CLIENT DATABASE SECURITY

1502 RXR Plaza
15th Floor, West Tower
Uniondale, NY 11556
Telephone: (516) 227-6600 Facsimile: (516) 227-1799
Website: http://www.openlink.com

Revision History

Document Name	Date	Description
Client Database Security	March 2010	Initial Release
Client Database Security	June 2010	Added reference stating OpenLink no longer supports legacy utility Exp
Client Database Security	February 2011	Added section 1.3.2 Shipping Physical Devices
Client Database Security	March 2011	Added item 1.5.a documenting the current encryption key fingerprint
Client Database Security	May 2012	Added info on Private Key Generation
Client Database Security	Jan. 2013	Clarified database delivery instructions
Client Database Security	Dec. 2013	Updated Key Fingerprint and key download location
Client Database Security	Apr 2015	Updated FTP Protocols and policy language mandate
Client Database Security	Oct 2015	Updated to reflect deprecation of FTP Port 21

Table of Contents

1 CLIENT DATABASE HANDLING POLICIES

1.1 Overview

This document describes the way in which OpenLink handles confidential client proprietary information, specifically the processing and handling of client databases.

1.2 Purpose of Client Database Delivery

Client databases may be required as part of the OpenLink Support process, particularly when attempting to reproduce issues that require specific data or configurations.

The first step in all issue resolution is recreation of the behavior in a similar test environment. The OpenLink Support Team attempts to reproduce client reported system issues utilizing various OpenLink support tools working on local internal test databases.

OpenLink Support will typically first try to recreate a reported issue on a generic testing database that does not contain client specific data. If successful, the support process continues and a Service Request Ticket (SR) is created in OpenLink's internal systems.

In situations where issues cannot be reproduced in local test databases, OpenLink will look to recreate the issue in an existing client database. In situations where the issue still cannot be recreated, OpenLink may ask clients to deliver a database that exhibits the system behavior. In these situations, OpenLink is typically requesting delivery of a copy of the client's test or production database.

In the event where a database is not or cannot be delivered, OpenLink will continue to attempt recreation, but will likely require significant additional time to turn around a resolution to the issue. In all cases, issues cannot be resolved without proper recreation.

1.3 How Databases Are Delivered to OpenLink

OpenLink understands the confidentiality of database deliveries and as such has enforced a standard policy to ensure that both parties are aware of the requirements and commitments. The database delivery process begins when the client completes the "OpenLink Database Delivery and Load Form" located on the OpenLink CRM portal. This form contains vital information that OpenLink System Administrators need to successfully load the database. The form requests information such as Database Server Type (for example, Oracle, or MSSQL), Database Operating System and Database Size.

Databases can be delivered in one of the following supported forms:

1. DVDs
2. External Hard Disk or USB Thumb Drive
3. Electronically on an OpenLink secure FTP site (see the following note)

When delivering any data via FTP, clients must use SFTP or FTPS to ensure that all transmissions are encrypted.

Note: The OpenLink Database Delivery and Load Form should accompany every database delivery (External Hard Disk, USB Thumb Drive, or delivery via FTP), and in addition, be e-mailed to support@openlink.com. Also, for External Hard Disk deliveries, please send the USB and power cable.

The OpenLink Client FTP Site supports the following File Transfer Protocols:

- FTP with TLS/SSL (Port 990 Implicit) over Port 990
- SFTP using SSH2 (Secure Shell) over Port 22
- Unencrypted FTP is no longer supported

When delivering databases via DVD, External Hard Drive or USB Thumb Drives, OpenLink requires that the database be delivered to OpenLink's Corporate Office sent to the Attention of the "Database Administration Department."

Regardless of the delivery method, all databases must be encrypted and sensitive data fields obfuscated or OpenLink will not accept it. See section 1.4 and 1.5 for details.

1.3.1 Large Database Deliveries

In order to facilitate large database deliveries, OpenLink suggests splitting the database dump into smaller manageable file sizes. When using Oracle 10G and higher and delivering databases that are 20GB or greater, OpenLink recommends that the database be split into 8GB schema dump files. The advantages to sending smaller database files include shorter FTP upload and download transfer times.
In order to split database files, the following utility can be used.

Oracle's utility 'expdp' (datapump) splits dump files into smaller files while exporting the database schema (schema export only):

For example, executing the command '*Expdp schema/schema FILESIZE=8000M (bytes) dumpfile=file_%u.dmp*', will split the schema dump files into 8GB files.

Please note that OpenLink Financial no longer supports legacy utility Exp, please use Expdp only.
To reduce, further, the file size after splitting the database schema into smaller files, clients can use file compression utilities such as WinZip, Winrar on Windows, and gun zip or tar on Linux systems. Clients can also use WinZip or Winrar with public and private key encryption methods to encrypt the files. Any method used must be explicitly noted on the "OpenLink Database Delivery and Load Form."

1.3.2 Shipping Physical Devices

When shipping physical devices, such as hard drives, the device must be fully wrapped in anti-static packaging material (i.e., anti-static bag) and tightly packed in the shipping container in order to avoid excessive movement during shipping. Anti-static protection is especially important for hard drives, which are highly susceptible to electrostatic discharge (ESD), especially in cold weather. The packaging container should be of sufficient size to accommodate the physical device, but not too large where excessive movement of the device could occur

during shipping. As a best practice, OpenLink provides shipping insurance for all physical devices that are returned to clients.

1.4 Data Obfuscation

Obfuscation is the process by which client data is encoded (or transformed) to non client-specific information within the database. OpenLink clients must utilize obfuscation to overwrite reference data (such as account information, client names, market data curves, and any other fields that the client would deem as sensitive) so that this information cannot be understood by others outside the organization.

OpenLink can supply clients with a standard Obfuscation Script (STD_Encrypt_Database) that will encrypt selected sensitive data within the database. As an example, this script replaces the actual counterparty names with replacement pseudo names (actually id numbers). Please note that this script modifies data in the database and should only be run on copies of the database prior to delivery to OpenLink.

For Data Obfuscation in V16.0, please refer to the Database Obfuscation User Guide in the CRM Portal.

1.5 Database Encryption

OpenLink requires that all client databases, in all formats, be encrypted using standard encryption practices before delivery to OpenLink. Below are the basic guidelines for encrypting databases for acceptance by OpenLink.

OpenLink uses GnuPg as an encryption utility. GnuPg is compatible with most PGP releases and is available for most Linux distributions. Gpg4win is compatible with Windows and is available for download at http://www.gpg4win.org/ (full version is recommended).

Note: The following Private Key Generation is not required if you have an existing private key. Once a Private Key is generated, this step is no longer required.

Private Key Generation

```
> gpg --gen-key
```

gpg (GnuPG) 2.0.17; Copyright (C) 2011 Free Software Foundation, Inc. This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:

(1) RSA and RSA (default)

(2) DSA and Elgamal

(3) DSA (sign only)

(4) RSA (sign only)

Your selection? “1”

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048) "2048"

Requested keysize is 2048 bits

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0) "0"

Key does not expire at all

Is this correct? (y/N) "y"

GnuPG needs to construct a user ID to identify your key.

Real name: ("**unique user ID**")

Email address: ("**unique e-mail ID**")

Comment: (Optional)

You selected this USER-ID:

unique user ID <unique e-mail ID**>**

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? "**O**"

You need a Passphrase to protect your secret key.

"Enter and Re-enter your passphrase"

We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

gpg: C:/AppData/Roaming/gnupg/trustdb.gpg: trustdb created

gpg: key BA2EE343 marked as ultimately trusted

public and secret key created and signed.

gpg: checking the trustdb

gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model

gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u

pub 2048R/BA2EE343 2012-05-29

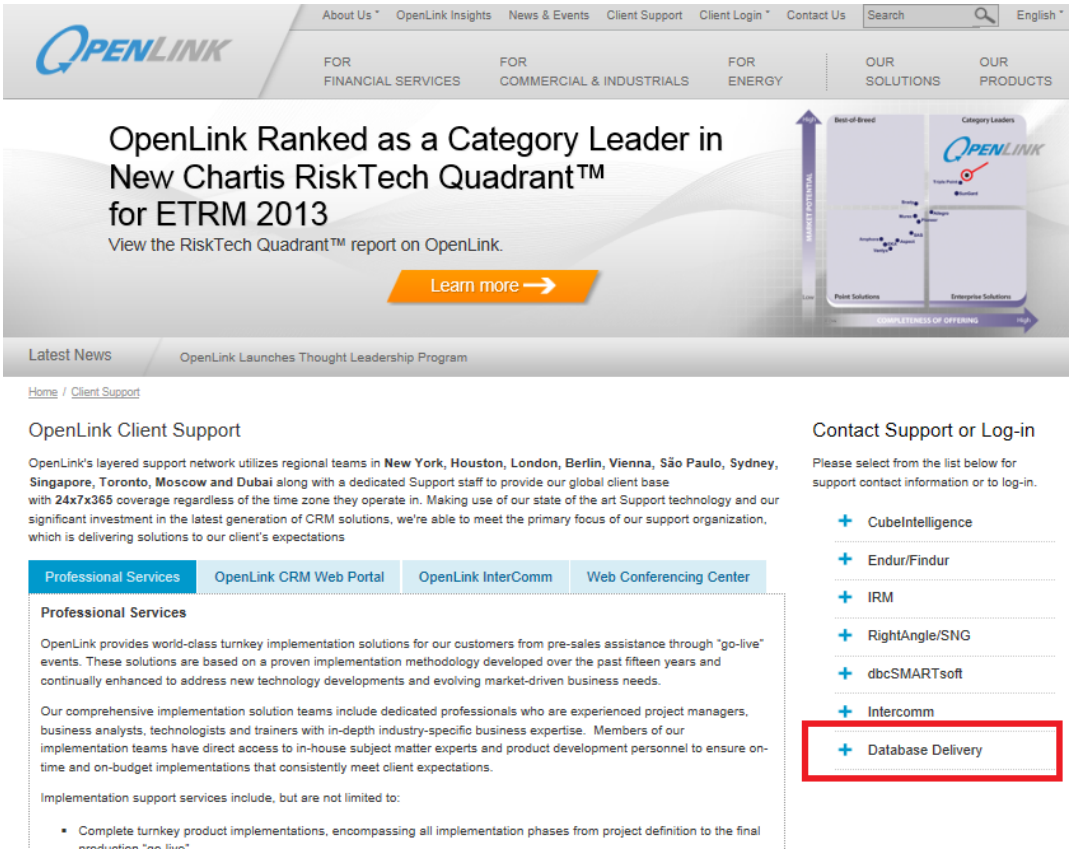
Key fingerprint = 0BE1 3F5F 0CFD 0F19 EF7A 6EDA B940 A1AE BA2E E343

uid unique user ID <unique e-mail ID**>**

sub 2048R/BF9EB308 2012-05-29

Import OLF's Public Key

The public key is available at http://www.openlink.com on the Client Support page.



This will download the key as a ".zip" file. You should extract the zip file and rename openlink.asc to openlink.txt. That file should then be imported into your key ring.

Please Note that the commands are the same for a Linux session and a Windows command prompt.

- **gpg --import openlink.txt**

Sign the key

- **gpg --edit-key clientencrypt@openlink.com**

```
pub  1024D/3E773845  created: 2013-11-14  expires: never       usage: SC
                     trust: unknown       validity: unknown
sub  2048g/A49B997C  created: 2013-11-14  expires: never       usage: E
[ unknown] (1). Openlink Encryption clientencrypt@openlink.com
```

```
Command> fpr
pub   1024D/3E773845 2013-11-14 Openlink Encryption
clientencrypt@openlink.com
Primary key fingerprint: FF44 9F0B 2A81 2621 CD18  C1E3 E130 DC3A 3E77
3845
```

A key's fingerprint is verified with the key's owner. This may be done in person, over the phone, or through any other means as long as you can guarantee that you are communicating with the key's true owner. If the fingerprint you get is the same as the fingerprint the key's owner gets, then you can be sure that you have a correct copy of the key. The fingerprint for OpenLink's encryption key can be verified against the fingerprint below published by OpenLink. Should any concerns arise regarding this verification, please contact the OpenLink Client Support Group.

OpenLink Encryption Key Fingerprint

FF44 9F0B 2A81 2621 CD18 C1E3 E130 DC3A 3E77 3845

After checking the fingerprint, you may sign the key to validate it. Since key verification is a weak point in public-key cryptography, you should be extremely careful and *always* check a key's fingerprint with the owner before signing the key.

```
Command> sign
pub  1024D/3E773845  created: 2013-11-14  expires: never       usage: SC
                     trust: unknown       validity: unknown
Primary key fingerprint: FF44 9F0B 2A81 2621 CD18  C1E3 E130 DC3A 3E77 3845
        Openlink Encryption <clientencrypt@openlink.com>
```

Are you really sure that you want to sign this key with your key: "Client <client@client.org>"

```
Really sign? Y
```

```
Enter your private key passphrase
```

Once signed you can check the key to list the signatures on it and see the signature that you have added. Every user ID on the key will have one or more self-signatures as well as a signature for each user that has validated the key.

```
Command> check
uid  <clientencrypt@olf.com>
sig!       A6FAC002 2008-07-15  [self-signature]
sig!       C19A87AB 2011-03-24   <unique user ID <unique e-mail ID>
Command> quit
```

The client will then be able to encrypt their file using our key as the recipient.

- **gpg --output db.gpg --encrypt --recipient clientencrypt@openlink.com db.dmp**

The output file “db.gpg” is the encrypted file and can now be sent to OpenLink through the supported transmission format.

1.6 Data Sanitation

Databases delivered to OpenLink are loaded on internal servers and sanitized from the original delivery media. Databases uploaded to the FTP site will be deleted from the FTP site within 30 days from the time of upload.

External Hard Drives and USB Thumb Drives will be erased using a commercial sanitation product that overwrites the contents of the drive.