

**XLX マルチプロトコルゲートウェイリフレクター
インストール
(編集中)**

Shu JA3GQJ/JR3VHクラブ

リフレクターって何だろうから始まって G2+D-Plus から ircDDBGateway に軸足を移し、ircDDBGateway フォーラムに「リフレクターを開設するにはどうしたらよいか」を投稿しました。その結果、ルクセンブルクの Luc, XL1IQ から、「XLX Multiprotocol Reflector Gateway」なるソフトを進められ無料で使用できることが判り Linux の初心者である当局でもソフトのインストールが出来たが自信がなかったので彼には Teamviewer で当局のサーバーをチェックしてもらって OK となったのが 3 月 6 日午前 1 時でした。さらに [X Reflector Forum](#) で、John K6KD へ「XLX Multiprotocol Reflector Gateway」で取得した XLX081 の登録を依頼して RXF081 が JR3VH で試験運用できるように成りました。

前提条件

次の条件が整っているものとしてソフトをインストールする。

1. インターネットにたくさん紹介されている web ページを参照して CentOS6.7 がインストールされている。
2. <https://groups.yahoo.com/neo/groups/xlxd-star/files> からダウンロードした XLX Installation Doc. for CentOS. pdf ファイルを CentOS のデスクトップに開いたら CentOS の「端末」(コマンド入力画面)を開いて XLX Installation Doc. for CentOS. pdf ファイルの「#」の部分例えば「sudo yum install git git-core」をコピーして貼り付ける作業を繰り返しインストールを終了する。
3. ルーターは次のポートを開放している (カッコ内は用途)。
TCP port 80 (http) 、 TCP port 443 (https) 、 TCP port 8080 (RepNet)、
TCP port 22 (ssh) または TCP port 10022、
UDP port 10001 (json interface XLX Core)、 UDP port 10002 (XLX interlink)、
UDP port 30001 (DExtra protocol) 、 UDP port 20001 (DPlus protocol)、
UDP port 30051 (DCS protocol)
4. ダッシュボード用 WEB サーバーはプロバイダーの前述のルーターポート開放の対応有無の心配があるのでレンタルサーバーは使用しない。
5. 私設サーバーのグローバル IP アドレスは <http://ieserver.net/>の無料ダイナミック DNS(DDNS)サービスを利用する。
6. プロバイダーの都合やルーターの電源が落ちると使用する IP アドレスが変わってしまうのでリフレクターはドメイン名で公開する。

7. リフレクター番号は dvc@rlx.lu にメールをして出来れば自分が希望する番号を取得する。
8. 初めにも申し上げたように今回、公開の手続きを取ったのは XRF081(Dextra)ですが他に REF081(DPlus)と DCS081(DCS)を加えて三種類のプロトコルによる三つのリフレクターが使用できます。
9. 公開手続き上の問題は別としてこのプログラムは広い意味でヨーロッパ系とかアメリカ系のリフレクターの壁を取り除こうとしているようにも思えます。
10. お手持ちの dongle などのリフレクターリストに XRF081、REF081、DCS081
- 11.

ソフトのインストール

32ビット版か64ビット版のCentOS6.7がインストールされているサーバー用PCの端末ウインドウを開いて次のコマンドを順次コピーして貼り付けて行きながらソフトのインストールを進める。

1. 次のコマンドでホスト名とFQDNをチェックする。

```
# hostname
```

当局の場合はja3gqj.dip.jpが返ってきた。

次のコマンドで“Unknown host”が返されたら・・・

```
# hostname -f
```

次のコマンドでFQDNをセットする。

```
# echo "your IP" "FQDN" >> /etc/hosts
```

なお“your IP”にはPCのローカル固定IPアドレスを“FQDN”にはドメイン名を入れる。

2. 次のコマンドでSELinuxのレベルを確認する

```
# getenforce
```

このコマンドで、もし、“Enforcing”が返ってきたら次のコマンドでレベルを“Permissive”にする。

```
# setenforce 0
```

PC（サーバー）が再起動されてもSELinuxのレベルが“Permissive”に保たれるよう次のコマンドでSELinux設定画面を開いてSELINUX=enforcingの行をSELINUX=permissiveに変更する。

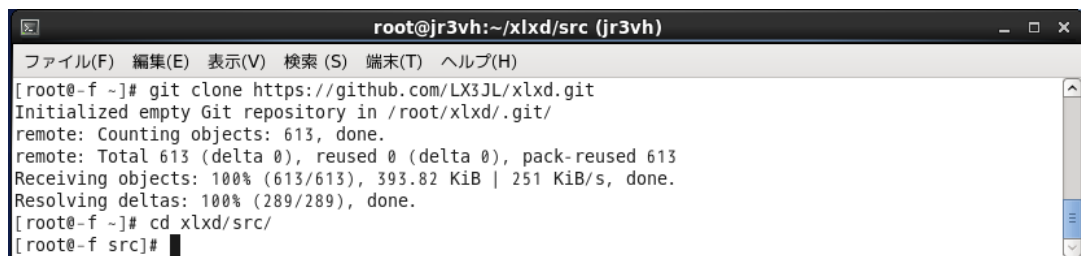
```
# nano /etc/selinux/config または # gedit /etc/selinux/config
```

なお、nano やgeditはテキストエディターで、当局は使いやすいgeditを使った。

3. 次はOSのアップグレードのコマンドのようですが当局の場合特に変化はなかった。

```
# sudo yum install upgrade
```

4. 次のコマンドでgit (ギット) をインストールする。
sudo yum install git git-core
5. 次のコマンドでApach (ウェブサーバー) とPHP5をインストールする。
sudo yum install httpd php
6. 次のコマンドでg++ コンパイラをインストールする。
sudo yum install centos-release-scl-rh
sudo yum install devtoolset-3-gcc devtoolset-3-gcc-c++
scl enable devtoolset-3 bash
7. 次のコマンドでXLX のソースをダウンロードしてコンパイルする。
git clone <https://github.com/LX3JL/xlxd.git>

A terminal window titled 'root@jr3vh:~/xlxd/src (jr3vh)' showing the execution of 'git clone https://github.com/LX3JL/xlxd.git'. The output shows the repository being initialized, objects counted, and deltas resolved. The prompt then changes to '[root@-f ~]# cd xlxd/src/' and '[root@-f src]#'.

```
root@jr3vh:~/xlxd/src (jr3vh)
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[root@-f ~]# git clone https://github.com/LX3JL/xlxd.git
Initialized empty Git repository in /root/xlxd/.git/
remote: Counting objects: 613, done.
remote: Total 613 (delta 0), reused 0 (delta 0), pack-reused 613
Receiving objects: 100% (613/613), 393.82 KiB | 251 KiB/s, done.
Resolving deltas: 100% (289/289), done.
[root@-f ~]# cd xlxd/src/
[root@-f src]#
```

- ```
cd xlxd/src/
make
make clean
make install
```
8. 次のコマンドで“xlxd”フォルダーのスタートアップスクリプト (記述) を  
/etc/init.d へコピーしてソフトのインストールを終了する。  
# cp ~/xlxd/scripts/xlxd /etc/init.d/

```
root@jr3vh:~/xlxd/src (jr3vh)
[root@-f ~]# cd xlxd/src/
[root@-f src]# make
g++ -c -std=c++11 -pthread cbuffer.cpp -o cbuffer.o
g++ -c -std=c++11 -pthread ccallsign.cpp -o ccallsign.o
g++ -c -std=c++11 -pthread ccallsignlist.cpp -o ccallsignlist.o
g++ -c -std=c++11 -pthread ccallsignlistitem.cpp -o ccallsignlistitem.o
g++ -c -std=c++11 -pthread cclient.cpp -o cclient.o
g++ -c -std=c++11 -pthread cclients.cpp -o cclients.o
g++ -c -std=c++11 -pthread cdcsclient.cpp -o cdcsclient.o
g++ -c -std=c++11 -pthread cdcsprotocol.cpp -o cdcsprotocol.o
g++ -c -std=c++11 -pthread cdextraclient.cpp -o cdextraclient.o
g++ -c -std=c++11 -pthread cdextraprotocol.cpp -o cdextraprotocol.o
g++ -c -std=c++11 -pthread cdplusclient.cpp -o cdplusclient.o
g++ -c -std=c++11 -pthread cdplusprotocol.cpp -o cdplusprotocol.o
g++ -c -std=c++11 -pthread cdvframepacket.cpp -o cdvframepacket.o
g++ -c -std=c++11 -pthread cdvheaderpacket.cpp -o cdvheaderpacket.o
g++ -c -std=c++11 -pthread cdvlastframepacket.cpp -o cdvlastframepacket.o
g++ -c -std=c++11 -pthread cgatekeeper.cpp -o cgatekeeper.o
g++ -c -std=c++11 -pthread cip.cpp -o cip.o
g++ -c -std=c++11 -pthread cnotification.cpp -o cnotification.o
g++ -c -std=c++11 -pthread cpacket.cpp -o cpacket.o
g++ -c -std=c++11 -pthread cpacketstream.cpp -o cpacketstream.o
g++ -c -std=c++11 -pthread cpeerccallsignlist.cpp -o cpeerccallsignlist.o
g++ -c -std=c++11 -pthread cprotocol.cpp -o cprotocol.o
g++ -c -std=c++11 -pthread cprotocols.cpp -o cprotocols.o
g++ -c -std=c++11 -pthread creflector.cpp -o creflector.o
g++ -c -std=c++11 -pthread ctimepoint.cpp -o ctimepoint.o
g++ -c -std=c++11 -pthread cudpsocket.cpp -o cudpsocket.o
g++ -c -std=c++11 -pthread cuser.cpp -o cuser.o
g++ -c -std=c++11 -pthread cusers.cpp -o cusers.o
g++ -c -std=c++11 -pthread cxlxclient.cpp -o cxlxclient.o
g++ -c -std=c++11 -pthread cxlxprotocol.cpp -o cxlxprotocol.o
g++ -c -std=c++11 -pthread main.cpp -o main.o
g++ -std=c++11 -pthread cbuffer.o ccallsign.o ccallsignlist.o ccallsignlistitem.o cclient.o cclients.o
cdcsclient.o cdcsprotocol.o cdextraclient.o cdextraprotocol.o cdplusclient.o cdplusprotocol.o cdvframep
acket.o cdvheaderpacket.o cdvlastframepacket.o cgatekeeper.o cip.o cnotification.o cpacket.o cpacketstr
eam.o cpeerccallsignlist.o cprotocol.o cprotocols.o creflector.o ctimepoint.o cudpsocket.o cuser.o cuser
s.o cxlxclient.o cxlxprotocol.o main.o -o xlxd
[root@-f src]# make clean
rm *.o
[root@-f src]# make install
mkdir -p /xlxd
cp ./xlxd /xlxd/
cp -i ../config/xlxd.blacklist /xlxd/
cp -i ../config/xlxd.whitelist /xlxd/
cp -i ../config/xlxd.interlink /xlxd/
[root@-f src]# cp ~/xlxd/scripts/xlxd /etc/init.d/
[root@-f src]#
```

## インストール後の設定

1.// adapt the default startup parameters to your needs

# nano /etc/init.d/xlxd

// to make the startup script working under CentOS following lines need to be replaced.

The top line is the original line (commented out with a #). The next line is the line to replace with:

#. /Lib/lsb/init-functions

**. /etc/init.d/functions**

# start-stop-daemon --start --exec \$DAEMON --chuid \$USER:\$GROUP --background --  
\$ARGUMENTS

**\$DAEMON \$ARGUMENTS**

# start-stop-daemon --stop --pidfile \$PIDFILE

**killproc \$DAEMON**

```
#!/bin/sh
BEGIN INIT INFO
Provides: xlxd service
Processname: xlxd
Required-Start: $remote_fs $syslog
Required-Stop: $remote_fs $syslog
Default-Start: 2 3 4 5
Default-Stop: 0 1 6
Short-Description: starts the dstar reflector as a linux service
Description: dstar reflector by LX3JL
start script by LXIIQ
END INIT INFO

PATH=/sbin:/bin:/usr/sbin:/usr/bin

#. /lib/lsb/init-functions
. /etc/init.d/functions

change below settings according to your system
NAME="xlxd"
DAEMON="/xlxd/xlxd"
ARGUMENTS="XLX270 158.64.26.132"
PIDFILE="/var/log/xlxd.pid"
USER=root
GROUP=root

[-f $DAEMON] || exit 0

start () {
start daemon

 echo -n "Starting $NAME: "
 # start-stop-daemon --start --exec $DAEMON --chuid $USER:$GROUP --background -- $ARGUMENTS
 $DAEMON $ARGUMENTS
 RETVAL=$?
 echo
 sleep 4
 echo `pidof xlxd` > $PIDFILE
 return $RETVAL
}

stop () {
 echo -n "Stopping $NAME: "
 #start-stop-daemon --stop --pidfile $PIDFILE
 killproc $DAEMON
 RETVAL=$?
 echo
}
```

2.次のコマンドでサービスをスタートしてストップする。

```
service xlxd start
```

```
service xlxd stop
```

```
root@jr3vh:~/xlxd/src (jr3vh)
cp -i ../config/xlxd.blacklist /xlxd/
cp -i ../config/xlxd.whitelist /xlxd/
cp -i ../config/xlxd.interlink /xlxd/
[root@-f src]# cp ~/xlxd/scripts/xlxd /etc/init.d/
[root@-f src]# gedit /etc/init.d/xlxd
[root@-f src]# service xlxd start
Starting xlxd:
[root@-f src]# service xlxd stop
Stopping xlxd: [OK]
[root@-f src]#
```

// copy dashboard to /var/www/html

# cp -r ~/xlxd/dashboard/\* /var/www/html

```
root@jr3vh:~/xlxd/src (jr3vh)
cp -i ../config/xlxd.whitelist /xlxd/
cp -i ../config/xlxd.interlink /xlxd/
[root@-f src]# cp ~/xlxd/scripts/xlxd /etc/init.d/
[root@-f src]# gedit /etc/init.d/xlxd
[root@-f src]# service xlxd start
Starting xlxd:
[root@-f src]# service xlxd stop
Stopping xlxd: [OK]
[root@-f src]# cp -r ~/xlxd/dashboard/* /var/www/html
[root@-f src]#
```

// give the dashboard read access to the server log file

# chmod +r /var/log/messages



```
config.inc.php (/var/www/html/pgs) - gedit
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
[アイコン] 開く [アイコン] 保存 [アイコン] 元に戻す [アイコン] [アイコン] [アイコン] [アイコン] [アイコン]
@ config.inc.php x
<?php
/*
Possible values for IPModus

HideIP
ShowFullIP
ShowLastByteOfIP
ShowLast2ByteOfIP
ShowLast3ByteOfIP

*/

$Service = array();
$CallingHome = array();
$PageOptions = array();

$PageOptions['ContactEmail'] = 'dvc@rlx.lu'; // Support E-Mail address

$PageOptions['DashboardVersion'] = '2.2.3'; // Dashboard Version

$PageOptions['PageRefreshActive'] = true; // Activate automatic refresh
$PageOptions['PageRefreshDelay'] = '10000'; // Page refresh time in miliseconds

$PageOptions['RepeatersPage'] = array();
$PageOptions['RepeatersPage']['LimitTo'] = 99; // Number of Repeaters to show
$PageOptions['RepeatersPage']['IPModus'] = 'ShowLast2ByteOfIP'; // See possible options above
$PageOptions['RepeatersPage']['MasqueradeCharacter'] = '*'; // Character used for masquerade

$PageOptions['PeerPage'] = array();
$PageOptions['PeerPage']['LimitTo'] = 99; // Number of peers to show
$PageOptions['PeerPage']['IPModus'] = 'ShowFullIP'; // See possible options above
$PageOptions['PeerPage']['MasqueradeCharacter'] = '*'; // Character used for masquerade

$PageOptions['ModuleNames'] = array(); // Module nomination
$PageOptions['ModuleNames']['A'] = 'Int.';
$PageOptions['ModuleNames']['B'] = 'Regional';
$PageOptions['ModuleNames']['C'] = 'National';
$PageOptions['ModuleNames']['D'] = '';

$PageOptions['MetaDescription'] = 'XLX is a D-Star Reflector System for Ham Radio Operators.'; // Meta Tag Values, usefull for Search Engine
$PageOptions['MetaKeywords'] = 'Ham Radio, D-Star, XReflector, XLX, XRF, DCS, REF, '; // Meta Tag Values, usefull forSearch Engine
$PageOptions['MetaAuthor'] = 'LX1IQ'; // Meta Tag Values, usefull for Search Engine
$PageOptions['MetaRevisit'] = 'After 30 Days'; // Meta Tag Values, usefull for Search Engine
$PageOptions['MetaRobots'] = 'index,follow'; // Meta Tag Values, usefull for Search Engine

PHP タブの幅: 8 (1行, 1列) [挿入]
```

```
config.inc.php (/var/www/html/pgs) - gedit
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
[アイコン] 開く [アイコン] 保存 [アイコン] 元に戻す [アイコン] [アイコン] [アイコン] [アイコン] [アイコン]
@ config.inc.php x

$PageOptions['DashboardVersion'] = '2.2.3'; // Dashboard Version

$PageOptions['PageRefreshActive'] = true; // Activate automatic refresh
$PageOptions['PageRefreshDelay'] = '10000'; // Page refresh time in miliseconds

$PageOptions['RepeatersPage'] = array();
$PageOptions['RepeatersPage']['LimitTo'] = 99; // Number of Repeaters to show
$PageOptions['RepeatersPage']['IPModus'] = 'ShowLast2ByteOfIP'; // See possible options above
$PageOptions['RepeatersPage']['MasqueradeCharacter'] = '*'; // Character used for masquerade

$PageOptions['PeerPage'] = array();
$PageOptions['PeerPage']['LimitTo'] = 99; // Number of peers to show
$PageOptions['PeerPage']['IPModus'] = 'ShowFullIP'; // See possible options above
$PageOptions['PeerPage']['MasqueradeCharacter'] = '*'; // Character used for masquerade

$PageOptions['ModuleNames'] = array(); // Module nomination
$PageOptions['ModuleNames']['A'] = 'Int.';
$PageOptions['ModuleNames']['B'] = 'Regional';
$PageOptions['ModuleNames']['C'] = 'National';
$PageOptions['ModuleNames']['D'] = '';

$PageOptions['MetaDescription'] = 'XLX is a D-Star Reflector System for Ham Radio Operators.'; // Meta Tag Values, usefull for Search Engine
$PageOptions['MetaKeywords'] = 'Ham Radio, D-Star, XReflector, XLX, XRF, DCS, REF, '; // Meta Tag Values, usefull forSearch Engine
$PageOptions['MetaAuthor'] = 'LX1IQ'; // Meta Tag Values, usefull for Search Engine
$PageOptions['MetaRevisit'] = 'After 30 Days'; // Meta Tag Values, usefull for Search Engine
$PageOptions['MetaRobots'] = 'index,follow'; // Meta Tag Values, usefull for Search Engine

$Service['PIDFile'] = '/var/log/xld.pid';
$Service['XMLFile'] = '/var/log/xld.xml';

$CallingHome['Active'] = false; // xlx phone home, true or false
$CallingHome['MyDashboardURL'] = 'http://your_dashboard'; // dashboard url
$CallingHome['ServerURL'] = 'http://xlxapi.rlx.lu/api.php'; // database server, do not change !!!!
$CallingHome['PushDelay'] = 600; // push delay in seconds
$CallingHome['Country'] = 'your.country'; // Country
$CallingHome['Comment'] = 'your.comment'; // Comment. Max 100 character
$CallingHome['HashFile'] = '/tmp/callinghome.php'; // Make sure the apache user has read and write permissions
in this folder.
$CallingHome['OverrideIPAddress'] = ''; // Leave blank for autodetection.

?>

PHP タブの幅: 8 (1行, 1列) [挿入]
```

// start the webserver

# service httpd start

```
// enable xlxd and httpd service on server startup
```

```
chkconfig httpd on
```

```
chkconfig xlxd on
```

```
// reboot server to see if the auto-start is working
```

```
reboot
```

By default, on CentOS iptables is enabled. You need to open the needed ports !!

If you are behind a hardware firewall you can stop the service.

```
service iptables stop
```

### **Firewall settings:**

XLX Server requires the following ports to be open and forwarded properly for in- and outgoing

network traffic:

TCP port 80 (http) optional TCP port 443 (https)

TCP port 8080 (RepNet)

UDP port 10001 (json interface XLX Core)

UDP port 10002 (XLX interlink)

TCP port 22 (ssh) alternative TCP port 10022

UDP port 30001 (DExtra protocol)

UPD port 20001 (DPlus protocol)

UDP port 30051 (DCS protocol)

### リフレクター開設の広報

1. 設定が終わり、ソフトを起動したら XRF081 B または REF081 B でリフレクターにアクセスして動作を確認する。

2. <http://xrflmaster.net/dns/>へ開設するリフレクターの登録をする。

暫くするとメールが入り次の一行を crontab に挿入した。どうやらリフレクターデータ自動更新のまじないらしい。

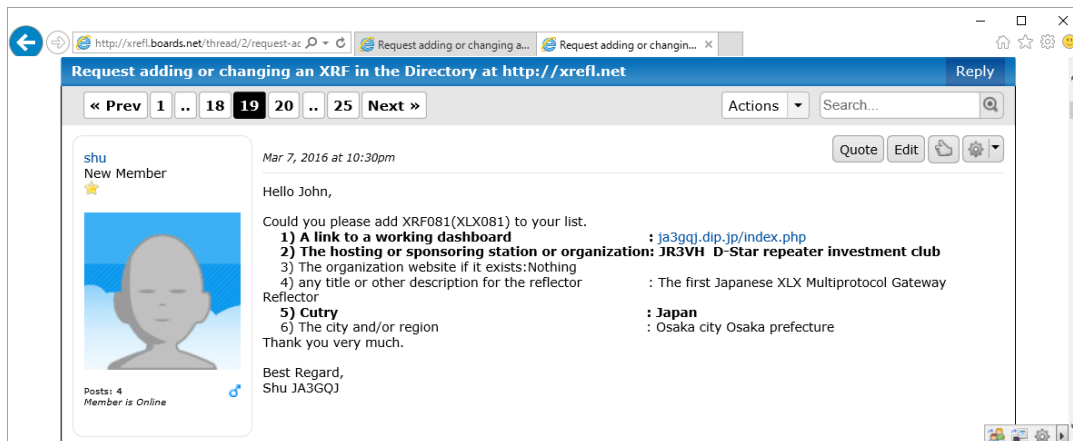
```
0 * * * * wget
```

```
http://xrflmaster.net/dns/update.php?key=xxxxxxxxxxxxxxxxxxxxxxxxnOM&id=XRF081
```

3. 続いて

<http://xrfl.boards.net/thread/2/request-adding-changing-directory-xrfl?page=19>

(下図) のように [X Reflector Forum](#) で、John K6KD へ「XLX Multiprotocol Reflector Gateway」で取得した XLX081 の登録を依頼する。



4. 問題はリフレクターリストをどこから入手するか、新しいリフレクターをどのようにして手持ちの青色DVドングル、赤色DVAP, ThmbDV, DV4mini,DVmega などに追加するかが判り難い事である。取り敢えずは各ソフトが持っているリフレクターリストのテキストファイルを開けて「XRF081 ja3gqj.dip.jp」や「REF081 ja3gqj.dip.jp」を書き込んで使用を始めた。

5. その後 XRFxxx リフレクターリストは

[http://www.tms-itdienst.at/sites/default/files/downloads/DV/DExtra\\_Hosts.txt](http://www.tms-itdienst.at/sites/default/files/downloads/DV/DExtra_Hosts.txt) で

入手できるようであり DCSxxxリフレクターリストは

<http://xlxapi.rlx.lu/api.php?do=GetReflectorHostname> で入手できることが判った。(しかし、ドメイン名が使われていないので感心しない。)

特にDPlus 関係への手続きは何もしていないのでREFxxx は現在REF076 までしか公にないがこのソフトではXRF081 (Dextra系)の他にREF081とDCS081が使用できるので公開手続き上の問題は別として広い意味でヨーロッパ系とかアメリカ系のリフレクターの壁を取り除こうとしているようにも思える。

XLX yahoo support group at:

<https://groups.yahoo.com/neo/groups/xlxd-star/info>

グループ参加申し込みメールアドレス

[xlxd-star-subscribe@yahoogroups.com](mailto:xlxd-star-subscribe@yahoogroups.com)

Now you must make your XLX081 public.

A good start is <http://xrefl.net>

To link to the XLX081 you us: UR CALL

DCS081BL for DCS  
XRF081BL for DExtra  
REF081BL for DPlus