logs/streamlink.log" logtype="FILE" maxfiles="10" maxfilesize="25"/>
  </LogSettings>

  <NoopSettings interval="60" timeout="60" />
  <ReplaySettings timeout="60" />
</FDIConfiguration>
```

The above would limit each of the log files to a maximum size of 25Mb before a new log file is created up to a maximum of 10 log files, at which point the initial log will be overwritten.

FAILOVER.XML

This configuration file is optional, but is needed if the "FTSE API failover" functionality is required.

This file resides in the same location as the ***FDIConfiguration.xml*** file.



In the above example, the address could be an IP address or a FQD name.

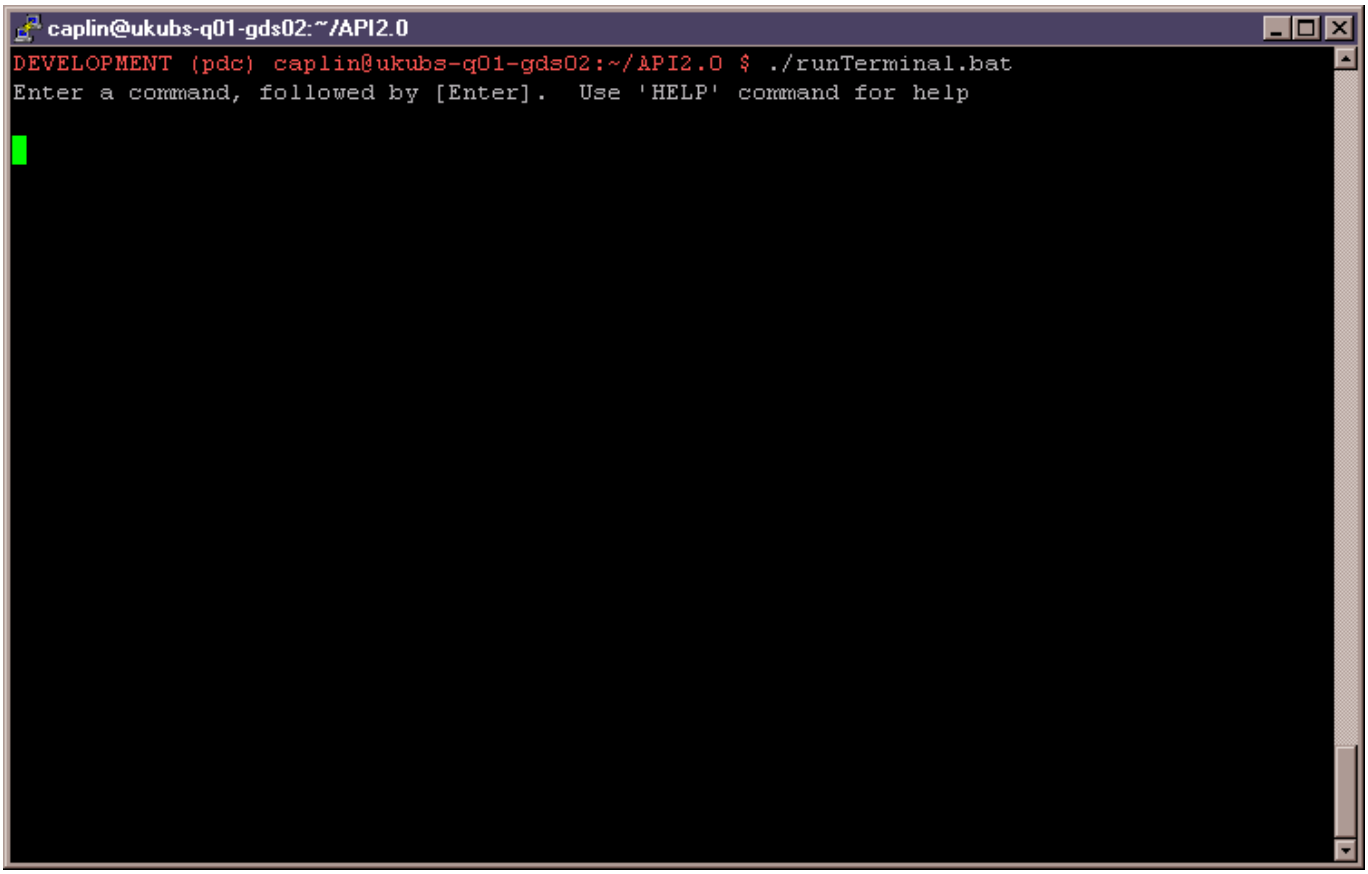
NOTE: For external client access to FTSE Production/DR platforms only https on port 443 is used.

CONNECTING TO THE FTSE TEST PLATFORM

CONNECTION

To connect to the Test Platform using the standard API tool execute the sample Terminal application via the runTerminal batch file.

NOTE: In the original file the semi colon ";" was replaced with a colon ":"



From the command line two methods exist to connect to the service.

The first and standard method is via a simple login request, syntax as follows:

login <hostname> <port number> <login name> <password>

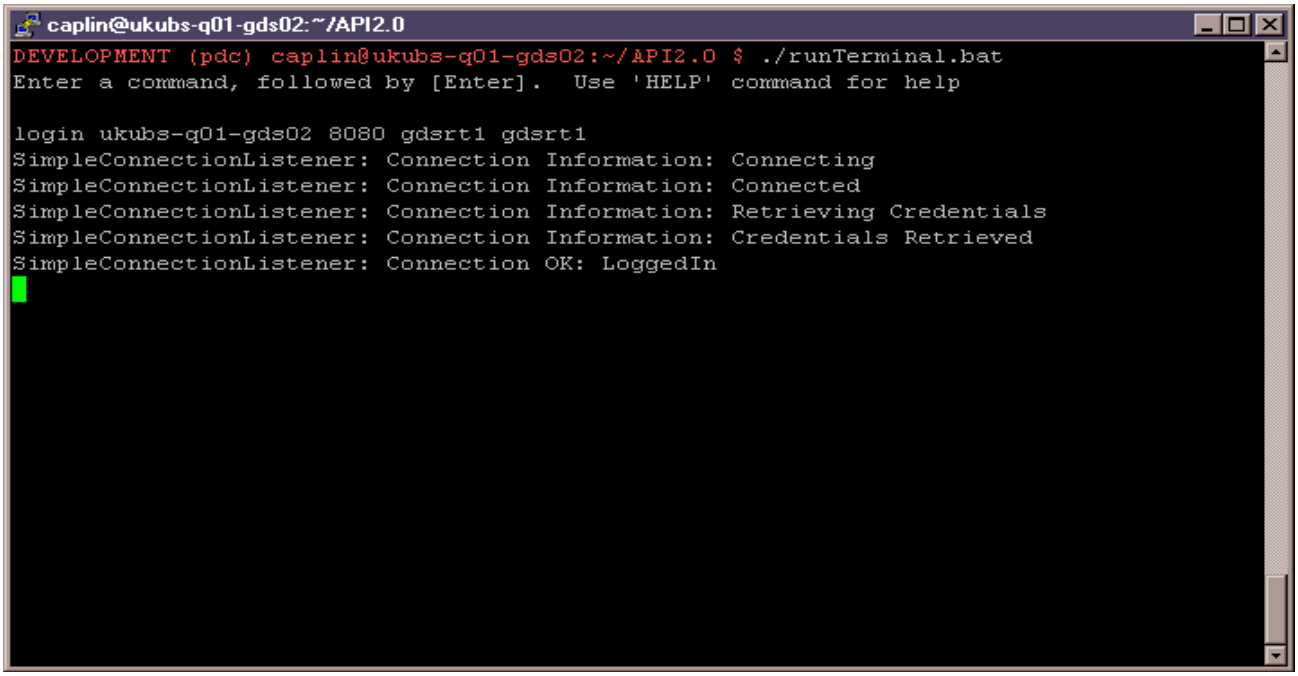
The second method is invoked via the built in API failover, syntax as follows:

loginfailover <username> <password> <configURL>

In the following example the standard login method is used.

NOTE: For an external connection the following command should be input before login:

setprotocol https

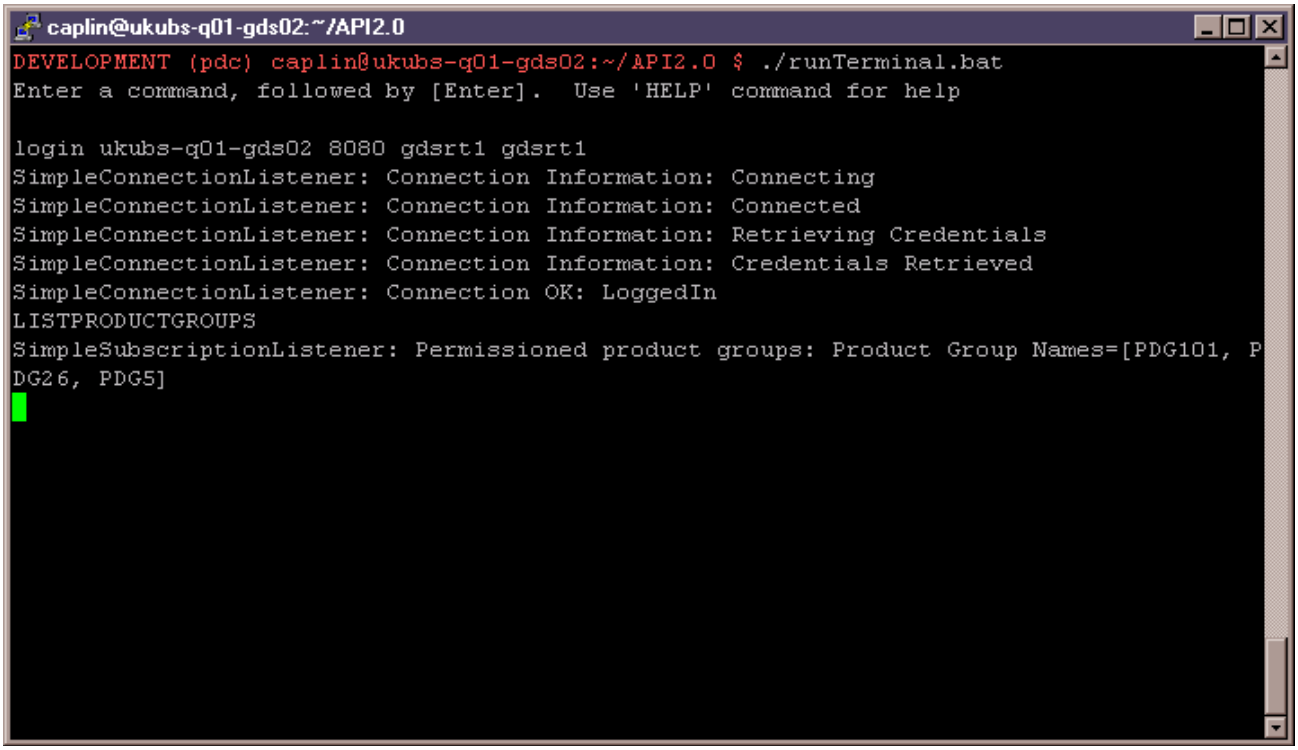


NOTE: FTSE support will provide individual clients with suitable login parameters on request.

Successful connectivity is achieved when the "***LoggedIn***" message is observed.

ENTITLEMENT & SUBSCRIPTION

To confirm entitlement run the ***listproductgroups*** command from within the Terminal example.

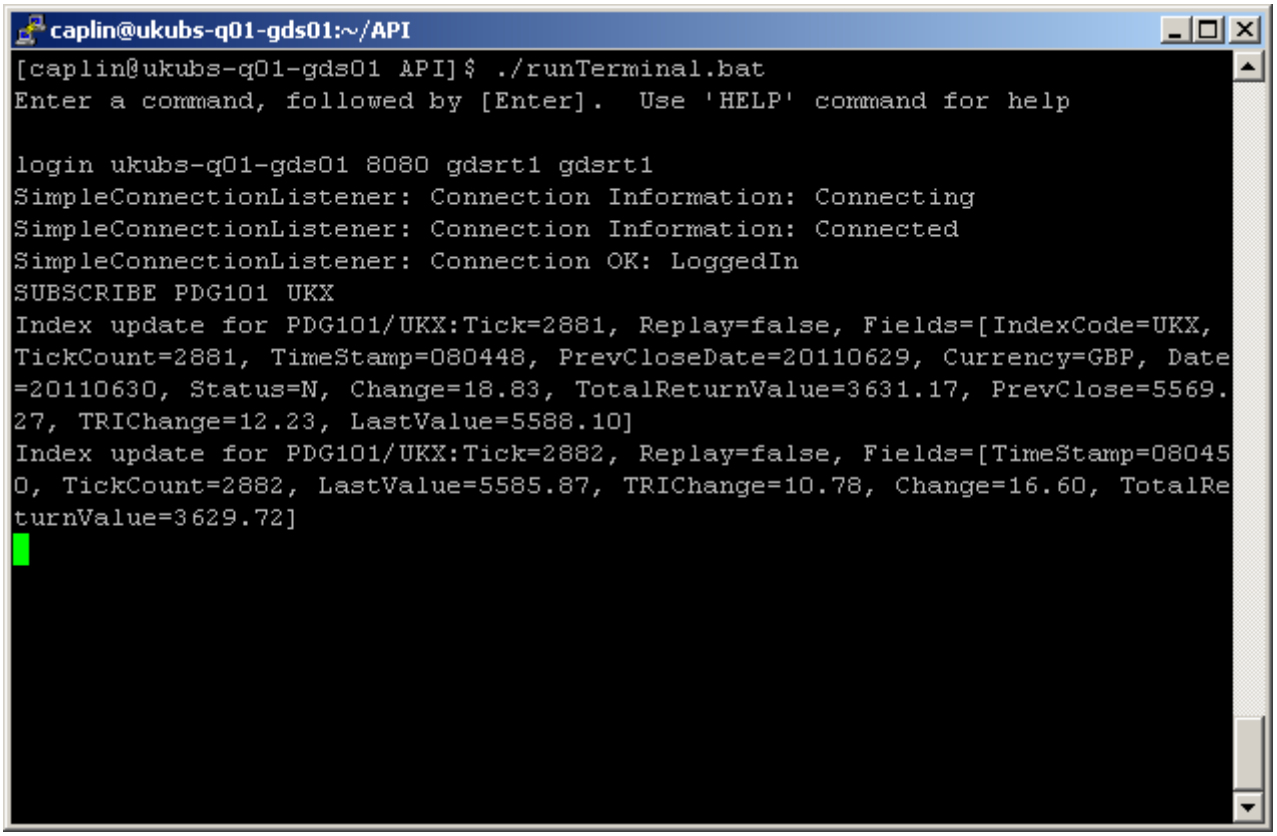


FTSE provide permission at the Product Data Group (PDG) level, therefore by definition if permission is granted to the PDG all indices contained therein are granted.

Subscriptions can be achieved at an individual index level or at the PDG level. At a PDG level all updates for all indices contained therein will be provided, at the individual index subscription only updates for that index will be provided.

For an individual index the syntax is: ***SUBSCRIBE <PDG> INDEX***

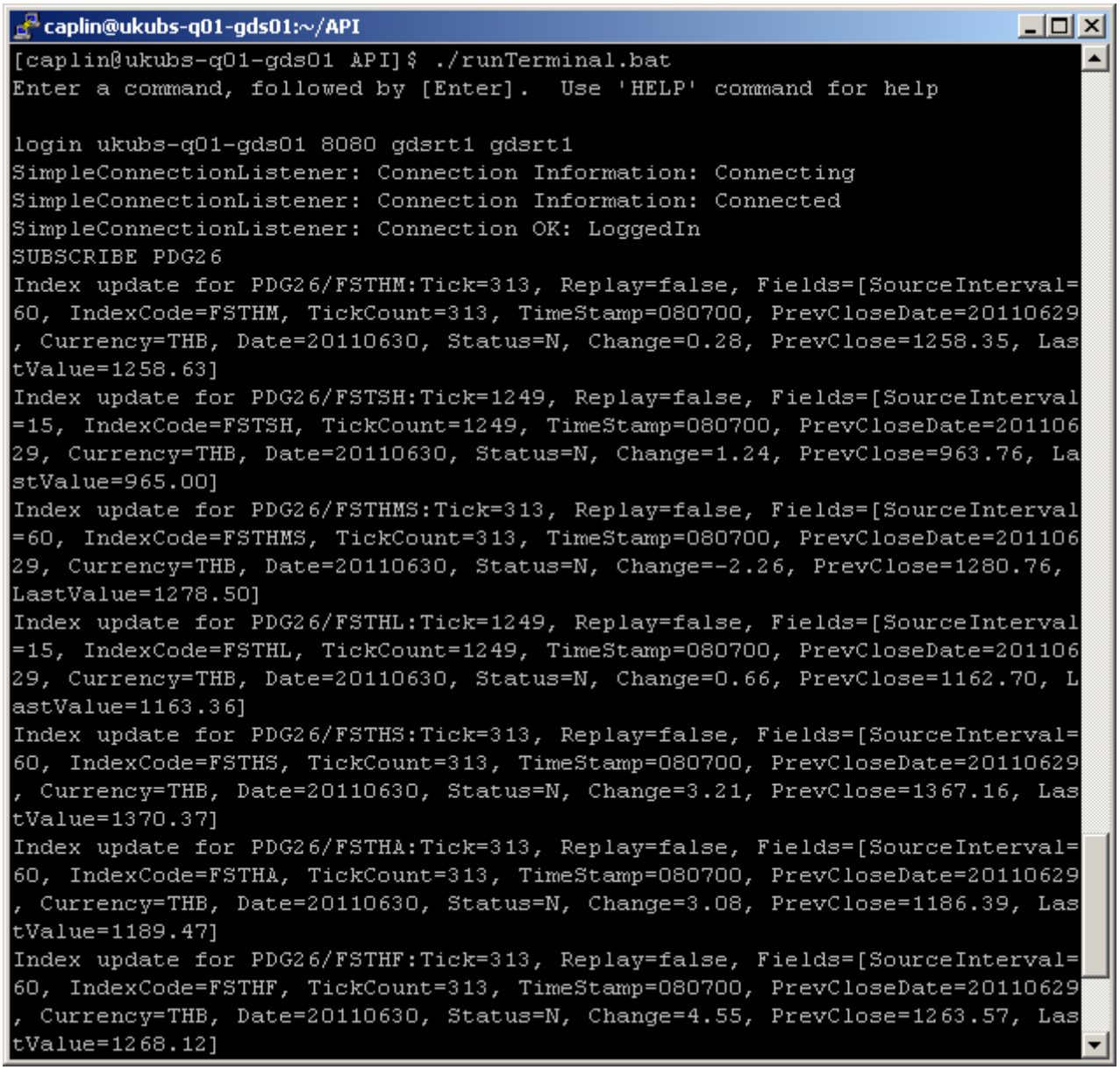
e.g. ***SUBSCRIBE PDG101 UKX*** (as in the example below)



NOTE : Example screenshot from prior API version

Subscribing to a PDG is simpler, the syntax is : ***SUBSCRIBE PDG***

e.g. ***SUBSCRIBE PDG26***



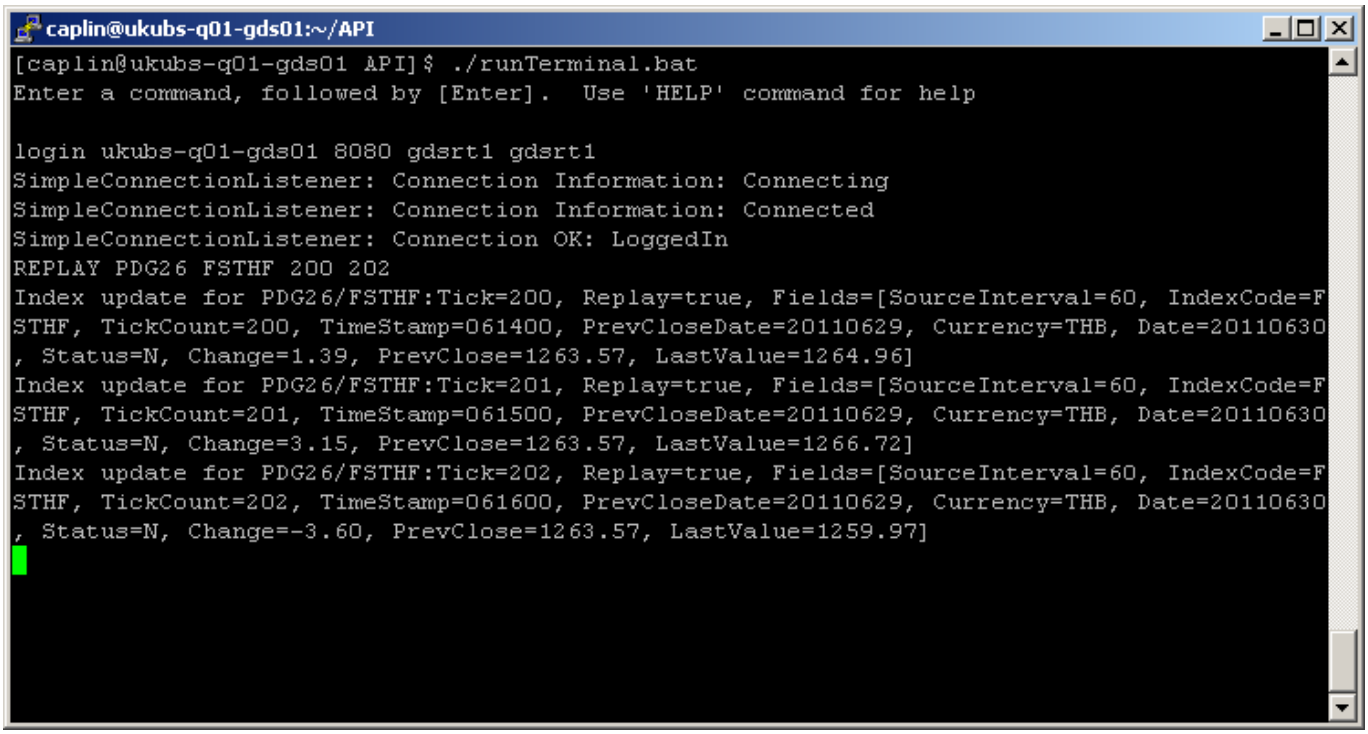
NOTE : Example screenshot from prior API version

Log out from the session by typing ***logout*** at the prompt.

REPLAY

It is possible to request a replay of data for Pulse indices only. The replay request itself must be made at the index level and is qualified by the TickCount.

NOTE: Replays are only available in the current trade day and are not supported across day boundaries.

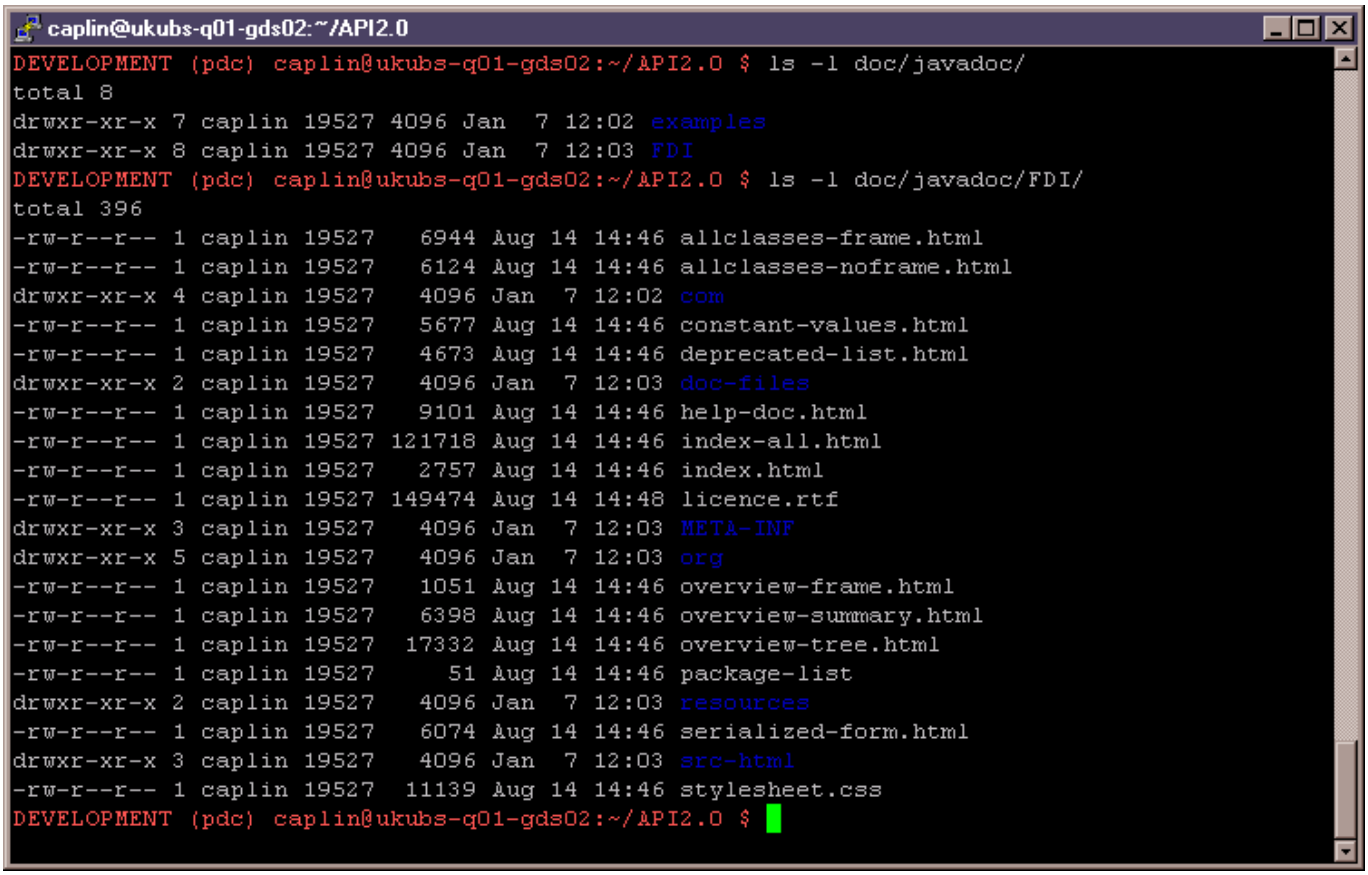


NOTE : Example screenshot from prior API version

DOCUMENTATION

The documentation can be accessed via a Browser from the ***/doc/javadoc*** directory.

The starting point for navigation is ***index.html***



CONNECTIONS VIA HTTPS

The Production environment ONLY support HTTPS connections and clients are encouraged to configure their systems for HTTPS as soon as possible.

In order to run an encrypted secure socket connection the FTSE service needs to be set up as a known and certified application.

If required contact info@ftse.com for a copy of the certification file which will need to be added to the local certificates file on the computer where the API is installed.

TROUBLESHOOTING

UNABLE TO RUN THE JAVA COMPILER.

Check the install details for your selected development environment, in particular check that the PATH settings are correct for ***javac***.

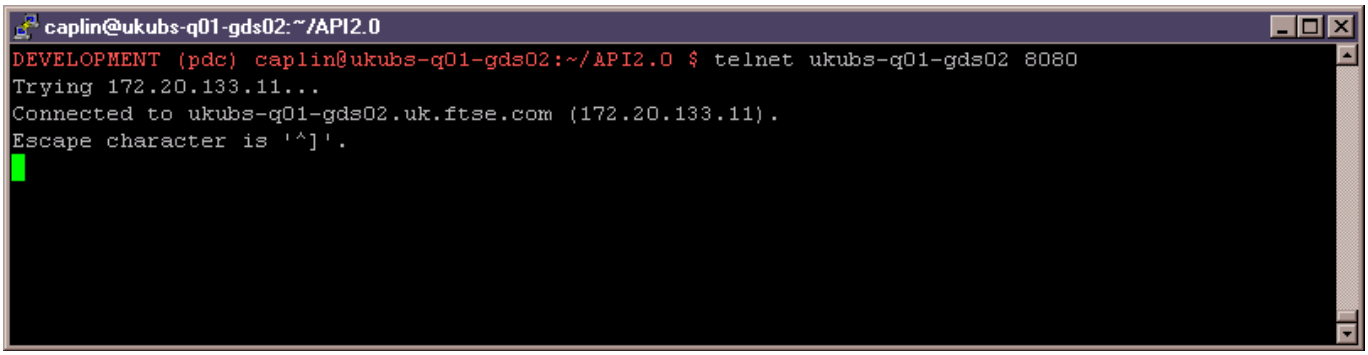
NO CONNECTION

In the first instance it is advised to check the connectivity between your client and the FTSE server.

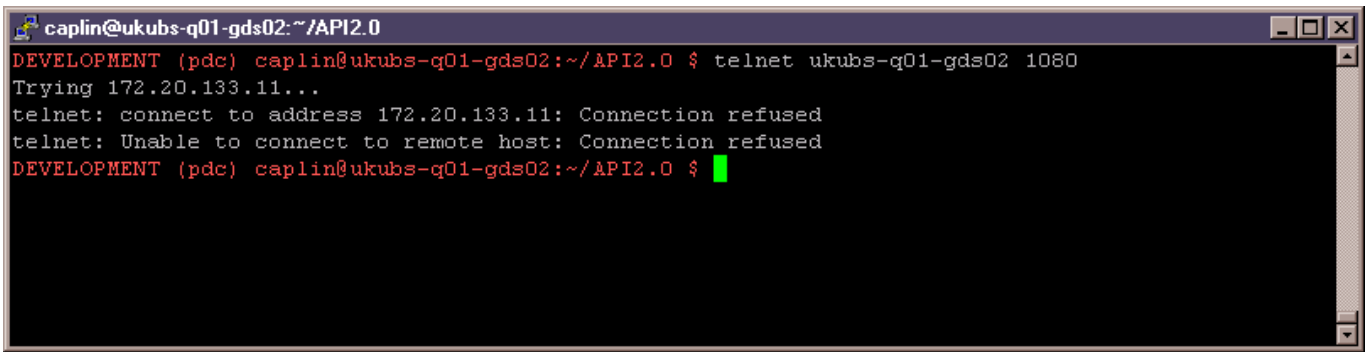
testgds.ftse.com is the name the Test system is known to the internet.

telnet is the advised connectivity check method which must be qualified by the GDS port. Externally this will typically be **443**.

Successful telnet example (internal connection hence port is not 443):



Unsuccessful telnet example :

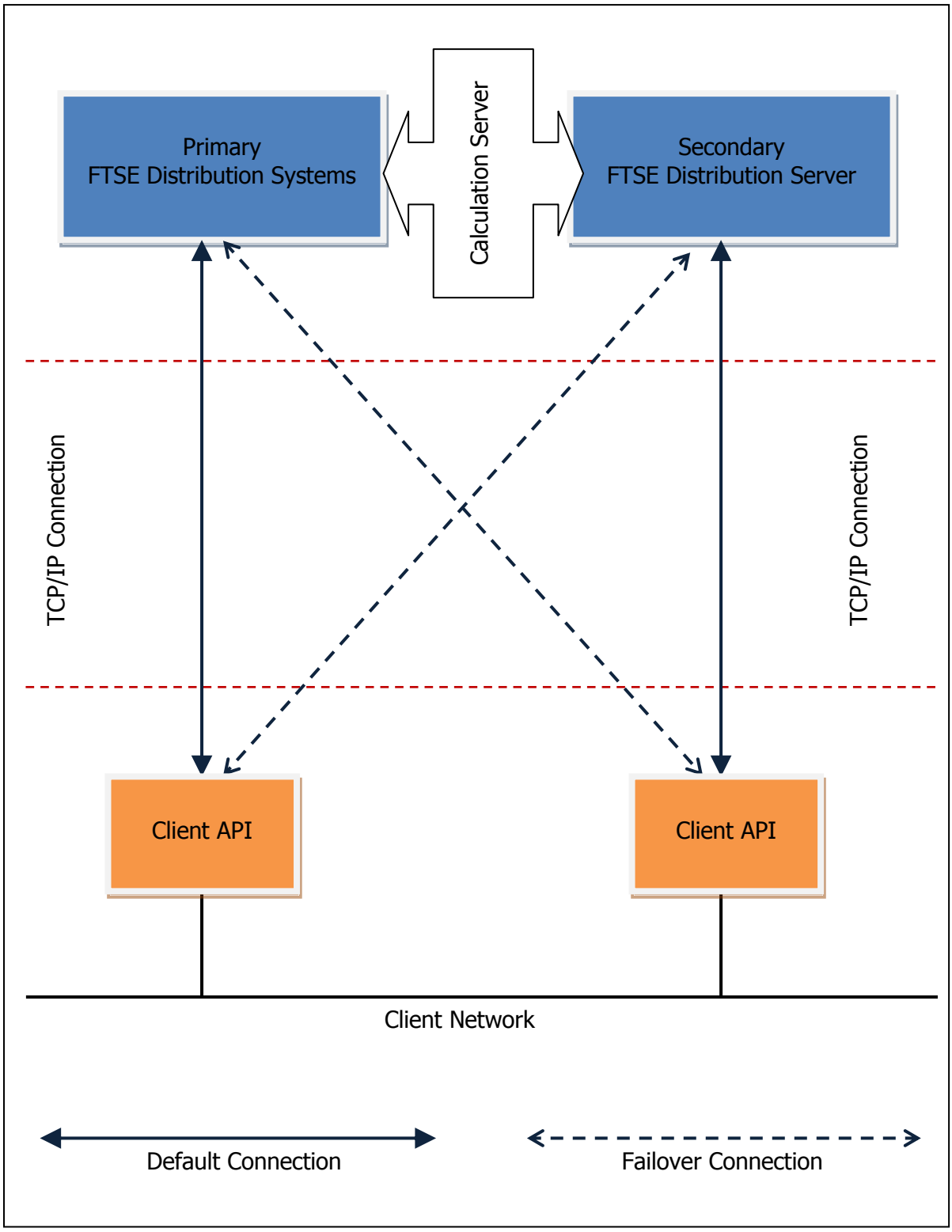


USER UNKNOWN

Confirm that the username and password are accurate; these will have been provided by FTSE previously.

TIPS & HINTS
SERVICE RESILIENCE

The recommended standard distribution architecture is outlined below. This offers unbroken service in the event of any single FTSE Distribution Server or single client API node losing connectivity.



It should be noted that there are no differences between either the service availability or content as delivered through either the Primary or Secondary Distribution Servers and both can be considered Production class nodes.

They are fed independently with the same content from the Live side of our Calculation Server.

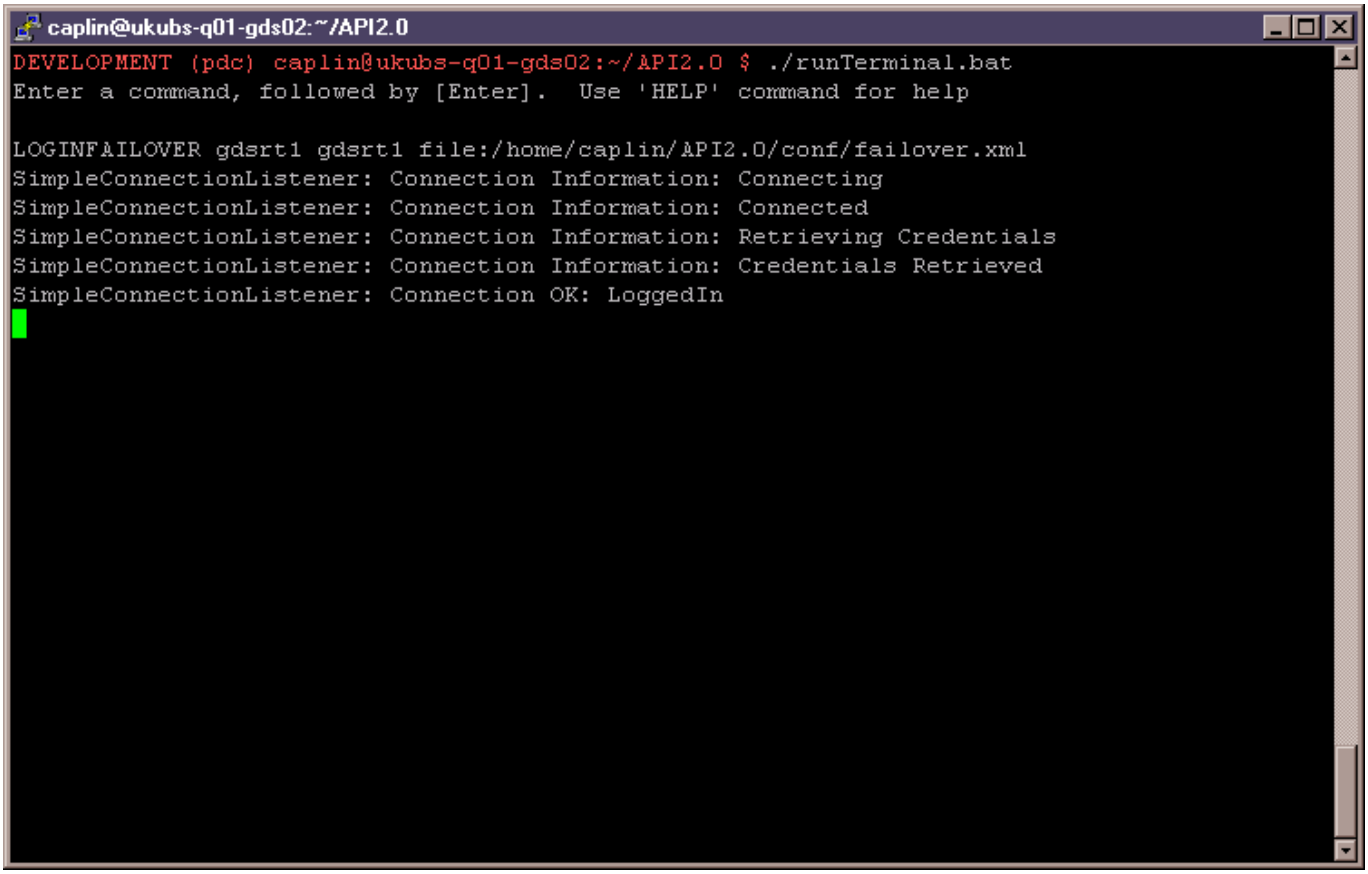
SERVICE FAILOVER

Instead of connecting just to a single named service the API supports automatic failover to a second service which is recommended.

The connection alternatives are defined in an XML file which itself can be referenced in the configuration parameters of the login method. When used in this way the host and port parameters are defined by the XML file and override any passed explicitly to the login method.

Should a disconnection occur, for any reason, the API itself will handle the login and re-subscription* automatically when utilising this functionality.

* if an automatic reconnection is achieved subscriptions will be maintained however clients are advised to check for re-subscription success in the event that an immediate reconnection has not been possible which could jeopardise subscription objects on the platform.



OPTIMISATION

The IndexUpdate method only returns changed fields, i.e. fields which have not changed are optimised out of the distributed message.

For example at start-up the IndexEvent Method returns data for the following

SUBSCRIBE PDG101 UKX
Index update for PDG1/UKX:Tick=1730, Replay=false, Fields=[SourceInterval=15, IndexCode=UKX, TickCount=1730, TimeStamp=151215, PrevCloseDate=20090313, Currency=GBP, Date=20090316, Status=N, Change=79.13, PrevClose=3753.68, LastValue=3832.81]

Whereas subsequent messages for the same PDG/index/session would feature a stripped down message:

Index update for PDG1/UKX:Tick=1731, Replay=false, Fields=[TimeStamp=151230, TickCount=1731, LastValue=3833.01, Change=79.33]

This field optimisation will exist across trade date boundaries as well intraday sequential messages. Clients need to anticipate optimisation in their market logic.

TICKCOUNTS

The TickCount increments on all iterations of an index sent to the FDI under normal operational conditions.

NOTE: There is a known (though rare) scenario where the TickCount for an index can increment by more than one or regress between consecutive messages.

For Pulse Indices the updates will be sent at regular intervals. If the LastValue itself has not changed between messages then the TickCount will still increase by one and the TimeStamp field will update (please see Optimisation).

For Streaming Indices a new TickCount is generated for every calculation event.

It is possible however to get a Streaming Index message without a LastValue field due to Optimisation where the actual price change granularity is less than two decimal places which is the dissemination limit.

REPLAY HANDLING

Whilst the FDI maintains sessions and failover behaviour it is expected that the client application will monitor TickCount numbers and invoke Replay requests where required for Pulse Indices.

The onus is on the client application to request any missed ticks through Replay.

If the replay request cannot be fulfilled due to lack of requested data then the application will receive a SubscriptionError.NOT_FOUND message.

Please note that Replay requests are only actioned for the current trade date and there is no facility to Replay prior trade day data at this time.